การควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทน
ด้วยฟังก์ชันเลียปูโนฟซึ่งขึ้นกับตัวแปรเสริมโดยใช้อสมการเมทริกซ์เชิงเส้น

นางสาวทูอันทิ โด

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549

ISBN 974-14-2962-2

ROBUST CONSTRAINED MODEL PREDICTIVE CONTROL WITH A PARAMETER-DEPENDENT LYAPUNOV FUNCTION USING LINEAR MATRIX INEQUALITIES

Miss Tu Anh Thi Do

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2006

ISBN 974-14-2962-2

Thesis Title	ROBUST CONSTRAINED MODEL PREDICTIVE CONTROL WITH A PARAMETER-DEPENDENT LYAPUNOV FUNCTION USING LINEAR MATRIX INEQUALITIES
By	Miss Tu Anh Thi Do
Field of Study	Electrical Engineering
Thesis Advisor	Assoc. Prof. David Banjerdpongchai, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of Requirements for Master's Degree

.... Dean of the Faculty of Engineering
(Professor Direk Lavansiri, Ph.D.)

THESIS COMMITTEE

.... Chairman
(Assistant Professor Manop Wongsaisuwan, D. Eng.)

.... Thesis Advisor
(Associate Professor David Banjerdpongchai, Ph.D.)

.... Member
(Naebboon Hoonchareon, Ph.D.)

.... Member
(Assistant Professor Montree Wongsri, Ph.D.)

ทูอันทิ โด: การควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทนด้วยฟังก์ชันเลียปูโนฟซึ่งขึ้นกับตัวแปรเสริมโดยใช้อสมการเมทริกซ์เชิงเส้น (ROBUST CONSTRAINED MODEL PREDICTIVE CONTROL WITH A PARAMETER-DEPENDENT LYAPUNOV FUNCTION USING LINEAR MATRIX INEQUALITIES), อ. ที่ปรึกษา: รศ.ดร.เดวิด บรรเจิดพงศ์ชัย, 90 หน้า, ISBN 974-14-2962-2

วิทยานิพนธ์นี้นำเสนอการควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทน สำหรับระบบเชิงเส้นแปรผันตามเวลาภายใต้ความไม่แน่นอนเชิงพารามิเตอร์ กฎการควบคุมประยุกต์ใช้ฟังก์ชันเลียปูโนฟ ซึ่งขึ้นกับตัวแปรเสริมที่สอดคล้องกับจุดยอดของความไม่แน่นอนพอลีโทพ แนวทางการออกแบบแบ่งเป็นสองส่วน ส่วนแรกเน้นการออกแบบกฎการป้อนกลับสถานะคงทน ณ เวลาการสุ่ม เพื่อทำให้ขอบเขตบนของฟังก์ชันวัตถุประสงค์มีค่าต่ำสุด ภายใต้เงื่อนไขบังคับบนสัญญาณเข้าควบคุมและสัญญาณออกของกระบวนการ ปัญหาการออกแบบการป้อนกลับสถานะสามารถแปลงเป็นปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์ ที่เกี่ยวกับอสมการเมทริกซ์เชิงเส้น ซึ่งหาคำตอบได้อย่างมีประสิทธิภาพ ส่วนที่สองเน้นการป้อนกลับสัญญาณออกคงทนที่ใช้การป้อนกลับสถานะซึ่งหาได้จากส่วนแรก ร่วมกับตัวประมาณสถานะ แนวทางการออกแบบแก้ปัญหาอสมการเมทริกซ์เชิงเส้นแบบออฟไลน์ เพื่อประกันเสถียรภาพคงทนของระบบแต่งเติมวงปิด เมื่อเปรียบเทียบกับงานก่อนหน้านี้ที่ออกแบบโดยใช้ฟังก์ชันเลียปูนอฟเดียว วิธีที่เสนอในวิทยานิพนธ์นี้ให้ผลลัพธ์ที่อนุรักษ์ลดลงและปรับปรุงขั้นตอนวิธี โดยเฉพาะอย่างยิ่ง วิธีการออกแบบสามารถจัดการกับพารามิเตอร์แปรผันตามเวลาไม่แน่นอนที่มีพิสัยกว้างกว่าเดิม สุดท้ายเรานำเสนอการประยุกต์ใช้กับหลาย ๆ ระบบ เพื่อแสดงประสิทธิผลของเทคนิคการออกแบบ

ภาควิชา ....วิศวกรรมไฟฟ้า........	**ลายมือชื่อนิสิต** ....
สาขาวิชา ....วิศวกรรมไฟฟ้า........	**ลายมือชื่ออาจารย์ที่ปรึกษา** ....
ปีการศึกษา ....2549........	

##4470557521: MAJOR ELECTRICAL ENGINEERING

KEY WORD: MODEL PREDICTIVE CONTROL / UNCERTAIN LINEAR TIME-VARYING SYSTEMS / PARAMETER-DEPENDENT LYAPUNOV FUNCTION / INPUT AND OUTPUT CONSTRAINT / STATE FEEDBACK / OUTPUT FEEDBACK / ASYMPTOTICALLY STABLE INVARIANT ELLIPSOID / OFF-LINE DESIGN / LINEAR MATRIX INEQUALITIES

TU ANH THI DO: ROBUST CONSTRAINED MODEL PREDICTIVE CONTROL WITH A PARAMETER-DEPENDENT LYAPUNOV FUNCTION USING LINEAR MATRIX INEQUALITIES, THESIS ADVISOR: ASSOC. PROF. DAVID BANJERDPONGCHAI, Ph.D., 90 pp., ISBN 974-14-2962-2

This thesis presents robust constrained model predictive control for linear time-varying systems under parametric uncertainties. In order to guarantee robust performance, the control law applies a parameter-dependent Lyapunov function which corresponds to vertices of the polytopic uncertainty. The design approach is divided into two parts. The first part is focused on the design of a robust state feedback law that minimizes, at each sampling time, an upper bound of the worst-case objective function, subject to constraints on control inputs and process outputs. The state feedback design problem is cast as convex optimization involving linear matrix inequalities (LMIs) which can be efficiently solved. The second part emphasizes on a robust output feedback scheme that utilizes the state feedback obtained from the first part together with state estimator. The synthesis approach is to solve off-line LMI problems to guarantee the robust stability of the augmented closed-loop system. In comparison with the previous work whose design employs a single Lyapunov function, the method proposed in this thesis yields less conservative performance and further improves the algorithm. In particular, the design method is capable of handling a wider range of uncertain time-varying parameters. Finally, several applications are presented to illustrate the effectiveness of the control technique.

Department ....Electrical Engineering........ **Student's signature** ....

Field of study ....Electrical Engineering........ **Advisor's signature** ....

Academic year ....2006........

Acknowledgments

Without the help and strong inspiration from my principle advisor, Associate Professor David Banjerdpongchai, this thesis would have never been completed. He has always made himself available for discussions, productive comments, useful suggestions and careful readings of this thesis. I am indebted to him for his ever-present guidance, encouragement, and support throughout my educational endeavor at Control Systems Research Laboratory, Chulalongkorn University.

I would like to express my sincere gratitude to Doctor Hoang Minh Son, my undergrad advisor, for enlightening me to the world of Model Predictive Control, for directing me on my study plan at Chulalongkorn University, and for his concern of my well-being. I, too, am thankful to Assistant Professor Manop Wongsaisuwan, Doctor Neabboon Hoonchareon and Assistant Professor Montree Wongsri for their enthusiasm and willingness to read the thesis and kindly serve on the committee, despite their tight schedules.

Many other Professors in the Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University also deserve special recognition for this thesis. To them, I am appreciative for their initial interest and guidance which has led to my current path. Among them, I am deeply thankful to Professor Varaporn Jaovisidha, Professor Watcharapong Khovidhungij, and Professor Suchin Arunsawatwong not only for providing me the fundamental background on Control Systems and Applications, but also for inspiring my interest in teaching. Many thanks go to CSRL members and my friends at Chulalongkorn University for their friendship and support. I gratefully acknowledge the JICA project for AUN/SEED-Net (www.seed-net.org) for full financial support throughout my higher education.

Finally, I would like to thank my parents, Do Van Chin and Duong Thi Yen; my uncle, Ninh Duc Ton and my aunt, Duong Thi Oanh. For their many years of faithful love, overwhelming support, and brilliant advice, I shall remain eternally grateful.

Contents

List of Figures

List of Notations

Symbols

$\mathcal{R}$	Input weighting matrix of the LQ
$\mathcal{Q}$	State weighting matrix of the LQ
Δ	Uncertainty block
$\mathbf{\Omega}$	Uncertainty set of LTV systems
$\mathbf{\Psi}$	Uncertain set of explicit state feedback gains
$\mathbf{diag}[X_1, \ldots, X_n]$	Block diagonal matrix with diagonal elements equal to matrices $X_1, \ldots, X_n$
J_{LQ}	Traditional LQ performance index
J_{WC}	Worst-case performance index
$\mathcal{H}_\infty$	Hardy space of essentially bounded functions on the imaginary axis, with analytic continuation into right half-plane
$\mathcal{E}$	The ellipsoid $\mathcal{E} = \{x \in \mathbf{R}^n \mid x^T Q^{-1} x \leq 1\}$
F	State feedback gain matrix
P	Lyapunov matrix
$\mathbb{Q}$	Defines the ellipsoid $\mathcal{E} = \{x \in \mathbf{R}^n \mid x^T \mathbb{Q}^{-1} x \leq 1\}$, used for PDLF case
γ	Upper bound of J_{WC}
$(\cdot)^T$	Transpose of $(\cdot)$
$\lvert v \rvert$	ℓ_1 - norm of vector v
$\lVert v \rVert_2$	Euclidean norm of vector v
$\lVert x \rVert_{Q^{-1}}$	Weighted norm of vector x with respect to Q^{-1}
$x(k+i \mid k)$	State at time $k+i$, predicted based on the measurement at time k
$y(k+i \mid k)$	Plant output at time $k+i$, predicted based on the measurement at time k
$u(k+i \mid k)$	Control input at time $k+i$, computed at time k
L	Number of vertices of polytopic uncertainty set
L_p	Estimator gain matrix
ρ	Minimum decay rate of nominal error dynamics
λ	Uncertain parameter

Acronyms

Co	Convex hull
MPC	Model Predictive Control
RCMPC	Robust Constrained Model Predictive Control
LMI	Linear Matrix Inequality
LTV	Linear Time-Varying
LTI	Linear Time-Invariant
LQ	Linear Quadratic
PDLF	Parameter-Dependent Lyapunov Function
SLF	Single Lyapunov Function

Chapter 1

INTRODUCTION

1.1 Motivation

Model Predictive Control (MPC), also known as Moving Horizon Control (MHC) or Receding Horizon Control (RHC), has been adopted in industry as an effective means to deal with multivariable constrained control problems. There are many applications of predictive control successfully in use at present, especially in process industry, such as in the cement industry, drying towers, robot arms [1] or in distillation columns, PVC plants, steam generators [2, 3].

Due to its advantages over other control methods and some new and very promising results, one can think that MPC will experience greater expansion in the near future. However, it also has its drawbacks. One of these is that its implementation needs some computational complexities. When constraints are considered, the amount of computation required is even higher. This is not a problem for the research community where mathematical packages are fully available but a drawback for the use of the technique by control engineers in practice. Another disadvantage is the need for an appropriate model of the plant. Since the design algorithm is based on the prior knowledge of the model, it is obvious that the obtained benefits will be affected by the plant-model mismatch. Because of the above reasons, we are interested in doing research on the synthesis of robust constrained model predictive controllers (RCMPC) using Linear Matrix Inequalities. The proposed technique will be applied to some typical examples to demonstrate the effectiveness of the derived control law.

1.2 Literature Review

Model Predictive Control is an effective multivariable constrained control algorithm in which a dynamic optimization problem is solved on-line. At each sampling time, MPC uses an explicit process model to compute process inputs so as to optimize future plant behavior over a time interval known as the prediction horizon [4]. The system then implements the first control of the optimal control sequences and repeats the above procedures using the concept of 'receding horizon'. The employment of receding horizon control, however, does not inherently guarantee the closed-loop stability, especially when constraints on input and output are considered. Besides the stability, the robustness of the MPC algorithm to model uncertainty is an important issue, since MPC model is only an approximation of the real process [5, 6].

Robust constrained MPC has been studied extensively [5]. To guarantee robust stability, additional stability constraints must be imposed. For a finite input horizon, the strategy is to use the previous optimal input sequence at time k as a feasible input sequence at time $k+1$, and force the feasible cost at time $k+1$ less than the optimal cost at time k in the presence of model uncertainty [7, 8]. For an infinite input and output horizon, a state feedback control law must be adopted to facilitate a finite dimensional formulation [9]. At each sampling time, an optimal upper bound on the worst case performance cost over the infinite horizon is obtained by forcing a quadratic function of the state to decrease at each prediction time by at least the amount of the worst case performance cost at that prediction time. If the optimal feedback law computed at time k is applied at time $k+1$, the feasible upper bound at time $k+1$ must be less than the optimal upper bound at time k [9]. This approach was extended to Linear Parameter Varying (LPV) systems, in which the current model is known exactly, but the future models are uncertain [10].

For both finite and infinite horizon MPC, to guarantee robust asymptotic stability of the closed-loop system, the same optimality argument is used, that is the optimal cost at time $k+1$ must be less than or equal to the feasible cost at time $k+1$. This argument is valid only for the case where the optimal cost is based on the true state or based on the estimated state whose error dynamics are independent of and much faster than the system dynamics [11].

For an output feedback system, MPC design involves two separate designs, namely, design of a controller assuming all state elements are available and design of an estimator to reconstruct the state given a partial measurement [4, 12]. Thus at each sampling time, MPC computes optimal control moves based on the prediction of the estimated state over a fixed time horizon. It is well known that the combination of the separately designed controller and estimator has no stability margins [13]. Specifically, for systems with polytopic uncertainty or norm-bound uncertainty, supposing a robust constrained state feedback MPC [9] and an estimator based on a nominal model are designed separately, the error dynamics of the estimator are dependent on the systems dynamics, and the estimator is no longer valid. Since the optimal cost based on the estimated state may not be the optimal cost based on the true state, monotonic decrease of the performance cost based on the true state is not guaranteed. As a result, controller designers have to analyze the robust stabilizability of the combined MPC and estimator for each specific design. But unlike $\mathcal{H}_2$ and $\mathcal{H}_\infty$ linear optimal control [14] which have an easily implementable linear feedback law, robust MPC requires on-line optimization and has no explicit off-line form of feedback control law. Furthermore, because of the nonlinearities of input and output constraints, the implicit MPC control law is nonlinear in nature. All of these factors make it very difficult to analyze the robust stability of the closed-loop system with output feedback MPC [5]. To avoid the interdependency of the controller and estimator, Zheng and Morari [15] choose the finite impulse response (FIR) model, where the input prediction in the controller is only based on the current measurement and the past input, and no estimator is involved in the control problem formulation.

On the other hand, rich theory exists for robust output feedback control of linear systems, e.g. $\mathcal{H}_\infty$ theory [16]. There have been efforts to extend $\mathcal{H}_\infty$ synthesis to handle input saturations [17]. But unlike MPC, $\mathcal{H}_\infty$ control aims to obtain a linear feedback control law and an estimator off-line. Input saturation nonlinearities have to be formulated as polytopic model uncertainty [17], which leads to conservatism in handling input constraints. Moreover, the robust constrained output feedback controller synthesis has to be formulated into bilinear matrix inequalities, which are generally nonconvex with numerical properties usually hard to characterize [17].

From the preceding review, we see that the on-line determination of control actions makes it possible to handle input and output constraints explicitly, while the existence of an off-line control law makes it possible to analyze robust stability of the output feedback system. Current MPC algorithm and $\mathcal{H}_\infty$ theory possess one property but lack the other. In this thesis, we present a technique for robust constrained output feedback MPC synthesis that enjoys both properties. This scheme consists of an off-line part and an on-line part. Off-line, we design a sequence of state feedback laws and a state estimator that satisfy a robust stability criterion for the closed-loop system. On-line, a specific control law is determined from the sequence of state feedback laws based on the current estimated state with explicit satisfaction of the input and output constraints. The advantage of this design procedure is that the bilinear matrix inequality problem [17] can be reduced to a robust constrained state feedback MPC synthesis - an LMI minimization problem [9] and a robust stability test of the closed-loop system with the combined off-line control laws and estimator - an LMI feasibility problem, both of which are convex and can be solved effectively in polynomial time [18, 19].

Moreover, state feedback RCMPC has been studied extensively in [9, 20, 21]. The main idea is to design a state-dependent state feedback control law that maintains the state vector inside invariant feasible sets. However, since the invariant set constructed to guarantee stability is derived by using a single quadratic Lyapunov function, this method may lead to conservative results. In order to reduce the conservatism, Cuzzola et al. come up with an improved approach for RCMPC synthesis that uses several Lyapunov functions each one corresponding to a different vertex of the uncertainty polytope [22]. At the same time, they impose constraints on control input and output despite the use of many Lyapunov functions. Unfortunately, their method turns out to be applicable only for uncertain time-invariant systems [23].

In this thesis, we extend the results in [22, 23] to design a RCMPC algorithm for uncertain time-varying systems. The control algorithm employs a parameter-dependent Lyapunov function, so it yields less conservative results than the algorithm using a single Lyapunov function [9]. In particular, we will show via numerical examples that the developed algorithm is effective because it provides not only improved performance but also a robustness guarantee for a wider range of uncertain time-varying parameters.

1.3 Objectives

The primary objective of this thesis is to synthesize a robust model predictive controller for linear time-varying systems in the presence of model uncertainty and constraints on the input and output. The control technique uses a parameter-dependent Lyapunov function that corresponds to the vertices of polytopic uncertainty set. The proposed method not only guarantees robust stability of the system but also improves the system performance compared to the technique which employs a single Lyapunov function. These controllers are designed and applied to several illustrative examples.

1.4 Scope of Thesis

1. The thesis deals with uncertain linear time-varying systems subject to input and output constraints. The objective function is defined as an infinite horizon linear quadratic objective.

2. The design problem is solved within the framework of Robust Constrained Model Predictive Control with both control structures, namely state feedback and output feedback.

3. The proposed technique is applicable to a broad class of uncertain systems, from mechanical systems to chemical processes, from slow and small-scale systems to fast and large-scale processes.

4. The computational tool used in the thesis is the MATLAB LMI solver.

1.5 Methodology

1. Study related literature on RCMPC including model of uncertain systems, constraints, robust synthesis and robust analysis.

2. Formulate the design problem into convex optimization problems involving Linear Matrix Inequalities.

3. To improve the system performance, the control method uses a parameter-dependent Lyapunov function instead of a single Lyapunov function.

4. Develop a computer program for implementing RCMPC algorithms.

5. Compare the results with existing control methods.

1.6 Contributions

The expected contributions from this thesis are:

1. A method for designing and implementing RCMPC with improved performance and less conservative results

2. An off-line approach that reduces on-line computational burden and gives capability to carry out performance analysis to study the closed-loop behavior.

3. A computational tool for RCMPC used for several applications.

Chapter 2

BASIC KNOWLEDGE

A theory with some important tools and concepts is presented in this chapter. Section 2.1 introduces the principle of predictive controllers. Section 2.2 explains briefly some robustness issues of Model Predictive Control. Two model representations for uncertain systems are described in section 2.3. Section 2.4 presents the Linear Matrix Inequalities theory, which will be used frequently in the next chapters. Finally, the Lyapunov stability for discrete systems is given in section 2.5. This will help to analyze and understand the results presented in the thesis.

2.1 Model Predictive Control

Model Predictive Control (MPC) is formulated as the repeated solution of an open-loop optimal control problem subject to system dynamics and input and output constraints. Fig. 2.1 depicts the basic idea behind MPC. Based on measurements obtained at time k, the controller predicts the dynamic behavior of the system over a prediction horizon N_p in the future and determines (over a control horizon $N_m \leq N_p$) the input such that a predetermined open-loop performance objective in minimized. If there was no disturbances and no plant-model mismatch, and if the optimization problem could be solved over an infinite horizon, then the input signal found at $k = 0$ could be open-loop applied to the system for all $k \geq 0$. However, due to disturbances, plant-model mismatch and the finite prediction horizon, the actual system behavior is different from the predicted one. To incorporate feedback, the optimal open-loop input is implemented only until the next sampling time. Using the new system state at time $k+1$, the whole procedure - prediction and optimization - is repeated, moving the control and prediction horizon forward.

2.1.1 Objective Function

This thesis will deal with the case of infinite control and prediction horizon (i.e. $N_u = N_p = \infty$). The control objective will be, therefore, to minimize an infinite horizon linear quadratic (LQ) cost function

$$J_{LQ}(k) = \sum_{i=0}^{\infty} x^T(k+i|k)\mathcal{Q}x(k+i|k) + u^T(k+i|k)\mathcal{R}u(k+i|k), \tag{2.1}$$

where $x(k+i|k)$ denotes the state at time $k+i$, predicted based on the measurements at time k and $u(k+i|k)$ denotes the control input at time $k+i$, computed by minimizing (2.1) at time k. $\mathcal{Q}$ and $\mathcal{R}$ are symmetric, positive-definite matrices denoting suitable weighting matrices.

It is well known that the infinite approach can guarantee nominal stability of the closed-loop system.

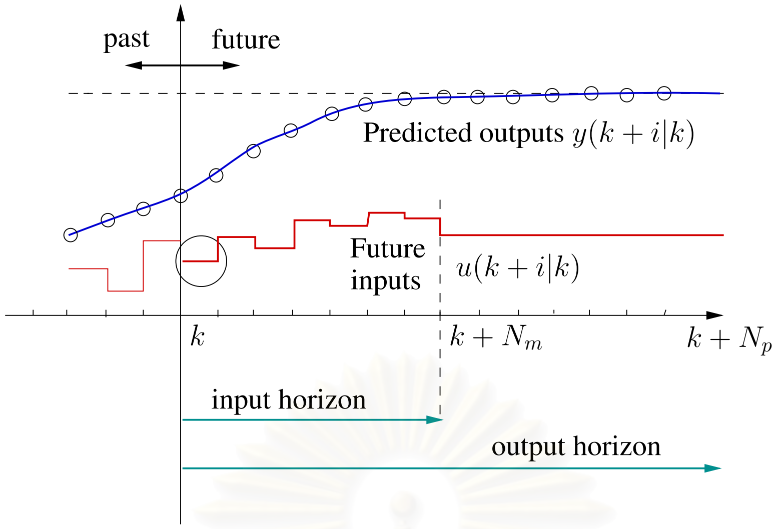


Figure 2.1: MPC scheme.

2.1.2 Constraints

In practice, many control systems are subject to constraints. Physical limitations of the process (e.g. valve saturation) and operational conditions (e.g. safety limits) impose the restrictions on control input and plant output. While output constraints can usually be relaxed or 'soften', input constraints are always regarded as 'hard' boundaries which can never be violated. The ability to handle constraints is the most contributed to the success of MPC in industry.

Consider Euclidean norm bounds on the input $u(k+i|k)$ and output $y(k+i|k)$, given respectively as

$$\|u(k+i|k)\|_2 \leq u_{\max}, \quad k \geq 0, i \geq 0, \tag{2.2}$$

$$\|y(k+i|k)\|_2 \leq y_{\max}, \quad k \geq 0, i \geq 1. \tag{2.3}$$

Here, $y(k+i|k)$ denotes the output at time $k+i$, predicted based on the system state at time k. The output constraints have been imposed strictly over the future horizon (i.e. $i \geq 1$) and not at the current time (i.e. $i = 0$). This is because the current output cannot be influenced by the current or future control action, and hence, imposing any constraints on y at the current time is meaningless. Note that u and y are vectors with n_u and n_y elements respectively.

This thesis will mainly deal with such constraints. The strategy is to directly include these constraints into an optimization problem as LMI conditions. Moreover, the formulation can also be extended to the case of componentwise peak bound on the input and output, given respectively as

$$|u_r(k+i|k)| \leq u_{r,\max}, \quad k \geq 0, i \geq 0, \quad r = 1, 2, \ldots, n_u, \tag{2.4}$$

$$|y_r(k+i|k)| \leq y_{r,\max}, \quad k \geq 0, i \geq 1, \quad r = 1, 2, \ldots, n_y. \tag{2.5}$$

2.2 Robust Model Predictive Control

When we say a control system is robust, we mean that stability is maintained and that the performance specifications are met for a specified range of model variations (uncertainty range). To be meaningful, any statement about 'robustness' of a particular control algorithm must make reference to a specific uncertainty range as well as specific stability and performance criteria. Although a rich theory has been developed for the robust control of linear systems, very little is known about the robust control of linear systems with constraints.

2.2.1 Robust Stability

The minimum closed-loop requirements is robust stability. In MPC, various design procedures achieve robust stability in two different ways. The first of these is specifying the performance objective and uncertainty description in such a way that the optimal control computations lead to robust stability. The second approach is enforcing a type of a robust contraction constraint which guarantees that the state will shrink for all plants in the uncertainty set.

2.2.2 Robust Performance

In the main stream of robust control literature, 'robust performance' is measured by determining the worst performance over the specified uncertainty range. In direct extension of this definition, it is natural to set up a new 'robust' MPC objective where the control action is selected to minimize the worst-case value of the objective function. Many attempts have been made to synthesize such a robust MPC, but they all had more or less drawbacks in terms of addressing robust stability or on-line implementation. For more details on this topic, the reader is referred to [5, 24].

This thesis provides a method to calculate a state feedback control law that minimizes an upper bound on the robust performance and by using Lyapunov arguments, guarantees the robust stability. For fairly general uncertainty descriptions, the optimization problem can be expressed as a set of LMIs for which efficient solution techniques exist. The idea behind the thesis is based on the work by Kothare et al. [9].

2.3 Models for Uncertain Systems

Two paradigms of uncertain systems that are commonly encountered in robust control will be discussed in this section. The first one is a *polytopic* uncertain model, and the second is a *norm-bound* uncertain model. These paradigms arise from two different modeling and identification procedures.

2.3.1 Polytopic Uncertain Model

Consider a discrete linear time-varying (LTV) model representation

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= Cx(k), \\ \begin{bmatrix} A(k) & B(k) \end{bmatrix} &\in \mathbf{\Omega}, \end{aligned} \tag{2.6}$$

where $x(k) \in \mathbf{R}^n$, $u(k) \in \mathbf{R}^{n_u}$, $y(k) \in \mathbf{R}^{n_y}$ are the state, the input and the output, respectively. $\mathbf{\Omega}$ is the polytopic uncertainty defined by

$$\mathbf{\Omega} = \mathbf{Co}\{\begin{bmatrix} A_1 & B_1 \end{bmatrix}, \begin{bmatrix} A_2 & B_2 \end{bmatrix}, \ldots, \begin{bmatrix} A_L & B_L \end{bmatrix}\} \tag{2.7}$$

where **Co** denotes the convex hull (see Fig. 2.2a). In other words, if $\begin{bmatrix} A(k) & B(k) \end{bmatrix} \in \mathbf{\Omega}$, then for some nonnegative $\lambda_1(k), \lambda_2(k), \ldots, \lambda_L(k)$ summing to one, we have

$$\begin{bmatrix} A(k) & B(k) \end{bmatrix} = \sum_{j=1}^{L} \lambda_j(k) \begin{bmatrix} A_j & B_j \end{bmatrix}.$$

$L = 1$ corresponds to the nominal linear time-invariant (LTI) system.

Polytopic uncertain models can be developed as follows. Suppose that for the (possibly nonlinear) system under consideration, we have input/output data set at different operating points, or at different times. From each data set, we develop a number of linear models (for simplicity, we assume that the various linear models involve the same state vector). Then this is reasonable to assume that analysis and design methods for the polytopic system (2.6)- (2.7) with vertices given by the linear model will apply to the real system.

Alternatively, suppose the Jacobian $\begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial u} \end{bmatrix}$ of a nonlinear discrete time-varying system $x(k+1) = f(x(k), u(k), k)$ is known to lie in the polytope $\mathbf{\Omega}$. Then it can be shown that every trajectory (x, u) of the original nonlinear system is also a trajectory of (2.6) for some LTV system in $\mathbf{\Omega}$ [25]. Thus the original nonlinear system can be approximated (possibly conservatively) by a polytopic uncertain LTV system. Similarly, it can be shown that bounds on impulse response coefficients of SISO FIR plants can be translated to a polytopic uncertainty description on the state-space matrices. Thus this polytopic uncertainty description is suitable for several engineering problems.

2.3.2 Norm-Bound Uncertain Model

This paradigm consists of an LTI model with uncertainties or perturbations appearing in the feedback loop (see Fig. 2.2b)

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_p p(k), \\ y(k) &= Cx(k), \\ q(k) &= C_q x(k) + D_{qu} u(k), \\ p(k) &= (\Delta q)(k). \end{aligned} \tag{2.8}$$

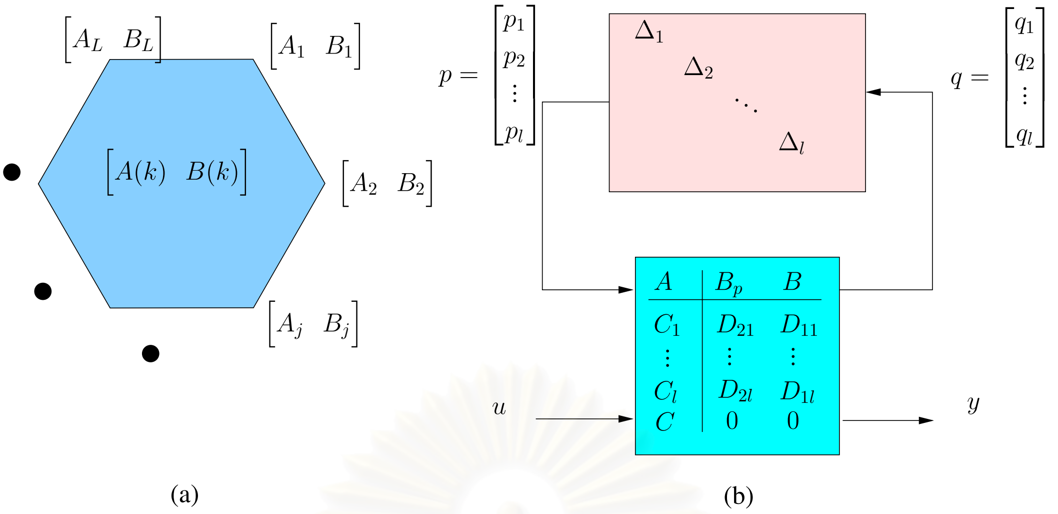


Figure 2.2: (a) Graphical representation of polytopic uncertainty. (b) Structured uncertainty.

The operator Δ is block-diagonal:

$$\Delta = \begin{bmatrix} \Delta_1 & & & \\ & \Delta_2 & & \\ & & \ddots & \\ & & & \Delta_l \end{bmatrix}, \tag{2.9}$$

with $\Delta_i : \mathbf{R}^{n_i} \rightarrow \mathbf{R}^{n_i}$. Δ can represent either a memoryless time-varying matrix with $\|\Delta_i(k)\|_2 \equiv \bar{\sigma}(\Delta_i(k)) \leq 1$ for $i = 1, 2, \ldots, l,\ k \geq 0$, or a convolution operator (e.g. a stable LTI dynamical system), with the operator norm induced by the truncated ℓ_2-norm less than 1, i.e.

$$\sum_{j=0}^{k} p_i(j)^T p_i(j) \leq \sum_{j=0}^{k} q_i(j)^T q_i(j), \quad i = 1, \ldots, l, \quad k \geq 0.$$

Each Δ_i is assumed to be either a *repeated scalar* block or a *full* block, and models a number of factors, such as nonlinearities, dynamics or parameters, that are unknown, unmodeled or neglected. A number of control systems with uncertainties can be cast in this framework.

Consider the LTV case, the system (2.8) then corresponds to the system (2.6) with

$$\mathbf{\Omega} = \{ \begin{bmatrix} A + B_p \Delta C_q & B + B_p \Delta D_{qu} \end{bmatrix} : \bar{\sigma}(\Delta_i) \leq 1 \}. \tag{2.10}$$

$\Delta(k) \equiv 0$, $p(k) \equiv 0$, $k \geq 0$ corresponds to the nominal LTI system.

2.4 Linear Matrix Inequalities

This section gives a brief introduction to linear matrix inequalities (LMIs) and some optimization problems based on LMIs. For more details, the interested reader is referred to Boyd et al. [18].

2.4.1 Definitions

A linear matrix inequality or LMI is a matrix inequality of the form

$$F(x) \triangleq F_0 + \sum_{i=1}^{m} x_i F_i > 0, \tag{2.11}$$

where $x \in \mathbf{R}^m$ is the variable and the symmetric matrices $F_i = F_i^T \in \mathbf{R}^{n\times n}$, $i = 0, \ldots, m$ are given. The inequality symbol in (2.11) means that $F(x)$ is positive definite, i.e. $u^T F(x) u > 0$ for all nonzero $u \in \mathbf{R}^n$.

We will also encounter non-strict LMIs, which have the form

$$F(x) \geq 0. \tag{2.12}$$

The inequality symbol in (2.12) means that $F(x)$ is positive semi-definite, i.e. $u^T F(x) u \geq 0$ for all $u \in \mathbf{R}^n$.

A very important property of the LMI is that a system of linear matrix inequalities $F^{(1)}(x) > 0, \ldots, F^{(p)}(x) > 0$ can be expressed as the single LMI

$$F(x) = \mathbf{diag}(F^{(1)}(x), \ldots, F^{(p)}(x)) > 0.$$

This has an important implication. If one defines a problem concerning a numerous LMIs, one does not need to reformulate the problem if additional LMIs need to be implemented. A new bigger LMI can be formulated, as a combination matrix of all the LMIs.

The linear matrix inequalities are useful since many problems in control, especially in the context of this thesis, can be formulated in (or reformulated to) this form. Then the problem can be solved in an efficient and reliable way. The LMIs defines a 'convex constraint' and problems involving the minimization or maximization of an 'affine' function subject to LMI constraints belong to a class of convex optimization problems. Therefore, the full power of convex optimization theory can be employed.

Another important property of the LMI is that nonlinear (convex) inequalities can be converted to LMI using Schur complements.

Theorem 2.1 (Schur complements)
Let $Q(x) = Q(x)^T$, $R(x) = R(x)^T$, and $S(x)$ depend affinely on x. Then the LMI

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0 \tag{2.13}$$

is equivalent to the matrix inequalities

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

or equivalently,

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

We often encounter problems in which the variables are matrices, for example the constraint $P > 0$, where the entries of P are the optimization variables. In such cases, we shall not write out the LMI explicitly in the form $F(x) > 0$, but instead make clear which matrices are the variables.

2.4.2 Solving the LMIs

The two most used algorithms to solve LMI problems are the Ellipsoid method and the Interior Point method. The former is very simple and not always as effective as the interior point, but can detect infeasible point easily. The Ellipsoid starts with an ellipsoid $\mathcal{E}^{(0)}$ that is guaranteed to contain an optimal point. If such point does not exist, the problem is infeasible and the algorithm stops. If the point exists, then a *cutting plane* is computed, which slices the ellipsoid to two parts. The optimal point belongs to one of the halves. In the next iteration of the algorithm, another ellipsoid $\mathcal{E}^{(1)}$ that contains this half is computed. $\mathcal{E}^{(1)}$ is then guaranteed to contain an optimal point. The process is then repeated.

The Interior Point method has been developed recently and is much more effective than the ellipsoid method. The main idea is to replace the constrained optimization problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && F(x) > 0, \end{aligned}$$

with the unconstrained optimization problem

$$F_t(x) := tf(x) + \phi(x),$$

where $t > 0$ is the penalty parameter, $f(x)$ is affine and $\phi(x)$ is a barrier function. The main idea is to determine the minimizer $x(t)$ of F_t and to consider the behavior of $x(t)$ as a function of the penalty parameter $t > 0$.

The MATLAB *LMI toolbox* has ready packages for solving the feasibility problem and solving the linear objective optimization problem subject to a set of LMIs, i.e.

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && F(x) > 0, \end{aligned} \tag{2.14}$$

where c is a real vector of appropriate size and F is a symmetric matrix. Since the LMI set is convex, the problem is then convex and a global solution is found within some prespecified accuracy. For more details about the MATLAB toolbox, the reader is referred to Gahinet et al. [26].

LMI-based problem is the central importance to this thesis. LMI problem can be solved in polynomial time, which means that it has low computational complexity, and is, therefore essential for MPC algorithm.

2.5 Lyapunov Theory for Discrete-Time Systems

In this section, the Lyapunov stability theorem for discrete-time systems is reviewed. For discrete-time systems, we use the forward difference

$$\Delta V(x(k)) = V(x(k+1)) - V(x(k)).$$

The next theory gives the conditions necessary for a discrete-time system to be stable.

Theorem 2.2 (Stability of discrete-time systems)
Consider the discrete-time system

$$x(k+1) = f(x(k)),$$

where $x \in \mathbf{R}^n$ *and* $f(x) \in \mathbf{R}^n$ *with property that* $f(0) = 0$. *Suppose there exists a scalar function* $V(x)$ *continuous in* x *such that*

- $V(x) > 0 \quad \forall x \neq 0$
- $\Delta V(x) < 0 \quad \forall x \neq 0$
- $V(0) = 0$

locally in region $\mathcal{D}$, *then the equilibrium state* $x = 0$ *is asymptotically stable and* $V(x)$ *is a Lyapunov function.*

For uncertain system (2.6) with polytopic uncertainty (2.7), a natural approach is to check the existence of a Lyapunov function that depends not only on the system state but also on the uncertain parameter. Such a function is called a parameter-dependent Lyapunov function (PDLF). If we denote $\lambda = [\lambda_1 \; \lambda_2 \; \ldots \; \lambda_L]^T$, we can state the following definition.

Definition 2.1 (Robust stability of discrete-time systems)
The uncertain discrete-time system (2.6) is said to be stable if there exists a Lyapunov function

$$V(x(k), \lambda(k)) = x(k)^T P(\lambda(k)) x(k), \quad P(\lambda(k)) = P(\lambda(k))^T > 0, \tag{2.15}$$

such that $\Delta V(x(k), \lambda(k)) < 0$ *for all none-zero* $x(k) \in \mathbf{R}^n$ *and admissible uncertain parameter* $\lambda(k)$. *Similarly, the uncertain discrete-time system (2.6) is said to be robustly stabilizable if there exists a state-feedback control law* $u(k) = Fx(k)$ *such that the resulting closed-loop system is robustly stable for all admissible uncertain parameter* $\lambda(k)$.

In fact, there is no general and systematic way to formally determine $P(\cdot)$ as a function of the uncertain parameter $\lambda(k)$. A traditional way of addressing this problem is to look for a single Lyapunov matrix $P(\cdot) = P$ which renders condition (2.15) satisfied. This constitutes one of the first results in the quadratic approach, that is the stability assessment over the compact set (2.7) can be determined by testing the discrete, enumerable and bounded set of the vertices of the polytope (2.7). Furthermore, a single matrix P that satisfies the condition given in Definition 2.1 can be found by using efficient LMI tools. The quadratic stability, however, is somewhat conservative.

In the attempt to reduce the conservatism, some nice results have been proposed for the construction of PDLF in the discrete-time case. The main advantage of the ‘new’ stability condition, beside the fact that it leads to less conservative results than the quadratic approach, consists in the introduction of an extra degree of freedom which allows to get a control law without an explicit dependence on the Lyapunov matrices [23, 27, 28].

The proposed RCPMC technique in this thesis will make use of such a PDLF. Specifically, it is based on the sufficient LMI condition given in [23].

Theorem 2.3 (New robust stability condition)
The uncertain discrete-time system (2.6) is robustly stabilizable if there exist L symmetric matrices Q_j with $j = 1, 2, \ldots, L$ and a pair of matrices $\{Y,\ G\}$ satisfying the following LMIs.

$$\begin{bmatrix} G + G^T - Q_j & G^T A_j^T + Y^T B_j^T \\ A_j G + B_j Y & Q_l \end{bmatrix} > 0, \quad \forall j = 1, 2, \ldots, L, \quad l = 1, 2, \ldots, L. \qquad (2.16)$$

Furthermore, the state feedback matrix is given by

$$F = Y G^{-1}. \qquad (2.17)$$

Proof See [23]. ■

The above theorem bases on Definition 2.1 to search for a state feedback law that robustly stabilizes the closed-loop system with a parameter-dependent Lyapunov matrix $P(\lambda(k)) = \sum_{j=1}^{L} \lambda_j(k) P_j$ and $P_j = Q_j^{-1}$. It is interesting to notice that, in contrast with the quadratic stability synthesis, the determination of the control (2.17) does not directly depend on the Lyapunov matrices P_j which are used to build the parameter-dependent Lyapunov matrix $P(\lambda(k))$.

Chapter 3

RCMPC WITH A SINGLE LYAPUNOV FUNCTION

This chapter presents the first strategy of robust constrained Model Predictive Control applicable to both polytopic and norm-bound uncertain systems. Section 3.1 discusses about RCMPC using state feedback law. The algorithm is then modified to obtain an off-line state feedback RCMPC in order to lower the on-line computational complexities. Section 3.3 offers an output feedback scheme that utilizes the above off-line state feedback together with an estimator. The knowledge in this chapter will be used later to develop another strategy which employs a parameter-dependent Lyapunov function, and will be used as a comparison tool.

3.1 State Feedback RCMPC with a SLF

This section begins with a state feedback robust MPC problem without input and output constraints and reduces it to a linear objective minimization problem. Then input and output restrictions are incorporated into the optimization as additional constraints. The resulting control law is shown to robustly stabilize the LTV system among the uncertain plants $\mathbf{\Omega}$.

3.1.1 Unconstrained Case

The system is described by (2.6) with the associated uncertainty set (2.7) or (2.10). The system state $x(k)$ is assumed to be measurable. As mentioned in section 2.2, the minimization of the nominal objective function (2.1) at each sampling k is replaced by the minimization of a robust performance objective as follows.

$$\min_{u(k+i|k),\ i\geq 0} J_{WC}(k), \tag{3.1}$$

where

$$J_{WC}(k) = \max_{[A(k+i)\quad B(k+i)]\in\mathbf{\Omega},\ i\geq 0} J_{LQ}(k). \tag{3.2}$$

This is a min-max problem. The maximization in (3.2) is taken over the set $\mathbf{\Omega}$, and corresponds to choosing that time-varying plant $\begin{bmatrix} A(k+i) & B(k+i) \end{bmatrix} \in \mathbf{\Omega},\ i \geq 0$, which, if uses as a model for predictions, would lead to the largest or worst-case value of $J_{LQ}(k)$ among all plants in $\mathbf{\Omega}$. This worst-case value is minimized over present and future control moves $u(k+i|k),\ i \geq 0$. We then address problem (3.1) by deriving an upper bound on the robust performance and minimizing this upper bound with a constant state feedback control law $u(k+i|k) = Fx(k+i|k),\ i \geq 0$.

Derivation of the upper bound

At sampling time k, we define a quadratic function

$$V(x) = x^T P x, \quad P > 0.$$

For any $\begin{bmatrix} A(k+i) & B(k+i) \end{bmatrix} \in \mathbf{\Omega},\ i \geq 0$, suppose $V(x)$ satisfy the following robust stability constraint

$$V(x(k+i+1|k)) - V(x(k+i|k)) \leq -[x(k+i|k)^T \boldsymbol{\mathcal{Q}} x(k+i|k) + u(k+i|k)^T \boldsymbol{\mathcal{R}} u(k+i|k)]. \quad (3.3)$$

Summing (3.3) from $i = 0$ to $i = \infty$ and requiring $x(\infty|k) = 0$ or $V(x(\infty|k)) = 0$, it follows that

$$J_{WC}(k) \leq V(x(k|k)) \leq \gamma. \quad (3.4)$$

Here, γ is an upper bound of $V(x(k|k))$. Thus, minimizing $J_{WC}(k)$ might be obtained by minimizing this upper bound. The goal of the robust MPC algorithm has, therefore been redefined to synthesize, at each sampling time k, a state feedback control law $u(k+i|k) = Fx(k+i|k),\ i \geq 0$ so as to minimize γ. As in standard MPC, only the first computed input $u(k|k) = Fx(k|k)$ is implemented. At the next sampling time, the state $x(k+1)$ is measured and the optimization is repeated to recompute F.

Let $Q = \gamma P^{-1}$. The following theorem gives conditions for the existence of the matrix P satisfying (3.3) and the corresponding state feedback matrix F.

Theorem 3.1 (State feedback robust unconstrained MPC)
Let $x(k) = x(k|k)$ be the state of the uncertain system (2.6) measured at sampling time k. Assume that there are no constraints on the control input and plant output.

(a) *Suppose that the uncertainty set is defined by a polytope as in (2.7). 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 γ on the robust performance objective function at sampling time k is given by*

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

where $Q > 0$ and Y are obtained from the solution of the following linear objective minimization problem.

$$\min_{\gamma, Q, Y} \quad \gamma$$

subject to

$$\begin{bmatrix} 1 & x(k|k)^T \\ x(k|k) & Q \end{bmatrix} \geq 0 \quad (3.5)$$

and

$$\begin{bmatrix} Q & QA_j^T + Y^T B_j^T & Q\boldsymbol{\mathcal{Q}}^{1/2} & Y^T \boldsymbol{\mathcal{R}}^{1/2} \\ A_j Q + B_j Y & Q & 0 & 0 \\ \boldsymbol{\mathcal{Q}}^{1/2} Q & 0 & \gamma I & 0 \\ \boldsymbol{\mathcal{R}}^{1/2} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \quad (3.6)$$

$$j = 1, 2, \ldots, L.$$

(b) *Suppose that the uncertainty set is defined by a norm-bound perturbation Δ as in (2.10). In this case, F is given by*

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

where $Q > 0$ and Y are obtained from the solution of the following linear objective minimization problem:

$$\min_{\gamma,Q,Y,\Lambda} \quad \gamma$$

subject to

$$\begin{bmatrix} 1 & x(k|k)^T \\ x(k|k) & Q \end{bmatrix} \geq 0 \tag{3.7}$$

and

$$\begin{bmatrix} Q & QA^T + Y^T B^T & QC_q^T + Y^T D_{qu}^T & Q\boldsymbol{\mathcal{Q}}^{1/2} & Y^T \boldsymbol{\mathcal{R}}^{1/2} \\ AQ + BY & Q - B_p \Lambda B_p^T & 0 & 0 & 0 \\ C_q Q + D_{qu} Y & 0 & \Lambda & 0 & 0 \\ \boldsymbol{\mathcal{Q}}^{1/2} Q & 0 & 0 & \gamma I & 0 \\ \boldsymbol{\mathcal{R}}^{1/2} Y & 0 & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \tag{3.8}$$

*where $\Lambda =$ **diag** $[\lambda_1 I_{n_1}, \lambda_2 I_{n_2}, \ldots, \lambda_l I_{n_l}] > 0$.*

The theorem was proposed in the work by Kothare et al. [9] and its proof is not repeated here. Notice that the formulation in Theorem 3.1 is an LMI optimization problem. Note also that the variables in this problem should be strictly written as $Q(k), Y(k), F(k)$ etc. to emphasize that they are computed at time k. For notational convenience, we omit the time index here, we will, however, utilize this notation in next sections.

3.1.2 State Feedback RCMPC and Stability

In industry, many processes are subject to constraints on the control input and plant output. Any system possessing physical or operational limitations should be treated with care. Furthermore, the explicit handling of constraints may allow the process to operate closer to constraints and optimal operating conditions. In LMI framework, input and output constraints are formulated as follows.

Input constraints

In this case, both polytopic and norm-bound uncertain models have the same formula. For the Euclidean norm constraint (2.2), it is proved in [9] that at each sampling time k, $\|u(k+i|k)\|_2 \leq u_{\max}$, $i \geq 0$, if

$$\begin{bmatrix} u_{\max}^2 I & Y \\ Y^T & Q \end{bmatrix} \geq 0. \tag{3.9}$$

This is a *sufficient* LMI condition with variables Y and Q.

Output constraints

Performance specifications impose constraints on the process output $y(k)$. As in input constraint case, we derive sufficient LMI conditions for both the uncertainty descriptions (2.7) and (2.10) that render the output constraints satisfied.

At each sampling time k, consider the Euclidean norm constraint

$$\max_{[A(k+i)\ \ B(k+i)]\in\mathbf{\Omega},\ i\geq 0} \|y(k+i|k)\|_2 \leq y_{\max}, \quad i \geq 1. \tag{3.10}$$

This is the worst-case constraint over the set $\mathbf{\Omega}$, and is imposed strictly over the future prediction horizon ($i \geq 1$).

In case of polytopic uncertainty, the set $\mathbf{\Omega}$ is given by (2.7). The constraint (3.10) is satisfied if

$$\begin{bmatrix} y_{\max}^2 I & C(A_jQ+B_jY) \\ (A_jQ+B_jY^T)C^T & Q \end{bmatrix} \geq 0, \quad j = 1, 2, \ldots, L. \tag{3.11}$$

For norm-bound uncertainty case, the set $\mathbf{\Omega}$ is described by (2.10) in terms of structured Δ block. The constraint (3.10) is satisfied if

$$\begin{bmatrix} y_{\max}^2 Q & (C_qQ+D_{qu}Y)^T & (AQ+BY)^TC^T \\ C_qQ+D_{qu}Y & T^{-1} & 0 \\ C(AQ+BY) & 0 & I-CB_pT^{-1}B_p^TC^T \end{bmatrix} \geq 0, \tag{3.12}$$

where $T = \text{diag}[t_1I_{n_1}, t_2I_{n_2}, \ldots, t_rI_{n_r}] > 0$. The proofs of these conditions are given in [9].

Theorem 3.2 (State feedback RCMPC)

Let $x(k) = x(k|k)$ be the state of the uncertain system (2.6) measured at sampling time k.

(a) *Suppose that the uncertainty set is defined by a polytope as in (2.7). Then the state-feedback matrix $F(k)$ in the control law $u(k+i|k) = F(k)x(k+i|k)$ for $k, i \geq 0$ that minimizes the upper bound γ on the robust performance objective function at sampling time k and satisfies a set of specified input and output constraints is given by*

$$F(k) = YQ^{-1},$$

where $Q > 0$ and Y are obtained from the solution of the following linear objective minimization problem

$$\min_{\gamma,Q,Y} \quad \gamma \qquad \text{subject to (3.5), (3.6), (3.9), and (3.11).} \tag{3.13}$$

(b) *Suppose that the uncertainty set is defined by a norm-bound perturbation Δ as in (2.10). In this case, $F(k)$ is given by*

$$F(k) = YQ^{-1},$$

where $Q > 0$ and Y are obtained from the solution of the following linear objective minimization problem

$$\min_{\gamma,Q,Y,\Lambda,T} \quad \gamma \qquad \text{subject to (3.7), (3.8), (3.9), and (3.12).} \tag{3.14}$$

Furthermore, the feasible receding horizon state feedback control law robustly asymptotically stabilizes the closed-loop system.

Proof See [9]. ■

As mentioned in Kothare et al. [9], a single Lyapunov function is used to guarantee the robust stability of the closed-loop system. This Lyapunov function is $V(k|k) = x(k|k)^T P x(k|k)$ where the matrix $P > 0$ is obtained by the optimal solution of the optimization problem (3.13) or (3.14).

We are now ready to come up with the algorithm for the implementation of state feedback RCMPC. Since at each sampling time k, the control law is obtained by solving an optimization problem on-line, we then refer to it as *on-line state feedback RCMPC* and use the time index for every parameter and variable. Later, in section 3.2, we will modify this problem to obtain another version in which the optimization problem is solved *off-line*.

Algorithm 3.1 (On-line state feedback RCMPC with a SLF)

1. *Get the new state* $x(k|k) = x(k)$
2. *Solve the optimization problem (3.13) or (3.14) and obtain* $F(k) = Y(k)Q(k)^{-1}$
3. *Apply only* $u(k) = F(k)x(k)$
4. *Set* $k := k + 1$ *and go to 1.*

3.2 Off-Line State Feedback RCMPC with a SLF

The requirement of optimality leads to high on-line computation and limits the application of MPC to relatively slow dynamics and small-scale processes. Moreover, when MPC incorporates explicit model uncertainty and constraints on the input and output, the resulting on-line computation is much higher. In the optimization (3.13), for example, the computational demand will grow significantly with the number of vertices of the polytopic uncertainty set which itself grows exponentially with the number of independent uncertain process parameters.

This section reviews the off-line formulation of RCMPC proposed in [21]. The algorithm gives a sequence of explicit control law corresponding to a sequence of asymptotically stable invariant ellipsoids constructed one inside another in state space. Both polytopic uncertain systems and norm-bound uncertain systems are applicable. With the off-line approach, the computation for on-line implementation is reduced significantly with minor loss in performance, thereby potentially facilitating the application of RCMPC in fast processes and large scale systems.

3.2.1 Asymptotically Stable Invariant Ellipsoid

Definition 3.1 (Asymptotically stable invariant ellipsoid)
Given a discrete dynamical system $x(k+1) = f(x(k))$, *a subset* $\mathcal{E} = \{x \in \mathbf{R}^n | x^T Q^{-1} x \leq 1\}$ *of the state space* $\mathbf{R}^n$ *is said to be an asymptotically stable invariant ellipsoid, if it has the property that, whenever* $x(k_1) \in \mathcal{E}$, *then* $x(k) \in \mathcal{E}$ *for all times* $k \geq k_1$ *and* $x(k) \to 0$ *as* $k \to \infty$.

Next, the following lemma is devoted to construct an asymptotically stable invariant ellipsoid (see Fig. 3.2.1).

Lemma 3.1 *Consider a closed-loop system composed of an uncertain plant (2.6) or (2.8) and a state feedback controller* $u(k) = YQ^{-1}x(k)$ *where* Y *and* Q *are obtained by applying the state feedback RCMPC algorithm in Theorem 3.2 to a system state* x_0*. Then the subset* $\mathcal{E} = \{x \in \mathbf{R}^n | x^T Q^{-1} x \leq 1\}$ *of the state space* $\mathbf{R}^n$ *is an asymptotically stable invariant ellipsoid.*

Proof See [20]. ■

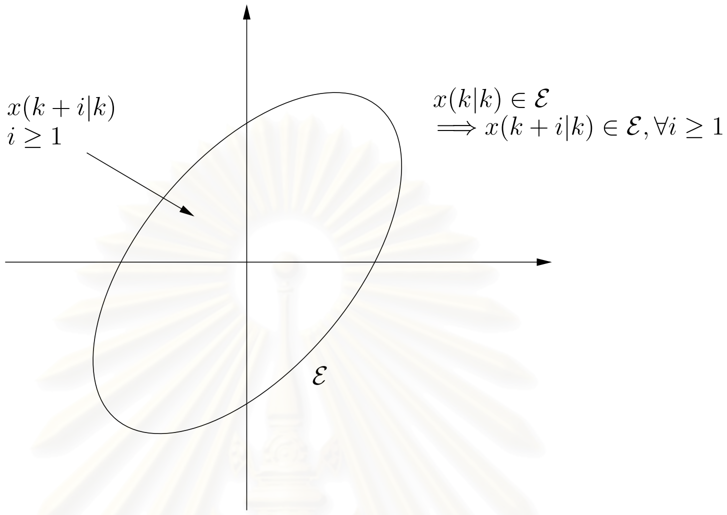


Figure 3.1: Graphical representation of the state invariant ellipsoid $\mathcal{E}$ in 2-dimensions.

3.2.2 Off-Line Strategy

The off-line approach presented in this section is based on the concept of the asymptotically stable invariant ellipsoid. Without loss of generality, we use the algorithm for polytopic uncertain systems described by (2.6) to illustrate the subsequent off-line formulation. Similar results can be obtained for norm-bound uncertain systems (2.8).

As mentioned in [21], when we apply Theorem 3.2 to a state far from the origin, due to input constraints, the resulting asymptotically stable invariant ellipsoid is large and the norm of the feedback matrix is small. This fact can be observed from simulation results in the example of an angular positioning system given in section 5.1. It is not necessary to keep this feedback matrix constant while the state is converging to the origin where there are less constraints on the choice of the feedback matrix. Within an asymptotically stable invariant ellipsoid $\mathcal{E} = \{x \in \mathbf{R}^n | x^T Q^{-1} x \leq 1\}$, we define the distance between the state x and the origin as the weighted norm $\|x\|_{Q^{-1}} \triangleq \sqrt{x^T Q^{-1} x}$. By adding asymptotically stable invariant ellipsoids one inside another, we have more freedom to adopt varying feedback matrices based on the distance between the state and the origin. We show next that we can achieve this in the off-line formulation.

Algorithm 3.2 (Off-line state feedback RCMPC)
Off-line, given an initial feasible state x_1*, generate a sequence of* γ_i, Q_i, X_i *and* $Y_i,\ i = 1, \ldots, N$*. Let* $i := 1$.

1. *Compute the minimizer γ_i, Q_i, Y_i by using Theorem 3.2 with an additional constraint $Q_{i-1} > Q_i$ (ignored at $i = 1$), store $Q_i^{-1}, F_i(= Y_i Q_i^{-1})$ in a look-up table*

2. *If $i < N$, choose a state x_{i+1} satisfying $\|x_{i+1}\|^2_{Q_i^{-1}} < 1$. Let $i := i + 1$, go to 1.*

   On-line, given an initial state $x(0)$ satisfying $\|x(0)\|^2_{Q_1^{-1}} \leq 1$, let the state be $x(k)$ at time k.

3. *Perform a bisection search over Q_i^{-1} in the look-up table to find the index i corresponding to the smallest Q_i (or equivalently, the smallest ellipsoid $\mathcal{E}_i = \{x \in \mathbf{R}^n | x^T Q_i^{-1} x \leq 1\}$) such that $\|x(k)\|^2_{Q_i^{-1}} \leq 1$.*

4. *Apply the control law $u(k) = F_i x(k)$.*

5. *Set $k := k + 1$ and go to 3.*

Note that Step 1 in the off-line part of Algorithm 3.2 is always feasible for $i > 1$, assuming that it is feasible for $i = 1$. This is because if the minimizer is $\gamma,\ Q,\ Y$ at x, then at an arbitrary $\tilde{x}$ satisfying $\|\tilde{x}\|_{Q^{-1}} < 1$, there exists a scalar $\alpha > 1$ such that $\|\alpha\tilde{x}\|_{Q^{-1}} = 1$ and $1/\alpha^2\gamma,\ 1/\alpha^2 Q,\ 1/\alpha^2 Y$ is a feasible solution with the additional constraint $Q > 1/\alpha^2 Q$.

From the on-line state feedback RCMPC in Algorithm 3.1, it can be seen that the optimal RCMPC law and the corresponding asymptotically stable invariant ellipsoid depend on the state. Although the control law can be applied to all the states within the ellipsoid, it is not necessarily optimal. So the off-line formulation sacrifices optimality somewhat while significantly reducing the on-line computational burden.

Furthermore, in each off-line optimization in Algorithm 3.2, we can minimize the performance cost based on a set of state instead of a single state. This can help in averaging the effect of the individual state on the suboptimal RCMPC law. In addition, since the RCMPC law is available off-line, performance analysis can be carried out to study the closed-loop responses, e.g. in the proof of Theorem 3.3 below. In section 3.3, this approach will be used to guarantee the robust stability criterion for the combined robust state feedback and estimator in *output feedback RCMPC* law.

Note that the sequence of state feedback matrices generated in Algorithm 3.2 is constant between two adjacent asymptotically stable invariant ellipsoids and discontinuous on the boundary of each asymptotically stable invariant ellipsoid. The following result is devoted to constructing a continuous feedback matrix over the state space.

Algorithm 3.3 (Continuous off-line state feedback RCMPC with a SLF)

Consider the lookup table generated by the off-line part of Algorithm 3.2. If for each x_i ($i = 1, \ldots, N-1$), the following condition is satisfied.

$$Q_i - (A_j + B_j F_{i+1})^T Q_i^{-1} (A_j + B_j F_{i+1}) > 0, \quad j = 1, \ldots, L. \tag{3.15}$$

Then, on-line, given an initial state $x(0)$ satisfying $\|x(0)\|^2_{Q_1^{-1}} \leq 1$, let the state be $x(k)$ at time k. Perform a bisection search over Q_i^{-1} in the look-up table to find the index i equivalent to the smallest ellipsoid $\mathcal{E} = \{x \in \mathbf{R}^n | x^T Q_i^{-1} x \leq 1\}$ such that $\|x(k)\|^2_{Q_i^{-1}} \leq 1$. If $i \neq N$, solve

$x(k)^T \left(\alpha_i Q_i^{-1} + (1-\alpha_i) Q_{i+1}^{-1}\right) x(k) = 1$ *for* α_i *and apply the control law* $u(k) = (\alpha_i F_i + (1 - \alpha_i) F_{i+1}) x(k)$*. If* $i = N$*, apply* $u(k) = F_N x(k)$*.*

The LMI minimization problem in this chapter is solved by using interior point methods which generally use a sequence of strictly convex unconstrained minimization problems to solve a convex constrained minimization problem. For a strictly convex unconstrained minimization problem, not only the objective function but also all the minimizers are unique. Hence, it is reasonable to assume that the optimal solutions for the optimizations in the off-line part of Algorithm 3.2 are unique. Under this assumption we can always find a sequence of minimizers for Algorithm 3.3, because condition (3.15) becomes trivial if x_{i+1} is chosen to be sufficiently close to x_i.

The robust stability of the control law in the above Algorithms can be stated below.

Theorem 3.3 (Robust stability of off-line RCMPC with a SLF)
Given an uncertain dynamical system (2.6) and an initial state $x(0)$ *satisfying* $\|x(0)\|^2_{Q_1^{-1}} \leq 1$*, the off-line state feedback RCMPC in Algorithm 3.2 and Algorithm 3.3 robustly asymptotically stabilizes the closed-loop system.*

Proof See [21]. ■

Up to this point, we have a critical remark. Both Algorithms 3.2 and 3.3 are a general approaches to construct a Lyapunov function for uncertain and constrained systems. The Lyapunov function is

$$V(x) = \begin{cases} x^T Q_i^{-1} x & \text{if} \quad \|x(k)\|^2_{Q_i^{-1}} \leq 1, \|x(k)\|^2_{Q_{i+1}^{-1}} \geq 1, i \neq N \\ x^T Q_N^{-1} x & \text{if} \quad \|x(k)\|^2_{Q_N^{-1}} \leq 1. \end{cases} \tag{3.16}$$

This Lyapunov function is not necessarily continuous on the boundary of each asymptotically stable invariant ellipsoid. It is enough to have $V(x)$ be monotonically decreasing within the smallest ellipsoid and within each ring region between two adjacent ellipsoids to stabilize the closed-loop system.

From both Algorithms 3.2 and 3.3, we can see that the choice of the state x_{i+1} satisfying $\|x_{i+1}\|^2_{Q_i^{-1}} < 1$ is arbitrary. For ease of implementation, Wan and Kothare [21] provide the following suggestions. We can choose an arbitrary one dimensional subspace $\mathcal{S} = \{\alpha x^{\max} | 1 \geq \alpha > 0, \alpha \in \mathbf{R}, x^{\max} \in \mathbf{R}^n\}$, where $x^{\max}$ is a state chosen to be as far from the origin as it is feasible for the problem. We can then discretize this set and construct a set of discrete points, $\mathcal{S}^d = \{\alpha_i x^{\max} | 1 \geq \alpha_1 > \cdots > \alpha_N > 0, \alpha_i \in \mathbf{R}, x^{\max} \in \mathbf{R}^n\}$. Since the asymptotically stable invariant ellipsoid constructed for each discrete point actually passes through that point, $\|\alpha_{i+1} x^{\max}\|^2_{Q_i^{-1}} < \|\alpha_i x^{\max}\|^2_{Q_i^{-1}} = 1$ is satisfied. In order to obtain a look-up table that can cover a very large portion of the state space with a limited number of discrete points, those authors suggest a discretization of the one dimensional subspace using a logarithmic scale.

3.3 Output Feedback RCMPC with a SLF

Most of the theoretical developments in the area of robust constrained MPC are based on the assumption that the full state is available for measurement. In a control environment, however, it often happens that the measurement of the state variables of the system is not practical, very costly or even impossible. The solution to this problem is the use of an state estimator. An estimator reconstructs the state variables using a dynamical model of the system, the inputs, and specific measurable outputs of the system. Based on the estimated state, the state feedback control law can then be utilized, and the resulting controller is called output feedback controller (see Fig. 3.3). Since we employ an off-line approach for the controller design which gives a sequence of explicit control laws, we are able to analyze the robust stabilizability of the combined control laws and estimator, and by adjusting the design parameters, guarantee robust stability of the closed-loop system in the presence of constraints.

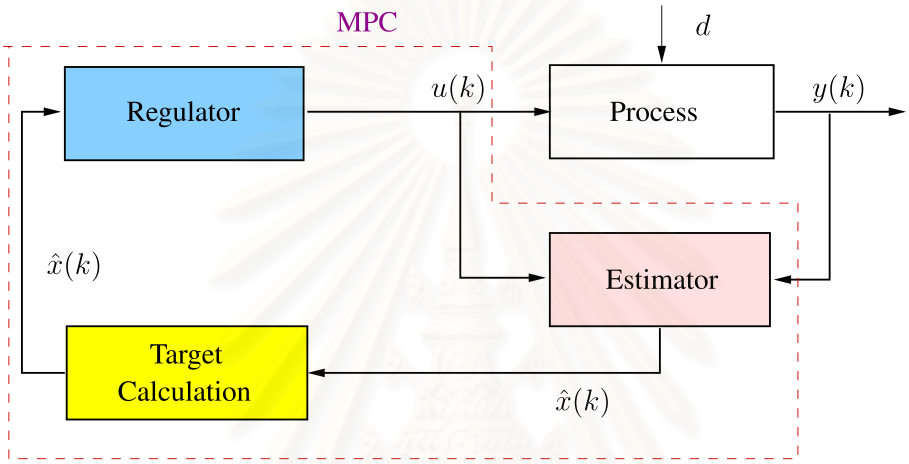


Figure 3.2: Output feedback MPC scheme.

3.3.1 Separate Design of Controller and Estimator

The state feedback controller design is the same as the off-line part of Algorithm 3.2. Based on the nominal model $[A_0 \quad B_0]$, we design a state estimator of the form

$$\begin{aligned} \hat{x}(k+1) &= A_0\hat{x}(k) + B_0u(k) + L_p(y(k) - C\hat{x}(k)), \\ \hat{x}(0) &= 0, \end{aligned} \tag{3.17}$$

where L_p is the estimator gain. The nominal model can be the plant model identified at the steady state. Assuming that as the system state converges to the steady state, the time varying plant $[A(k) \quad B(k)]$ converges to $[A_0 \quad B_0]$, then there will be no plant-model mismatch in the estimator around the steady state, and the exact state can be reconstructed using the state estimator which will speed up the feedback control performance. In general, the error dynamics of the estimator are:

$$\begin{aligned} e(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= (A_0 - L_pC)e(k) + f(x(k), u(k)), \end{aligned}$$

where $f(x(k), u(k)) = (A(k) - A_0)x(k) + (B(k) - B_0)u(k)$ for system (2.6) and $f(x(k), u(k)) = B_p\Delta C_q x(k) + B_p\Delta D_{qu}u(k)$ for system (2.8) with $A_0 = A, B_0 = B$. The error dynamics depends on the system dynamics.

At the stage of estimator design, we only focus on the nominal error dynamics, taking the term $f(.)$ as an external signal. The interaction between the controller and the estimator will be taken care of after the design by testing the robust stability of the closed-loop system. The speed of the nominal error dynamics $e(k+1) = (A_0 - L_pC)e(k)$ can be influenced by specifying a minimum decay rate ρ $(0 < \rho < 1)$ such that there exists a matrix $K > 0$ and L_p satisfying

$$\rho^2 e(k)^T K e(k) \geq e(k+1)^T K e(k+1),$$

which is equivalent to

$$\rho^2 K - (A_0 - L_pC)K(A_0 - L_pC)^T \geq 0. \tag{3.18}$$

Let $M = K^{-1}$ and $N = ML_p$, then (3.18) can be formulated into the LMI constraint

$$M > 0, \quad \begin{bmatrix} \rho^2 M & MA_0 - NC \\ A_0^T M - C^T N^T & M \end{bmatrix} \geq 0. \tag{3.19}$$

Thus, once we choose the parameter ρ, we can find the estimator gain $L_p = M^{-1}N$ by solving the LMI feasibility problem (**??**).

3.3.2 Robust Stability Criteria for Output Feedback Systems

When we implement the designed controller and estimator on-line, we determine a specific $F(k)$ from the lookup table of the controller based on the current estimated state $\hat{x}(k)$. Here for simplicity, we assume that $F(k)$ is independent of $\hat{x}(k)$ and that it belongs to an uncertain set $\mathbf{\Psi} = \mathbf{Co}\{F_1, F_2\} \cup \cdots \cup \mathbf{Co}\{F_{N-1}, F_N\}$. Therefore, the augmented closed-loop system for the polytopic uncertain system (2.6)-(2.7) is

$$\mathcal{X}(k+1) = \mathcal{A}_{poly}(k)\mathcal{X}(k), \tag{3.20}$$

where $\mathcal{X} = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$, $\mathcal{A}_{poly}(k) = \begin{bmatrix} A(k) & B(k)F(k) \\ L_pC & A_0 + B_0F(k) - L_pC \end{bmatrix}$.

The augmented closed-loop system for the norm-bound system (2.8) is

$$\begin{aligned} \mathcal{X}(k+1) &= \mathcal{A}_{norm}(k)\mathcal{X}(k) + \mathcal{B}_p p(k), \\ q(k) &= \mathcal{C}_q(k)\mathcal{X}(k), \\ p(k) &= (\Delta q)(k), \end{aligned} \tag{3.21}$$

where $\mathcal{A}_{norm}(k) = \begin{bmatrix} A & BF(k) \\ L_pC & A_0 + B_0F(k) - L_pC \end{bmatrix}$, $\mathcal{B}_p = \begin{bmatrix} B_p \\ 0 \end{bmatrix}$ and $\mathcal{C}_q = \begin{bmatrix} C_q & D_{qu}F(k) \end{bmatrix}$.

It is obvious that if the augmented systems (3.20) and (3.21) are stable, so are the original output feedback systems where the dependency of $F(k)$ on $\hat{x}(k)$ is taken into account.

Lemma 3.2 (Robust stability criteria for output feedback systems)

(a) *The augmented system (3.20) is stable, if there exists a matrix $Q > 0$ such that for all vertices of $\mathbf{\Omega}$ and all F_i in the set $\mathbf{\Psi}$,*

$$\begin{bmatrix} Q & Q\mathcal{A}_{poly,i,j}^T \\ \mathcal{A}_{poly,i,j}Q & Q \end{bmatrix} > 0, \tag{3.22}$$

where $\mathcal{A}_{poly,i,j} = \begin{bmatrix} A_j & B_jF_i \\ L_pC & A_0 + B_0F_i - L_pC \end{bmatrix}$, $j = 1, \ldots, L$, $i = 1, \ldots, N$.

(b) *The augmented system (3.21) is stable, if there exists a matrix $Q > 0$ and a matrix $\Lambda = \boldsymbol{diag}(\lambda_1 I_{n_1}, \ldots, \lambda_l I_{n_l}) > 0$ such that for all F_i in the set $\mathbf{\Psi}$,*

$$\begin{bmatrix} Q & Q\mathcal{A}_{norm,i}^T & Q\mathcal{C}_{q,i}^T \\ \mathcal{A}_{norm,i}Q & Q - \mathcal{B}_p\Lambda\mathcal{B}_p^T & 0 \\ \mathcal{C}_{q,i}Q & 0 & \Lambda \end{bmatrix} > 0, \tag{3.23}$$

where $\mathcal{A}_{norm,i} = \begin{bmatrix} A & BF_i \\ L_pC & A_0 + B_0F_i - L_pC \end{bmatrix}$ *and* $\mathcal{C}_{q,i} = \begin{bmatrix} C_q & D_{qu}F_i \end{bmatrix}$, $i = 1, \ldots, N$.

Proof See [21]. ■

By forcing F to satisfy condition (3.22) or (3.23) in Lemma 3.2, we lump together the effect of constraints on the stability of the closed-loop system. The same simplification has been exploited in constrained $\mathcal{H}_\infty$ control [17] and anti-windup schemes [29], where input constraints are formulated into polytopic model uncertainty and sector bound nonlinearity, respectively. But in our MPC algorithm, the conservatism is reduced by online determination of a feedback matrix F from the set $\mathbf{\Psi}$.

Consequently, the off-line design of the output feedback RCMPC is as follows.

Step 1 Specify the controller design parameters $\mathcal{Q}$ and $\mathcal{R}$, obtain a look-up table of (Q_i, F_i), $i = 1, \ldots, N$, by following the steps in the off-line part of Algorithm 3.2 in section 3.2.

Step 2 Specify the estimator design parameters ρ, obtain an estimator gain $L_p = M^{-1}N$ satisfying (3.19).

Step 3 Test one of the robust stability criteria in Lemma 3.2. If not satisfied, go back to Step 1 and 2.

It is obvious that the off-line design of the output feedback RCMPC involves iterative design of the controller (Step 1) and the estimator (Step 2). The design parameters $\mathcal{Q}$, $\mathcal{R}$ and ρ can be adjusted to satisfy condition (3.22) or (3.23). Without loss of generality, consider the augmented system (3.20). If the original system is open loop stable, increasing $\mathcal{R} \to \infty$ or $\rho \to 1$ is guaranteed to find a feasible design which satisfies the robust stability criterion (3.22) in Lemma 3.2, because for the extreme case when $F \to 0$ (or $L_p \to 0$), the augmented system becomes lower (or upper) triangular, and each diagonal block is stable. On the other hand, increasing F or L_p can help recover the robustness properties of the estimator and the state feedback controller, respectively. So decreasing $\mathcal{R}$ or ρ can also achieve the satisfaction of the robust stability criterion with improved performance [20].

The main theorem in this section is stated below.

Theorem 3.4 (Output feedback RCMPC with a SLF)
Consider an uncertain dynamical system (2.6) or (2.8) subject to input and output constraints (2.2) and (2.3). Off-line, iterate Steps 1, 2 and 3 until the designed controller and estimator satisfy one of the robust stability criteria in Lemma 3.2. On-line, given the estimated state $\hat{x}$ *at time* k *computed by the state estimator (3.17) and provided* $\|\hat{x}(k)\|^2_{Q_1^{-1}} \leq 1$*, perform a bisection search over* Q_i^{-1} *in the lookup table of the controller to find the largest index* i *such that* $\|\hat{x}(k)\|^2_{Q_i^{-1}} \leq 1$*. If* $i < N$*, solve* $\hat{x}(k)^T(\alpha_i Q_i^{-1} + (1-\alpha_i)Q_{i+1}^{-1})\hat{x}(k) = 1$ *for* α_i *and apply the control law* $u(k) = (\alpha_i F_i + (1-\alpha_i)F_{i+1})\hat{x}(k)$*. If* $i = N$*, apply* $u(k) = F_N \hat{x}(k)$*. The resulting time varying state feedback matrix* $F(k)$ *robustly asymptotically stabilizes the closed-loop system.*

Proof See [20]. ■

3.4 Summary and Discussion

3.4.1 Summary

In this chapter, we have presented some RCMPC algorithms with guaranteed robust stability of the close-loop system for two classes of uncertainty descriptions. In the on-line strategy, the goal is to design a state feedback law that minimizes an upper bound of the robust performance objective at each time instant. By solving the on-line optimization problem, the robust stability of the closed-loop system is guaranteed despite input and output constraints.

The advantage of the off-line state feedback MPC is that it provides off-line set of stabilizing state feedback laws, corresponding to a set of invariant ellipsoids one inside another in state space. Since no optimization is involved except a simple bisection search, the on-line MPC computation is reduced with little or no less of performance. This makes robust MPC a very attractive control methodology for application to large scale systems and fast processes.

Moreover, the off-line approach can be used in output feedback RCMPC scheme. Off-line, we design iteratively a sequence of state feedback laws and a state estimator until the robust stability criterion for the closed-loop system is satisfied. On-line, a specific control law is determined from the sequence of state feedback laws based on the current estimated state with explicit satisfaction of the input and output constraints. Simulation results in chapter 5 will show that this algorithm can guarantee robust stability of the uncertain output feedback systems with substantial reduction of on-line computation.

3.4.2 Discussion

Input/Output componentwise peak bounds

The LMI conditions for Euclidean norm constraints on the input/output can be extended to the case of componentwise peak bounds in a straightforward manner. For the componentwise peak bounds on the input,

$$|u_r(k+i|k)|_2 \leq u_{r,\max},\ i \geq 0, \quad r = 1, 2, \ldots, n_u,$$

if there exists a symmetric matrix X such that

$$\begin{bmatrix} X & Y \\ Y^T & Q \end{bmatrix} \geq 0, \quad \text{with} \quad X_{rr} \leq u_{r,\max}^2. \tag{3.24}$$

Similarly, componentwise peak bounds on the output

$$\max_{[A(k+i) \;\; B(k+i)] \in \mathbf{\Omega},\; i \geq 0} |y_r(k+i|k)|_2 \leq y_{r,\max}, \quad i \geq 1,$$

is satisfied if for polytopic uncertain model, there exists a symmetric matrix Z such that for each vertex of $\mathbf{\Omega}$,

$$\begin{bmatrix} Z & C(A_jQ + B_jY) \\ (A_jQ + B_jY^T)C^T & Q \end{bmatrix} \geq 0, \quad j = 1, 2, \ldots, L, \tag{3.25}$$

where $Z_{rr} \leq y_{r,\max}^2$, $r = 1, 2, \ldots, n_y$.

For norm-bound uncertain model, the constraint

$$\max_{[A(k+i) \;\; B(k+i)] \in \mathbf{\Omega},\; i \geq 0} |y_r(k+i|k)|_2 \leq y_{r,\max}, \quad i \geq 1,$$

is satisfied if for each row of matrix C,

$$\begin{bmatrix} y_{r,\max}^2 Q & (C_qQ + D_{qu}Y)^T & (AQ + BY)^T C_r^T \\ C_qQ + D_{qu}Y & T_r^{-1} & 0 \\ C_r(AQ + BY) & 0 & I - C_r B_p T_r^{-1} B_p^T C_r^T \end{bmatrix} \geq 0, \tag{3.26}$$

where $T_r = \mathbf{diag}[t_{r,1}I_{n_1}, t_{r,2}I_{n_2}, \ldots, t_{r,l}I_{n_r}] > 0$, $r = 1, 2, \ldots, n_y$ and C_r denotes the rth row of C. The proofs of these conditions are given in [9].

Model Extension

For simplicity, it is assumed in this thesis that only $\begin{bmatrix} A(k) & B(k) \end{bmatrix}$ are uncertain. However, the uncertainty in C can also be incorporated into the formulation. For polytopic uncertainty, $\mathbf{\Omega}$ is the polytope $\mathbf{Co}\{\begin{bmatrix} A_1 & B_1 & C_1 \end{bmatrix}, \ldots, \begin{bmatrix} A_L & B_L & C_L \end{bmatrix}\}$ where $\begin{bmatrix} A_j & B_j & C_j \end{bmatrix}$ are vertices of the convex hull. The output constraint (3.10) is changed to

$$\begin{bmatrix} y_{\max}^2 I & C_i(A_jQ + B_jY) \\ (A_jQ + B_jY^T)C_i^T & Q \end{bmatrix} \geq 0, \quad i, j = 1, 2, \ldots, L.$$

Numbers of off-line states

It all depends. As one can see, the way the ellipsoids are generated in the chapter determines the solutions which are all suboptimal, more ellipsoids can't change this suboptimality. Moreover, the sacrifice of optimality is for the purpose of computational efficiency. And less number of ellipsoids means less time for searching the off-line solution.

Complexity Analysis

In Algorithms 3.2 and 3.3, the on-line computation mainly comes from the bisection search in a lookup table. A sequence of K stored Q_i^{-1} (K generally less than 20) requires $\log_2 K$ searches, and the matrix-vector multiplication in one search has quadratic growth $\mathcal{O}(n_s^2)$ in the number of flops, with n_s the number of state variables. Therefore, the total number of flops required to calculate an input move is $\mathcal{O}(n_s^2 \log_2 K)$. On the other hand, the fastest interior point algorithms show $\mathcal{O}(MN^3)$ growth in computation [26] where M is the total number of scalar decision variables. M is proportional to L and $N \sim n_s^2/2 + n_s n_c$, with L the number of vertices of the uncertain model and n_c the number of manipulated variables. Therefore we can conclude that this off-line approach substantially reduce the on-line computational burden in RCMPC.

Chapter 4

RCMPC WITH A PDLF

This chapter introduces an improved algorithm of RCMPC discussed in the previous chapter, but only for polytopic uncertain systems. The employment of a single Lyapunov function is replaced by that of a parameter-dependent Lyapunov function, and hence, yielding less conservative results.

Section 4.1 gives the detailed derivation of the state feedback RCMPC with a PDLF. The proposed formulation is based on the new robust stabilizability condition introduced in section 2.5. Section 4.2 is then devoted to analyze the robustness of the control law. In the sequence, off-line state feedback and output feedback RCMPC with a PDLF are presented by the next two sections. This chapter ends with some concluding remarks.

4.1 State Feedback RCMPC with a PDLF

4.1.1 Derivation of the Control Law

The main result of this section is given by the following theorem.

Theorem 4.1 (LMI conditions)

Consider the system (2.6). Given the state $x(k) = x(k|k)$ measured at sampling time k, assume that there exist L symmetric matrices Q_j with $j = 1, 2, \ldots, L$, a pair of matrices $\{Y, G\}$ and a positive scalar γ such that

$$\begin{bmatrix} 1 & x(k|k)^T \\ x(k|k) & Q_j \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L, \tag{4.1}$$

$$\begin{bmatrix} G + G^T - Q_j & * & * & * \\ A_j G + B_j Y & Q_l & * & * \\ \boldsymbol{\mathcal{Q}}^{1/2} G & 0 & \gamma I & * \\ \boldsymbol{\mathcal{R}}^{1/2} Y & 0 & 0 & \gamma I \end{bmatrix} > 0, \quad \forall j = 1, 2, \ldots, L, \quad l = 1, 2, \ldots, L, \tag{4.2}$$

$$\begin{bmatrix} u_{\max}^2 I & Y \\ Y^T & G + G^T - Q_j \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L, \tag{4.3}$$

$$\begin{bmatrix} y_{\max}^2 I & * \\ (A_j G + B_j Y)^T C^T & G + G^T - Q_j \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L, \tag{4.4}$$

where the symbol $*$ stands for the transpose of the symmetric blocks in matrix inequalities. Then, applying the state feedback control law

$$u(k+i|k) = F(k)x(k+i|k), \quad \forall i \geq 0, \tag{4.5}$$

to the process (2.6) where $F(k) = YG^{-1}$, the following inequalities hold.

$$J_{WC}(k) < V(k|k) \leq \gamma, \tag{4.6}$$

$$\|u(k+i|k)\|_2 \leq u_{\max}, \quad i \geq 0, \tag{4.7}$$

$$\|y(k+i|k)\|_2 \leq y_{\max}, \quad i \geq 1, \tag{4.8}$$

where

$$\begin{aligned} V(k+i|k) &:= x(k+i|k)^T P(k+i) x(k+i|k), \\ P(k+i) &:= \sum_{j=1}^{L} \lambda_j(k+i) P_j, \;\; P_j := \gamma Q_j^{-1}. \end{aligned}$$

Proof We first show that the first inequality of (4.6) holds. The condition (4.2) implies that $G + G^T - Q_j > 0$ and $Q_j > 0$, hence the matrix G is nonsingular. Furthermore, since $Q_j > 0$, we have $(Q_j - G)^T Q_j^{-1}(Q_j - G) \geq 0$. Consequently,

$$G^T Q_j^{-1} G \geq G^T + G - Q_j. \tag{4.9}$$

By taking into account of (4.9), we can state that (4.2) implies

$$\begin{bmatrix} G^T Q_j^{-1} G & * & * & * \\ A_j G + B_j Y & Q_l & * & * \\ \mathcal{Q}^{1/2} G & 0 & \gamma I & * \\ \mathcal{R}^{1/2} Y & 0 & 0 & \gamma I \end{bmatrix} > 0, \quad \forall j = 1, 2, \ldots, L, \quad l = 1, 2, \ldots, L. \tag{4.10}$$

Substituting $Y = F(k)G$ into (4.10) and multiplying it from the left by **diag**$[G^{-T}, Q_l^{-1}, I, I]$ and from the right by **diag**$[G^{-1}, Q_l^{-1}, I, I]$, we have

$$\begin{bmatrix} Q_j^{-1} & * & * & * \\ Q_l^{-1}(A_j + B_j F(k)) & Q_l^{-1} & * & * \\ \mathcal{Q}^{1/2} & 0 & \gamma I & * \\ \mathcal{R}^{1/2} F(k) & 0 & 0 & \gamma I \end{bmatrix} > 0, \quad \forall j = 1, 2, \ldots, L, \quad l = 1, 2, \ldots, L. \tag{4.11}$$

Then, substituting $Q_j = \gamma P_j^{-1}$, $Q_l = \gamma P_l^{-1}$ into (4.11) and multiplying the resulting inequalities by $\lambda_l(k+i+1)$ and summing them up for $l = 1, 2, \ldots, L$, we obtain

$$\begin{bmatrix} \frac{1}{\gamma} P_j & * & * & * \\ \frac{1}{\gamma} P(k+i+1)(A_j + B_j F(k)) & \frac{1}{\gamma} P(k+i+1) & * & * \\ \mathcal{Q}^{1/2} & 0 & \gamma I & * \\ \mathcal{R}^{1/2} F(k) & 0 & 0 & \gamma I \end{bmatrix} > 0, \quad \forall j = 1, 2, \ldots, L. \tag{4.12}$$

Next, multiplying (4.12) by $\lambda_j(k+i)$ and summing them up for $j = 1, 2, \ldots, L$, we get

$$\begin{bmatrix} \frac{1}{\gamma} P(k+i) & * & * & * \\ \frac{1}{\gamma} P(k+i+1)\{A(k+i) + B(k+i)F(k)\} & \frac{1}{\gamma} P(k+i+1) & * & * \\ \mathcal{Q}^{1/2} & 0 & \gamma I & * \\ \mathcal{R}^{1/2} F(k) & 0 & 0 & \gamma I \end{bmatrix} > 0. \tag{4.13}$$

Applying the Schur complement to (4.13), we have

$$\begin{bmatrix} P(k+i) - \{\mathcal{Q} + F(k)^T \mathcal{R} F(k)\} & * \\ P(k+i+1)\{A(k+i) + B(k+i)F(k)\} & P(k+i+1) \end{bmatrix} > 0. \tag{4.14}$$

Then, applying the Schur complement to (4.14) and multiplying the resulting inequality from the left by $x(k+i|k)^T$ and from the right by $x(k+i|k)$ with taking into account of (2.6) and (4.5), we have

$$V(k+i+1|k) - V(k+i|k) < -\{x(k+i|k)^T \mathcal{Q} x(k+i|k) + u(k+i|k)^T \mathcal{R} u(k+i|k)\}, \\ \forall \begin{bmatrix} A(k+i) & B(k+i) \end{bmatrix} \in \mathbf{\Omega}, \ \ i \geq 0. \tag{4.15}$$

Summing (4.15) from $i = 0, 1, \ldots, \infty$ and requiring $x(\infty|k) = 0$ or $V(\infty|k) = 0$, it follows that

$$J_{WC}(k) < V(k|k).$$

Hence, the first inequality of (4.6) holds.

Next, we show that the second inequality of (4.6) holds. Applying the congruence transformation to (4.1) with $\mathbf{diag}[1, Q_j^{-1}]$ and multiplying the resulting inequalities by $\lambda_j(k)$ and summing them up for $j = 1, 2, \ldots, L$, we obtain

$$\sum_{j=1}^{L} \lambda_j(k) \begin{bmatrix} 1 & x(k|k)^T Q_j^{-1} \\ Q_j^{-1} x(k|k) & Q_j^{-1} \end{bmatrix} \geq 0. \tag{4.16}$$

Substituting $Q_j = \gamma P_j^{-1}$ into (4.16) and applying the congruence transformation to the resulting inequality with $\mathbf{diag}[1, \gamma P(k)^{-1}]$, we have

$$\begin{bmatrix} 1 & x(k|k)^T \\ x(k|k) & \gamma P(k)^{-1} \end{bmatrix} \geq 0,$$

which, by the Schur complement, yields

$$x(k|k)^T P(k) x(k|k) \leq \gamma. \tag{4.17}$$

Hence, we conclude that the second inequality of (4.6) holds.

Next, we show the that the constraint (4.7) holds. From (4.9), it is straightforward to see that (4.3) implies

$$\begin{bmatrix} u_{\max}^2 I & Y \\ Y^T & G^T Q_j^{-1} G \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L. \tag{4.18}$$

Substituting $Y = F(k)G$ into (4.18) and multiplying the resulting inequality from the left by $\mathbf{diag}[I, G^{-T}]$ and from the right by $\mathbf{diag}[I, G^{-1}]$, we have

$$\begin{bmatrix} u_{\max}^2 I & F(k) \\ F(k)^T & Q_j^{-1} \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L. \tag{4.19}$$

Substituting $Q_j = \gamma P_j^{-1}$ into (4.19) and multiplying the resulting inequalities by $\lambda_j(k+i)$ and summing them up for $j = 1, 2, \ldots, L$, we obtain

$$\begin{bmatrix} u_{\max}^2 I & F(k) \\ F(k)^T & \frac{1}{\gamma} P(k+i) \end{bmatrix} \geq 0. \tag{4.20}$$

Applying the Schur complement to (4.20) and multiplying the resulting inequality from the left by $x(k+i|k)^T$ and from the right by $x(k+i|k)$ and taking into account of (4.5), we get

$$\frac{1}{u_{max}^2}u(k+i|k)^T u(k+i|k) \leq \frac{1}{\gamma}x(k+i|k)^T P(k+i)x(k+i|k). \tag{4.21}$$

Since the inequality (4.15) implies that $V(k+i|k)$ strictly decreases as i goes to ∞ and $V(k|k) \leq \gamma$ from (4.17), we have

$$\frac{1}{\gamma}x(k+i|k)^T P(k+i)x(k+i|k) \leq 1, \quad \forall i \geq 0. \tag{4.22}$$

Hence, from (4.21) and (4.22), we conclude that (4.7) holds.

Finally, we show that (4.8) holds. From (4.9) and using $Y = F(k)G$, it is obvious to see that (4.4) implies

$$\begin{bmatrix} y_{\max}^2 I & * \\ G^T(A_j + B_j F(k))^T C^T & G^T Q_j^{-1} G \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L. \tag{4.23}$$

Multiplying (4.23) from the left by $\mathbf{diag}[I, G^{-T}]$ and from the right by $\mathbf{diag}[I, G^{-1}]$ and substituting $Q_j = \gamma P_j^{-1}$ into the resulting inequality, we have

$$\begin{bmatrix} y_{\max}^2 I & * \\ (A_j + B_j F(k))^T C^T & \frac{1}{\gamma}P_j \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L. \tag{4.24}$$

Multiplying (4.24) by $\lambda_j(k+i)$ and summing them up for $j = 1, 2, \ldots, L$, we have

$$\begin{bmatrix} y_{\max}^2 I & * \\ \{A(k+i) + B(k+i)F(k)\}^T C^T & \frac{1}{\gamma}P(k+i) \end{bmatrix} \geq 0. \tag{4.25}$$

Applying the Schur complement to (4.25) and multiplying the resulting inequality from the left by $x(k+i|k)^T$ and from the right by $x(k+i|k)$ and taking into account of

$$y(k+i+1|k) = C\{A(k+i) + B(k+i)F(k)\}x(k+i|k),$$

we obtain

$$\frac{1}{y_{max}^2}y(k+i+1|k)^T y(k+i+1|k) \leq \frac{1}{\gamma}x(k+i|k)^T P(k+i)x(k+i|k). \tag{4.26}$$

From (4.22) and (4.26), we have

$$\frac{1}{y_{max}^2}y(k+i+1|k)^T y(k+i+1|k) \leq 1, \quad \forall i \geq 0,$$

which implies that (4.8) holds. ■

Note that in the MPC scheme, the performance index γ is minimized at each sampling time k (equivalently, an upper bound of $J_{WC}(k)$ is minimized). A physical interpretation of this performance index is a measure on how close the state is to the origin without an excessive expenditure of the control effort. Moreover, the function $V(k|k)$ in this case is a parameter-dependent Lyapunov function used to guarantee robust stability of the uncertain system. It is clearly seen that the proposed control technique is less conservative than the previous control law using a single Lyapunov function [9]. As a result, we expect that the performance index γ obtained by the proposed technique will always be no greater than the one obtained by the method in chapter 3.

In fact, the formulation in Theorem 4.1 is an extension of the robust stabilization for uncertain time-varying systems which was addressed by Cuzzola et al. [22] and Mao [23]. Mao also pointed out that the robust stability conditions in [22] was actually held for uncertain time-invariant systems. Although, in [23], the conditions for robust stabilization for uncertain time-varying systems are given, the formulation does not consider constraints on control input and output. We extend the main results in [22, 23] to design robust MPC of uncertain LTV systems subject to constraints on control input and output. The condition (4.2) can be simplified for the time-invariant case by imposing $Q = Q_j$ in (4.2) and thus, the number of LMIs decreases from $L \times L$ to L.

Based on Theorem 4.1, we are now ready to state the algorithm for the implementation of state feedback RCMPC. It is given as follows.

Algorithm 4.1 (State feedback RCMPC with a PDLF)

1. *Get the measured state x(k).*
2. *Solve* $\min_{Y,G,Q_j} \gamma$, *s.t. (4.1)-(4.4) and compute* $F(k)$.
3. *Apply* $u(k) = F(k)x(k)$ *to the process.*
4. *Set* $k := k + 1$ *and go to 1.*

4.1.2 Robust Stability

In this section, we will ensure that the control law in Algorithm 4.1 stabilizes uncertain time-varying systems (2.6). In order to prove the robust stability of the closed-loop system, we need to establish the feasibility of Algorithm 4.1.

Lemma 4.1 (Feasibility)
Consider the system (2.6). Assume that the conditions (4.1)-(4.4) in Theorem 4.1 are feasible at time k. *Then Algorithm 4.1 is feasible for all times* $t > k$.

Proof Let us assume that the conditions (4.1)-(4.4) in Theorem 4.1 are feasible at time k. The only LMI in the problem that depends explicitly on the measured state $x(k|k) = x(k)$ of the system is (4.1). Thus, to prove the lemma, we only need to prove that this LMI is feasible for all future measured states $x(k+i|k+i) = x(k+i)$, $i \geq 1$.

Now, the feasibility of the algorithm at time k implies satisfaction of (4.22), which means that for any $\begin{bmatrix} A(k+i) & B(k+i) \end{bmatrix} \in \Omega$, $i \geq 0$, we must have

$$x(k+i|k)^T P^{(k)}(k+i) x(k+i|k) \leq \gamma^{(k)}, \quad \forall i \geq 1, \tag{4.27}$$

where $P^{(k)}(k+i) = \sum_{j=1}^{L} \lambda_j(k+i) P_j^{(k)}$ and $P_j^{(k)}$ denotes the solution obtained at time k. Because of the measured state at time $k+1$

$$x(k+1|k+1) = [A(k) + B(k)F(k)]x(k|k)$$

for some $\begin{bmatrix} A(k) & B(k) \end{bmatrix} \in \Omega$, it must also satisfy inequality (4.27), i.e.,

$$x(k+1|k+1)^T P^{(k)}(k+1)x(k+1|k+1) \leq \gamma^{(k)}. \tag{4.28}$$

In other words, the feasible solution at time k is also feasible at time $k+1$. Thus, the optimization problem is feasible at time $k+1$. This argument can be repeated for subsequent sampling times, i.e., the optimization problem is feasible at time $k+2, k+3, \ldots, \infty$. Hence, we conclude that Algorithm 4.1 is feasible for all times $t > k$. ■

The robust stability of the closed-loop system is stated in the next theorem.

Theorem 4.2 (Robust stability)
Consider the system (2.6). Assume that the conditions (4.1)-(4.4) in Theorem 4.1 are feasible at time 0*. Then the control law in Algorithm 4.1 robustly asymptotically stabilizes the closed-loop system.*

Proof To prove asymptotic stability, we will show that $V(k|k) = x(k|k)^T P^{(k)}(k)x(k|k)$ is a strictly decreasing Lyapunov function. Note that $P^{(k)}(k) = \sum_{j=1}^{L} \lambda_j(k) P_j^{(k)}$ and $P_j^{(k)} > 0$ is the optimal solution obtained at time k.

Let us assume that the conditions in Theorem 4.1 are feasible at time 0. Lemma 4.1 then ensures the feasibility of Algorithm 4.1 at all times $k > 0$. Since γ is minimized at each time k, the following inequality holds.

$$x(k+1|k+1)^T P^{(k+1)}(k+1)x(k+1|k+1) \leq x(k+1|k+1)^T P^{(k)}(k+1)x(k+1|k+1). \tag{4.29}$$

From (4.15) with $i = 0$, the following inequality holds for any $\begin{bmatrix} A(k) & B(k) \end{bmatrix} \in \Omega$.

$$x(k+1|k)^T P^{(k)}(k+1)x(k+1|k) < x(k|k)^T P^{(k)}(k)x(k|k), \quad (x(k|k) \neq 0). \tag{4.30}$$

Since the measured state $x(k+1|k+1) = x(k+1)$, i.e.,

$$x(k+1) = [A(k) + B(k)F(k)]x(k|k)$$

for some $\begin{bmatrix} A(k) & B(k) \end{bmatrix} \in \Omega$, it must also satisfy inequality (4.30), that is,

$$x(k+1|k+1)^T P^{(k)}(k+1)x(k+1|k+1) < x(k|k)^T P^{(k)}(k)x(k|k), \quad (x(k|k) \neq 0). \tag{4.31}$$

Hence, from (4.29) and (4.31), we have

$$x(k+1|k+1)^T P^{(k+1)}(k+1)x(k+1|k+1) < x(k|k)^T P^{(k)}(k)x(k|k), \quad (x(k|k) \neq 0). \tag{4.32}$$

Therefore, $V(k|k)$ is a strictly decreasing Lyapunov function, which implies that $x(k) \to 0$ as $k \to \infty$. ■

4.2 Off-line State Feedback RCMPC with a PDLF

4.2.1 Asymptotically Stable Invariant Ellipsoid

A part of the theory behind the off-line strategy for state feedback MPC is the construction of asymptotically stable invariant ellipsoids. In other words, it is necessary to find a region of attraction where the state, once inside the region, will never leave it and finally steer to the origin.

First, we review the LMI optimization problem used to find the state feedback law for RCMPC as mentioned in Algorithm 4.1 of section 4.1.

$$\begin{aligned} &\min_{\gamma,Y,G,Q_j} \quad \gamma \\ &\text{subject to} \quad (4.1), (4.2), (4.3) \text{ and } (4.4). \end{aligned} \tag{4.33}$$

The following lemma is to find an asymptotically stable invariant ellipsoid for the uncertain system.

Lemma 4.2 *Consider the polytopic uncertain system (2.6). Assume that LMI optimization problem (4.33) applied to a system state x_0 has a solution represented by a scalar γ, matrices Q_j with $j = 1, 2, \ldots, L$ and a pair of matrices $\{Y, G\}$. If the state feedback control law $u(k) = YG^{-1}x(k)$ is adopted, then an asymptotically stable invariant ellipsoid is $\mathcal{E} = \{x \in \mathbf{R}^n | x^T \mathbb{Q}^{-1} x \leq 1\}$ where the matrix $\mathbb{Q}$ can be obtained as the solution of the following LMI optimization problem.*

$$\begin{aligned} &\max_{\beta,\mathbb{Q}} \quad \beta \\ &\textit{subject to} \quad \beta I < \mathbb{Q} \leq Q_j, \quad \forall j = 1, 2, \ldots, L. \end{aligned} \tag{4.34}$$

Proof First, from Theorem 4.1, we observe that $Q(k+i) = \gamma P(k+i)^{-1},\ k, i > 0$ is a convex combination of matrices $Q_j,\ j = 1, \ldots, L$. Consequently, an invariant ellipsoid can be found by determining the maximal matrix $\mathbb{Q}$ that can be expressed as a convex combination of the matrices $Q_j,\ j = 1, \ldots, L$. Clearly, this matrix has to satisfy the constraints $\mathbb{Q} \leq Q_j,\ j = 1, \ldots, L$ and can be found by the LMI optimization problem (4.34).

Also in Theorem 4.1, the inequalities (4.1) that depend on the system state are automatically satisfied for all states within the ellipsoid $\mathcal{E}$. Therefore, the minimizer γ, Q_j, Y, G given at the state x_0 is also feasible (not necessarily optimal) for any other state in $\mathcal{E}$. Thus, we can apply the state feedback law $u = YG^{-1}x$ to any non-zero $\tilde{x}(k) \in \mathcal{E}$, where $\tilde{x}(k) \neq x_0$ and still satisfy (4.2)-(4.4), thereby ensuring that in real time

$$\tilde{x}(k+i+1)^T \mathbb{Q} \tilde{x}(k+i+1) < \tilde{x}(k+i)^T \mathbb{Q} \tilde{x}(k+i) \leq 1, \quad i \geq 0.$$

Therefore, $\tilde{x}(k+i) \in \mathcal{E},\ i \geq 0$ and $\tilde{x}(k+i)$ converges to the origin as i goes to infinity. This establishes that $\mathcal{E}$ is an asymptotically stable invariant ellipsoid. ■

4.2.2 Off-line Formulations

Once asymptotically stable invariant ellipsoids are established, we can obtain the off-line version of state feedback RCMPC with a PDLF following the idea of section 3.2.

Theorem 4.3 (Off-line state feedback RCMPC with a PDLF)
Consider a dynamical system (2.6) with input and output constraints (2.2) and (2.3). Off-line, given an initial feasible state x_1*, generate a sequence of* $\mathbb{Q}_i$ *and* F_i *(i=1,...,N) as follows.*

1. *Compute the minimizer* γ_i, Q_{ij} $(j = 1, \ldots, L)$, Y_i *and* G_i *at* x_i *by solving the LMI (4.33).*
2. *Compute the maximizer* $\mathbb{Q}_i$ *for each* x_i *by solving the LMI (4.34) with an additional constraint* $\mathbb{Q}_{i-1} > \mathbb{Q}_i$ *(ignored at* $i = 1$*); store* $\mathbb{Q}_i^{-1}$ *and* F_i $(= Y_i G_i^{-1})$ *in a look-up table.*
3. *If* $i < N$*, choose a state* x_{i+1} *satisfying* $\|x_{i+1}\|^2_{\mathbb{Q}_i^{-1}} < 1$*. Let* $i := i + 1$*, go to 1.*

On-line, given an initial state $x(0)$ *satisfying* $\|x(0)\|^2_{\mathbb{Q}_1^{-1}} \leq 1$*, let the state be* $x(k)$ *at time k. Perform a bisection search over* $\mathbb{Q}_i^{-1}$ *in the look-up table to find the index* i *equivalent to the smallest ellipsoid* $\mathcal{E} = \{x \in \mathbf{R}^n | x^T \mathbb{Q}_i^{-1} x \leq 1\}$ *such that* $\|x(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1$*. Apply the control law* $u(k) = F_i x(k)$*. The resulting time varying state feedback matrix* $F(k)$ *robustly asymptotically stabilizes the closed-loop system.*

Proof For the off-line minimization at x_i, $i = 2, \ldots, N$, the additional constraint $\mathbb{Q}_{i-1} > \mathbb{Q}_i$ is equivalent to $\mathbb{Q}_{i-1}^{-1} > \mathbb{Q}_i^{-1}$. This implies that the constructed asymptotically stable invariant ellipsoid

$$\mathcal{E}_i = \{x \in \mathbf{R}^n | x^T \mathbb{Q}_i^{-1} x \leq 1\}$$

is inside $\mathcal{E}_{i-1}$. In other words, $\mathcal{E}_i \subset \mathcal{E}_{i-1}$. So for a fixed x, the weighted norm $\|x\|^2_{\mathbb{Q}_i^{-1}}$ is monotonic with respect to the index i. This ensures the uniqueness of the on-line bisection search in the look-up table for the largest i satisfying $\|x\|^2_{\mathbb{Q}_i^{-1}} \leq 1$.

Given a dynamical system (2.6) and an initial state $x(0)$ satisfying $\|x(0)\|^2_{\mathbb{Q}_1^{-1}} \leq 1$, the closed-loop system becomes

$$x(k+1) = \begin{cases} (A(k) + B(k)F_i)\, x(k) & \text{if} \quad \|x(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1, \quad \|x(k)\|^2_{\mathbb{Q}_{i+1}^{-1}} \geq 1, \\ & \qquad i \neq N \\ (A(k) + B(k)F_N)\, x(k) & \text{if} \quad \|x(k)\|^2_{\mathbb{Q}_N^{-1}} \leq 1. \end{cases}$$

When $x(k)$ satisfies $\|x(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1$ and $\|x(k)\|^2_{\mathbb{Q}_{i+1}^{-1}} \geq 1$, $i = 1, \ldots, N-1$, the control law corresponding to the ellipsoid $\mathcal{E}_i$ is guaranteed to keep the state within $\mathcal{E}_i$ (using Lemma 4.2) and converge it into the ellipsoid $\mathcal{E}_{i+1}$, and so on. Finally, the smallest ellipsoid $\mathcal{E}_N$ is guaranteed to keep the state within $\mathcal{E}_N$ and converge it to the origin. ■

The next theorem is given to construct a continuous feedback matrix over the state space.

Theorem 4.4 (Continuous off-line state feedback RCMPC with a PDLF)
Consider the look-up table generated by the off-line part of Theorem 4.3. If for each x_i $(i = 1, \ldots, N-1)$*, the following condition is satisfied.*

$$\mathbb{Q}_i - (A_j + B_j F_{i+1})^T \mathbb{Q}_i^{-1} (A_j + B_j F_{i+1}) > 0, \quad j = 1, \ldots, L. \tag{4.35}$$

Then, on-line, given an initial state $x(0)$ *satisfying* $\|x(0)\|^2_{\mathbb{Q}_1^{-1}} \leq 1$*, let the state be* $x(k)$ *at time k. Perform a bisection search over* $\mathbb{Q}_i^{-1}$ *in the look-up table to find the index* i *equivalent to the* *smallest ellipsoid* $\mathcal{E} = \{x \in \mathbf{R}^n | x^T \mathbb{Q}_i^{-1} x \leq 1\}$ *such that* $\|x(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1$. *If* $i \neq N$, *solve* $x(k)^T \left(\alpha_i \mathbb{Q}_i^{-1} + (1-\alpha_i)\mathbb{Q}_{i+1}^{-1}\right) x(k) = 1$ *for* α_i *and apply the control law* $u(k) = (\alpha_i F_i + (1-\alpha_i)F_{i+1})x(k)$. *If* $i = N$, *apply* $u(k) = F_N x(k)$. *The resulting time varying state feedback matrix* $F(k)$ *robustly asymptotically stabilizes the closed-loop system.*

Proof The closed-loop system is given by

$$x(k+1) = \begin{cases} \big\{A(k) + B(k)F(\alpha_i(k))\big\}x(k) & \text{if} \quad \|x(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1, \ \|x(k)\|^2_{\mathbb{Q}_{i+1}^{-1}} \geq 1, \\ & \quad\ i \neq N \\ \big\{A(k) + B(k)F_N\big\}x(k) & \text{if} \quad \|x(k)\|^2_{\mathbb{Q}_N^{-1}} \leq 1. \end{cases}$$

Here,

$$F(\alpha_i(k)) = \alpha_i(k)F_i + (1-\alpha_i(k))F_{i+1},$$

with $\alpha_i(k)$ satisfying

$$x(k)^T \left(\alpha_i(k)\mathbb{Q}_i^{-1} + (1-\alpha_i(k))\mathbb{Q}_{i+1}^{-1}\right) x(k) = 1, \quad 0 \leq \alpha_i(k) \leq 1.$$

When $x(k)$ satisfies $\|x(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1$ and $\|x(k)\|^2_{\mathbb{Q}_{i+1}^{-1}} \geq 1$, $i \neq N$, let $F(\alpha_i) = \alpha_i F_i + (1-\alpha_i)F_{i+1}$, $\mathbb{Q}(\alpha_i)^{-1} = \alpha_i \mathbb{Q}_i^{-1} + (1-\alpha_i)\mathbb{Q}_{i+1}^{-1} > 0$, where α_i is solved by satisfying

$$x(k)^T \mathbb{Q}(\alpha_i)^{-1} x(k) = 1, \quad 0 \leq \alpha_i(k) \leq 1.$$

The satisfaction of (4.2) and (4.34) for x_i together with (4.35) ensures that

$$\begin{bmatrix} \mathbb{Q}_i^{-1} & * \\ A_j + B_j F(\alpha_i) & \mathbb{Q}_i \end{bmatrix} > 0, \quad \forall j = 1, \ldots, L.$$

Moreover, the satisfaction of (4.3), (4.4) and (4.34) also ensures that

$$\begin{bmatrix} u_{\max}^2 I & F(\alpha_i) \\ F(\alpha_i)^T & \mathbb{Q}(\alpha_i)^{-1} \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L,$$

$$\begin{bmatrix} y_{\max}^2 I & * \\ (A_j + B_j F(\alpha_i))^T C^T & \mathbb{Q}(\alpha_i)^{-1} \end{bmatrix} \geq 0, \quad \forall j = 1, 2, \ldots, L.$$

Therefore, the control law $u(k) = F(\alpha_i(k))x(k)$ between $\mathcal{E}_i$ and $\mathcal{E}_{i+1}$ is guaranteed to keep the state within $\mathcal{E}_i$ and converge it into the ellipsoid $\mathcal{E}_{i+1}$ with constraints satisfied. Finally, the smallest ellipsoid $\mathcal{E}_N$ is guaranteed to keep the state within $\mathcal{E}_N$ and converge it to the origin. ■

4.3 Output Feedback RCMPC with a PDLF

Robust constrained MPC in combination with discrete-time linear estimator theory is again studied in this section. In this approach, we first design an off-line state feedback RCMPC with a PDLF and an off-line state estimator independently. Then we analyze the robust stabilizability of the combined controller and estimator. If the robust stability criterion is not satisfied, we iterate the design of the controller and estimator by specifying new design parameters. As a comprehensive discussion of this theory was given in section 3.3, only the main topics are repeated here.

4.3.1 Off-line Estimator Design

The state estimator has the form

$$\begin{aligned} \hat{x}(k+1) &= A_0\hat{x}(k) + B_0u(k) + L_p(y(k) - C\hat{x}(k)), \\ \hat{x}(0) &= 0, \end{aligned} \tag{4.36}$$

where $\begin{bmatrix} A_0 & B_0 \end{bmatrix}$ is the nominal model and L_p is the estimator gain. As discussed in section 3.3 of chapter 3, we can obtain the estimator gain by solving the LMI constraint

$$M > 0, \quad \begin{bmatrix} \rho^2 M & MA_0 - NC \\ A_0^T M - C^T N^T & M \end{bmatrix} \geq 0. \tag{4.37}$$

with ρ the pre-specified minimum decay rate of the nominal error dynamics. Here, M and $N = QL_p$ are the variables.

4.3.2 Output Feedback Control Law

Now, we give the condition that guarantee stability of the closed-loop if an off-line state feedback RCMPC is used together with a full state estimator for the recovery of the system state. The augmented closed-loop system is

$$\mathcal{X}(k+1) = \mathcal{A}(k)\mathcal{X}(k) \tag{4.38}$$

where $\mathcal{X} = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$, $\mathcal{A}(k) = \begin{bmatrix} A(k) & B(k)F(k) \\ L_pC & A_0 + B_0F(k) - L_pC \end{bmatrix}$.

Recall that the uncertain set $\mathbf{\Psi} = \mathbf{Co}\{F_1, F_2\} \cup \cdots \cup \mathbf{Co}\{F_{N-1}, F_N\}$, where F_i $(i = 1, \ldots, N)$ are state feedback gains obtained by the off-line part of Theorem 4.3.

Lemma 4.3 *The augmented system (4.38) is stable, if there exists a matrix $Q > 0$ such that for all vertices of $\mathbf{\Omega}$ and all F_i in the set $\mathbf{\Psi}$,*

$$\begin{bmatrix} Q & Q\mathcal{A}_{i,j}^T \\ \mathcal{A}_{i,j}Q & Q \end{bmatrix} > 0 \tag{4.39}$$

where $\mathcal{A}_{i,j} = \begin{bmatrix} A_j & B_jF_i \\ L_pC & A_0 + B_0F_i - L_pC \end{bmatrix}$, $j = 1, \ldots, L,\ i = 1, \ldots, N.$

Proof For system (4.38), if for all vertices of $\mathbf{\Omega}$ and all F_i in the set $\mathbf{\Psi}$, condition (4.39) is satisfied, then for an arbitrary plant $[A(k) \;\; B(k)] \in \mathbf{\Omega}$ and $F(k) \in \mathbf{\Psi}$, we have

$$\begin{bmatrix} Q & Q\mathcal{A}(k)^T \\ \mathcal{A}(k)Q & Q \end{bmatrix} > 0. \tag{4.40}$$

Applying the Schur complement to (4.40) and substituting $P = Q^{-1}$ into the resulting inequality, we obtain

$$P - \mathcal{A}(k)^T P\mathcal{A}(k) > 0,$$

which implies that the quadratic function $\mathcal{X}^T P\mathcal{X}$ is monotonically decreasing. ■

The main result in this section is given by the theorem below.

Theorem 4.5 (Output feedback RCMPC with a PDLF)
Consider a dynamical system (2.6) with input and output constraints (2.2) and (2.3). Off-line, iterate the following steps.

1. *Specify the controller design parameters $\mathcal{Q}$ and $\mathcal{R}$, obtain a look-up table of $\{\mathbb{Q}, F_i\}$, $i = 1, \ldots, N$, by following the steps in the off-line part of Theorem 4.3.*

2. *Specify the estimator design parameters ρ, obtain an estimator gain $L_p = M^{-1}N$ satisfying (4.37).*

3. *Test the robust stability criterion in Lemma 4.3. If not satisfied, go back to Step 1 and 2.*

On-line, given the estimated state $\hat{x}(k)$ at time k computed by the state estimator (4.36) and provided $\|\hat{x}(k)\|^2_{\mathbb{Q}_1^{-1}} \leq 1$, perform a bisection search over $\mathbb{Q}_i^{-1}$ in the lookup table of the controller to find the largest index i such that $\|\hat{x}(k)\|^2_{\mathbb{Q}_i^{-1}} \leq 1$. If $i < N$, solve $\hat{x}(k)^T(\alpha_i \mathbb{Q}_i^{-1} + (1-\alpha_i)\mathbb{Q}_{i+1}^{-1})\hat{x}(k) = 1$ for α_i and apply the control law $u(k) = (\alpha_i F_i + (1-\alpha_i)F_{i+1})\hat{x}(k)$. If $i = N$, apply $u(k) = F_N\hat{x}(k)$. The resulting time varying state feedback matrix $F(k)$ robustly asymptotically stabilizes the closed-loop system.

Proof From Lemma 4.3 and Theorem 4.4. ■

4.4 Summary and Discussion

4.4.1 Summary

For ease of visualization, the above RCMPC algorithms are summarized by the flowcharts in Figures 4.1-4.5. The on-line state feedback scheme is illustrated by Fig. 4.1 according to Algorithm 4.1. Fig. 4.2 and 4.3 depict the algorithm of off-line state feedback RCMPC approach based on Theorem 4.3. Finally, Theorem 4.5 of the output feedback scheme is demonstrated by the flowchart in Fig. 4.4 and 4.5.

4.4.2 Discussion

In this chapter, we have presented an improved approach of RCMPC algorithm for polytopic LTV systems. The performance is defined in terms of an infinite horizon quadratic function whereas the constraints are specified by the Euclidean norm of control input and output. The proposed RCMPC algorithm employs a Lyapunov function which depends on uncertain time-varying parameters of the dynamical system. The design technique can be efficiently implemented since it is reduced to the convex optimization over LMI constraints. The use of a parameter-dependent Lyapunov function leads to a significant improvement of achievable performance. In next chapter, numerical examples based on the two-mass-spring system will confirm that the proposed robust MPC algorithm yields less conservative results than the algorithm with a single Lyapunov function.

After that, we extend the RCMPC using a parameter-dependent Lyapunov function to output feedback scheme. A synthesis approach is to employ off-line observers to estimate states, then feed the estimates through state feedback obtained by the RCMPC technique.

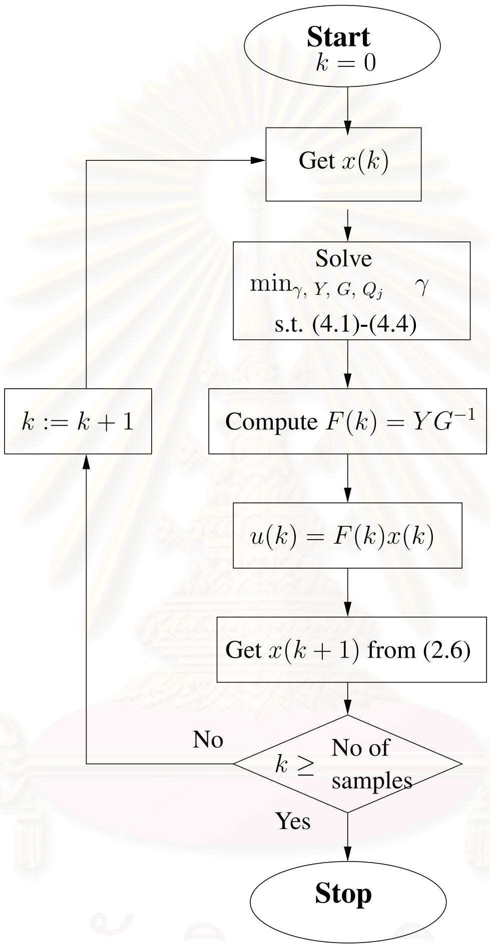


Figure 4.1: Flowchart of on-line state feedback RCMPC with a PDLF.

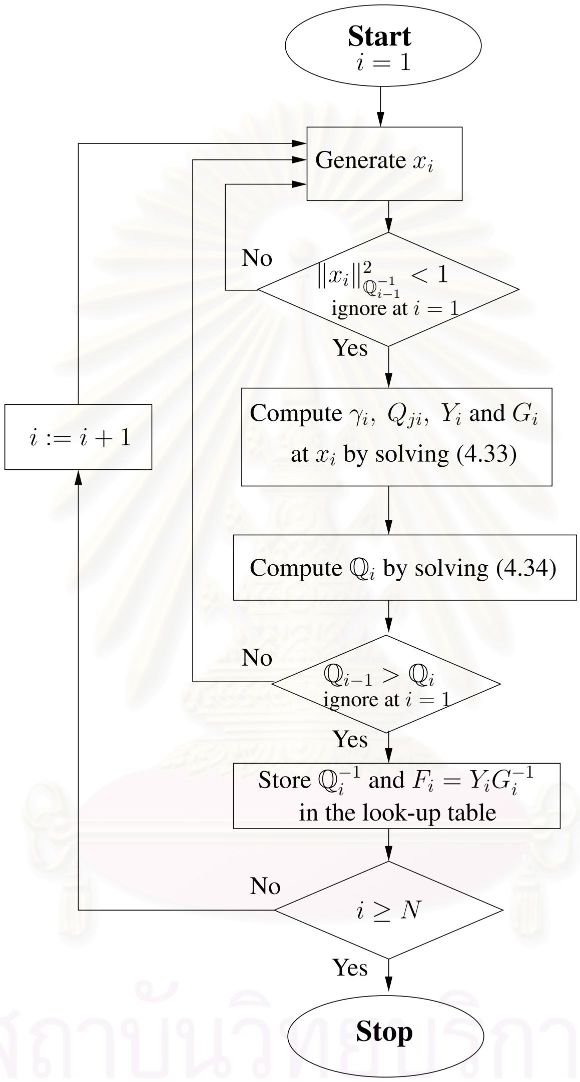


Figure 4.2: Flowchart of off-line part of state feedback RCMPC with a PDLF.

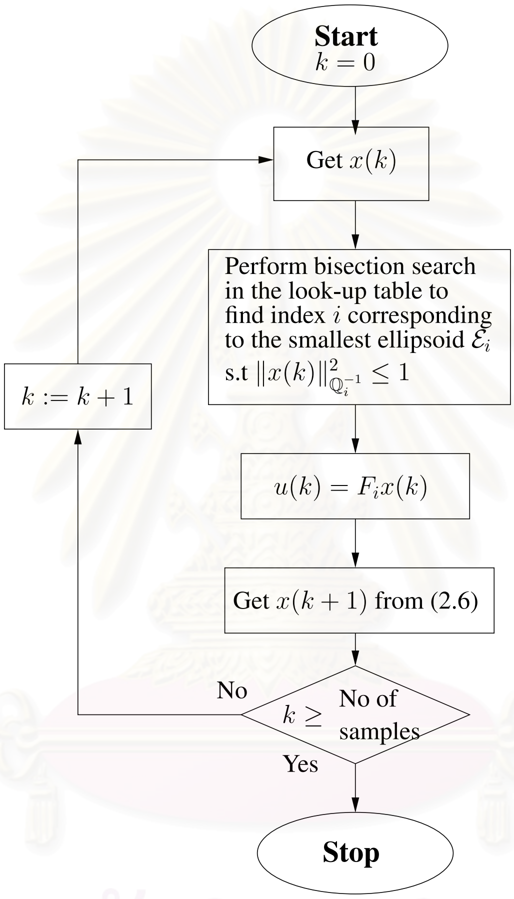


Figure 4.3: Flowchart of on-line part of state feedback RCMPC with a PDLF.

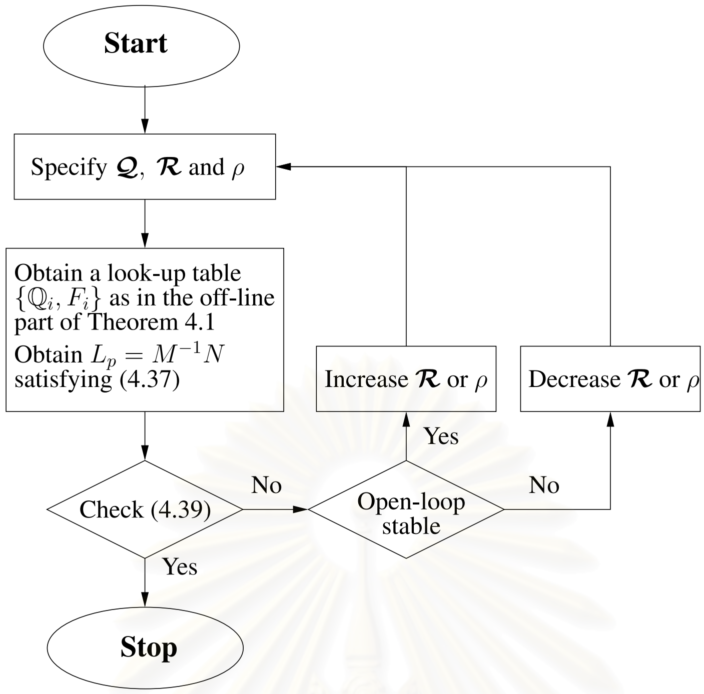


Figure 4.4: Flowchart of off-line part of output feedback RCMPC with a PDLF.

The advantage of using a PDLF over using a single Lyapunov function is that it reduces the conservatism. Therefore, we expect that the performance of the system with a PDLF is better than that with SLF, and the performance index γ is much smaller, as will be shown in an example of chapter 5.

Since we employ an off-line approach for the controller design which gives a sequence of explicit control laws, we can significantly reduce the computational burden. Especially, when the output feedback RCMPC with a PDLF is used, the number of LMIs constraints grows dramatically with the number of uncertain parameters (e.g., n uncertain parameters leads to $4^n + 3 * 2^n$ LMIs constraints in one optimization problem). In addition, testing the robust stability condition for the closed-loop system is a time consuming process, but as it is performed off-line, it should not present any difficulties.

Another advantage of the off-line approach is the ability to analyze the robust stabilizability of the combined control laws and estimator, and by adjusting the design parameters, guarantee robust stability of the closed-loop system in the presence of constraints.

However, a drawback of this method is that the control input at each time k is not optimal. The off-line formulation sacrifices optimality somewhat while significantly reducing the on-line computation and guaranteeing the robust stability of the closed-loop system.

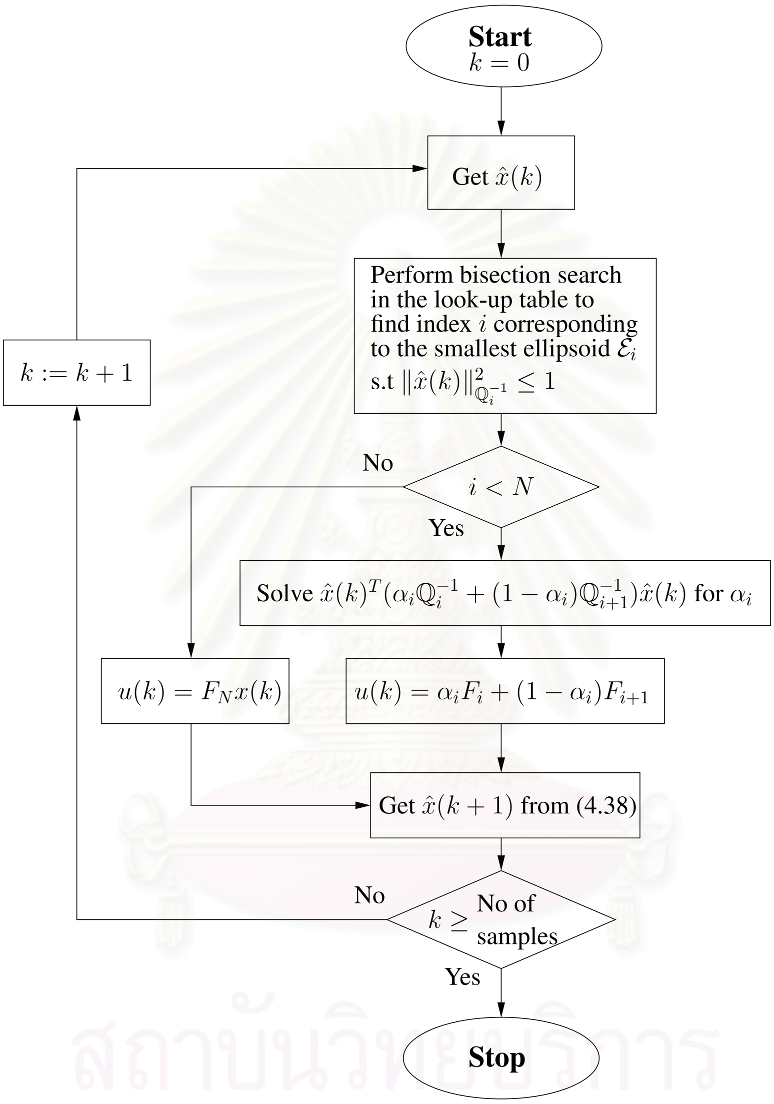


Figure 4.5: Flowchart of on-line part of output feedback RCMPC with a PDLF.

Chapter 5

NUMERICAL EXAMPLES

The results from the previous chapters shall be demonstrated by several examples. The first example based on an angular positioning system will be given to illustrate the implementation of the on-line state feedback RCMPC algorithm with a SLF. In order to show how the off-line state feedback RCMPC is utilized together with a full state estimator to ensure the stability of the output feedback system, another example of a distillation column will be taken into consideration. The last two examples, based on a two-mass-spring system and a non-isothermal CSTR will be presented to show the effectiveness of the proposed RCMPC with a PDLF. For these examples, the software LMI control toolbox [26] in MATLAB environment is used to compute the solution of the LMI problem.

5.1 Angular Positioning System

This example is taken from [9]. As shown in Fig. 5.1, the system consists of a rotating antenna at the origin of the plane, driven by an electric motor. The control problem is to use the input voltage to the motor (u volts) to rotate the antenna so that it always points in the direction of a moving object in the plane.

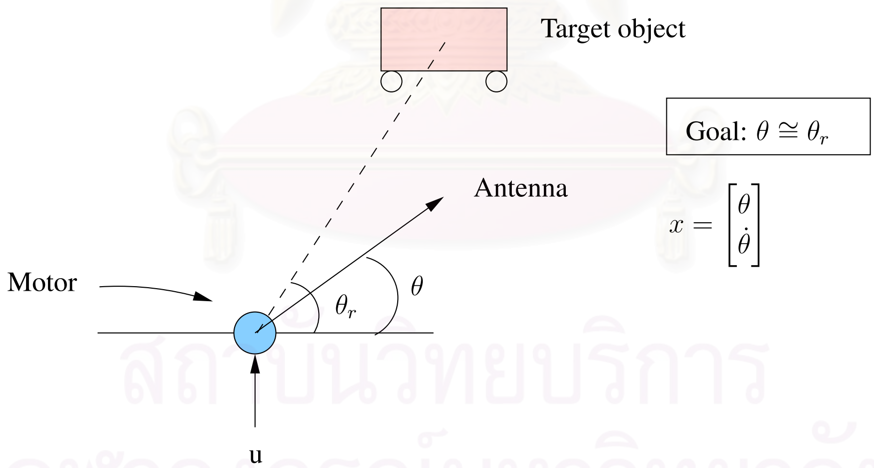


Figure 5.1: Angular positioning system.

Assume that the angular positions of the antenna and the moving object (θ and θ_r rad, respectively) and the angular velocity of the antenna ($\dot{\theta}$ rad.s^{-1}) are measurable. The motion of the antenna can be described by the following discrete-time equations obtained from their continuous-time counterparts by discretization, using a sampling time of 0.1 s and Euler's first-order approximation for the derivative.

$$\begin{aligned}\begin{bmatrix}\theta(k+1)\\ \dot\theta(k+1)\end{bmatrix} &= \begin{bmatrix}1 & 0.1\\ 0 & 1-0.1\alpha(k)\end{bmatrix}\begin{bmatrix}\theta(k)\\ \dot\theta(k)\end{bmatrix} + \begin{bmatrix}0\\ 0.0787\end{bmatrix}u(k),\\ y(k) &= \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta(k)\\ \dot\theta(k)\end{bmatrix},\end{aligned} \tag{5.1}$$

where $0.1\ s^{-1} \leq \alpha(k) \leq 10\ s^{-1}$. The parameter $\alpha(k)$ is proportional to the coefficient of viscous friction in the rotating parts of the antenna and assumed to be arbitrarily time-varying in the indicated range of variation.

We can write (5.1) in the form (2.6), i.e.,

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

Thus, if we use a polytopic model as in (2.7), we see that $A(k) \in \mathbf{\Omega} = \mathbf{Co}\{A_1, A_2\}$, where

$$A_1 = \begin{bmatrix}1 & 0.1\\ 0 & 0.99\end{bmatrix}, \quad A_2 = \begin{bmatrix}1 & 0.1\\ 0 & 0\end{bmatrix}.$$

Alternatively, if we define

$$\begin{aligned}\delta(k) &= \frac{\alpha(k) - 5.05}{4.95},\\ A &= \begin{bmatrix}1 & 0.1\\ 0 & 0.495\end{bmatrix}, \qquad B_p = \begin{bmatrix}0\\ -0.1\end{bmatrix},\\ C_q &= \begin{bmatrix}0 & 4.95\end{bmatrix}, \qquad D_{qu} = 0,\end{aligned}$$

then $\delta(k)$ is time-varying and norm-bounded with $|\delta(k)| \leq 1,\ k \geq 0$. The uncertainty can then be described as in (2.8) with

$$\mathbf{\Omega} = \{A + B_p\delta C_q : |\delta| \leq 1\}.$$

The robust performance index to be minimized at each time k is

$$J_{WC}(k) = \max_{A(k+i)\in\mathbf{\Omega},\ i\geq 0} \sum_{i=0}^{\infty}\{y(k+i|k)^2 + \boldsymbol{\mathcal{R}}u(k+i|k)^2\},$$

with $\boldsymbol{\mathcal{R}} = 2 \times 10^{-5}$. The system is subject to input constraint $|u(k+i|k)| \leq 2,\ i \geq 0$.

Fig. 5.2 and Fig. 5.3 show the closed-loop response of the system when $\alpha(k)$ is randomly time-varying between 0.1 and 10 s^{-1}. The control is synthesized according to the on-line state feedback RCMPC algorithm in Theorem 3.2. Also included in these figures are the response and control signal using a *static* state feedback control law, where the feedback matrix F computed from Theorem 3.2 at time $k = 0$ is kept constant for all times $k > 0$, i.e., it is not computed at each time k. Also the norm of F as a function of time for the state feedback MPC and static state feedback controller is given in Fig. 5.4.

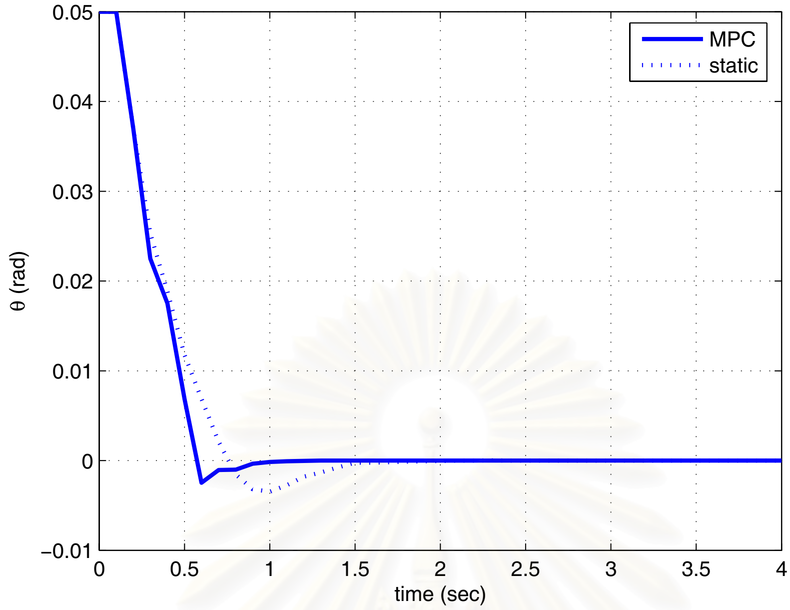


Figure 5.2: Time response of the angular positioning system, θ.

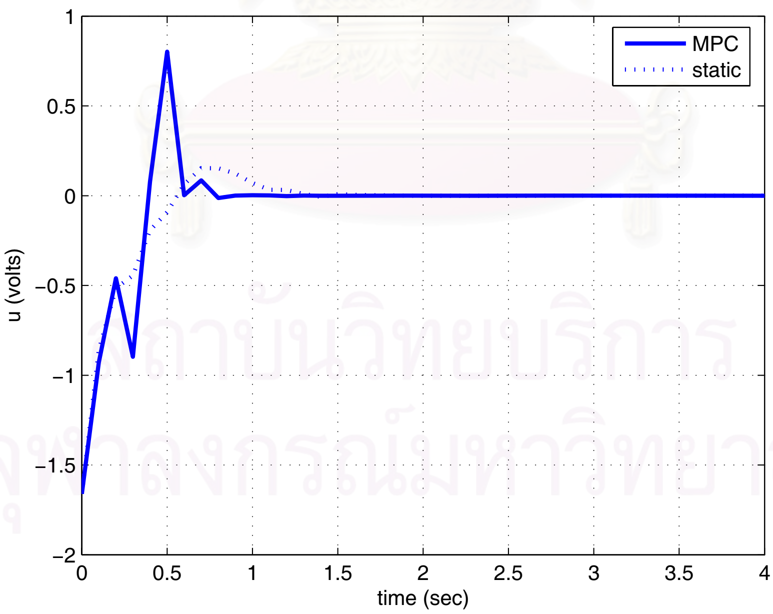


Figure 5.3: Control input of the angular positioning system, u.

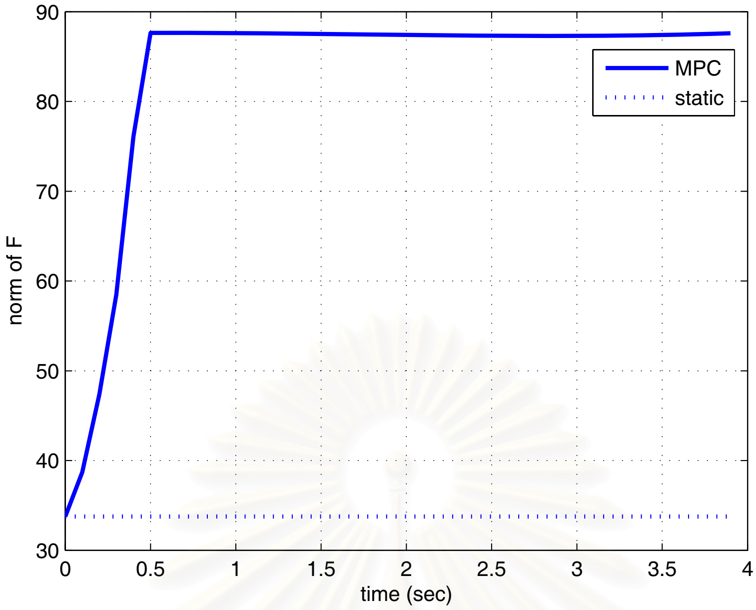


Figure 5.4: Norm of the feedback matrix F as a function of time.

5.2 Distillation Column

Consider a distillation column which has the following transfer function matrix [30]

$$\begin{bmatrix} x_D \\ x_B \end{bmatrix} = \begin{bmatrix} \frac{34}{54s+1} & \frac{-44.7}{114s+1} \\ \frac{31.6}{78s+1} & \frac{-45.2}{42s+1} \end{bmatrix} \begin{bmatrix} L \\ V \end{bmatrix}$$

where

$$\begin{aligned} x_D &- \text{distillate product composition [mole fraction]}, \\ x_B &- \text{bottom product composition [mole fraction]}, \\ L &- \text{reflux flow [kmol/min]}, \\ V &- \text{boilup flow [kmol/min]}. \end{aligned}$$

Assume that for each transfer function, standard deviation for both the gain and the time constant is $\sigma = 1\%$. The model is discretized using a sampling time of $T = 2$ min and given in terms of perturbation variables as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_p p(k), \\ y(k) &= Cx(k), \\ q(k) &= C_q x(k) + D_{qu} u(k), \\ p(k) &= (\Delta q)(k), \end{aligned}$$

where

$$A = \begin{bmatrix} 1-\frac{T}{54} & 0 & 0 & 0 \\ 0 & 1-\frac{T}{78} & 0 & 0 \\ 0 & 0 & 1-\frac{T}{114} & 0 \\ 0 & 0 & 0 & 1-\frac{T}{42} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{34T}{54} & 0 \\ \frac{31.6T}{78} & 0 \\ 0 & \frac{-44.7T}{114} \\ 0 & \frac{-45.2T}{42} \end{bmatrix},$$

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

$$C_q = \theta_1 \begin{bmatrix} \frac{T}{54} & 0 & 0 & 0 \\ 0 & \frac{T}{78} & 0 & 0 \\ 0 & 0 & \frac{T}{114} & 0 \\ 0 & 0 & 0 & \frac{T}{42} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_{qu} = \theta_2 \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{34T}{54} & 0 \\ \frac{31.6T}{78} & 0 \\ 0 & \frac{-44.7T}{114} \\ 0 & \frac{-45.2T}{42} \end{bmatrix},$$

with $\theta_1 = \max(|\frac{1}{1+\sigma} - 1|, |\frac{1}{1-\sigma} - 1|)$ and $\theta_2 = \max(|\frac{1-\sigma}{1+\sigma} - 1|, |\frac{1+\sigma}{1-\sigma} - 1|)$, and $\Delta = \text{diag}(\delta_1, \ldots, \delta_8)$ with $-1 \leq \delta_i \leq 1,\ i = 1, \ldots, 8$. The input constraints are $|u_1(k+i|k)| \leq 0.05$ and $|u_2(k+i|k)| \leq 0.05$.

The output feedback RCMPC scheme in Theorem 3.4 is applied to this system. We specify the design parameters $\mathcal{Q} = \text{diag}(1,1,1,1)$, $\mathcal{R} = 2 \times 10^{-5}\text{diag}(1,1)$ and $\rho = 0.95$, which satisfy the robust criterion (3.23). We choose a sequence of nine state vectors along the one dimensional subspace where

$$x_1 = x_3 \in [1,\ 0.5,\ 0.3,\ 0.2,\ 0.1,\ 0.07,\ 0.045,\ 0.033,\ 0.01] \quad \text{and} \quad x_2 = x_4 = 0.$$

Fig. 5.5 shows the intersection between the nine hyper-ellipsoids (defined by $Q_i^{-1},\ i = 1, \ldots, 9$) in the $x_1 - x_3$ plane. The hyper-ellipsoids are constructed one inside another in the state space and asymptotically stable invariant. Fig. 5.6 shows the norm of the off-line state feedback gains F_i along the chosen one dimensional subspace in logarithmic scale. As i increases, the norm of matrix F_i is larger and larger since the input constraints impose lesser and lesser limits on the feedback gain.

Given an initially perturbed state $x(0) = \begin{bmatrix} 0.05 & 0 & 0.05 & 0 \end{bmatrix}^T$, Fig. 5.7 shows that the combined control law F and the estimator robustly stabilizes the closed-loop system with each element of Δ block time varying randomly within $[-1, 1]$. Also in this figure, the open loop response of the system are shown with the dashed lines. The two control inputs are depicted in Fig. 5.8. Finally, Fig. 5.9 shows the convergence of the error between the estimated state and the actual state. The total time consumed for the off-line computations is 13.5 seconds whereas the time for the on-line simulation is 0.0938 seconds.

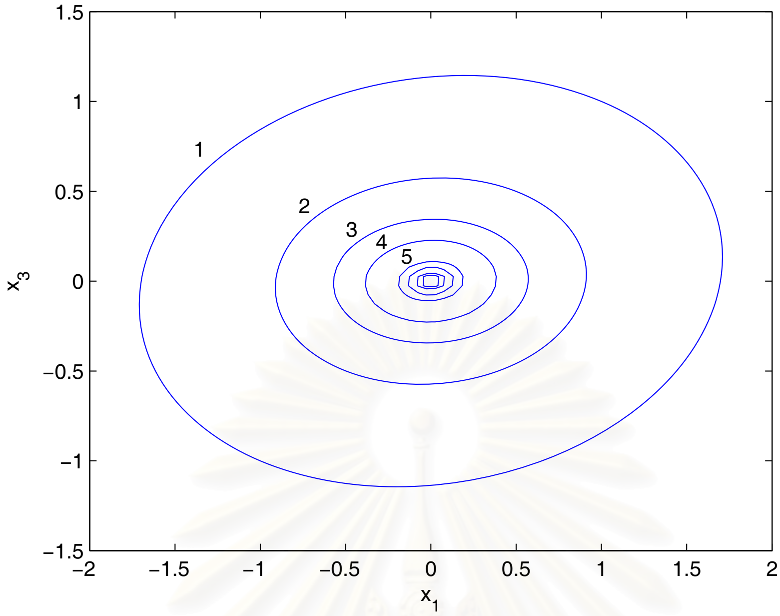


Figure 5.5: Intersection between the nine hyper-ellipsoids (defined by Q_i^{-1}) in the $x_1 - x_3$ plane.

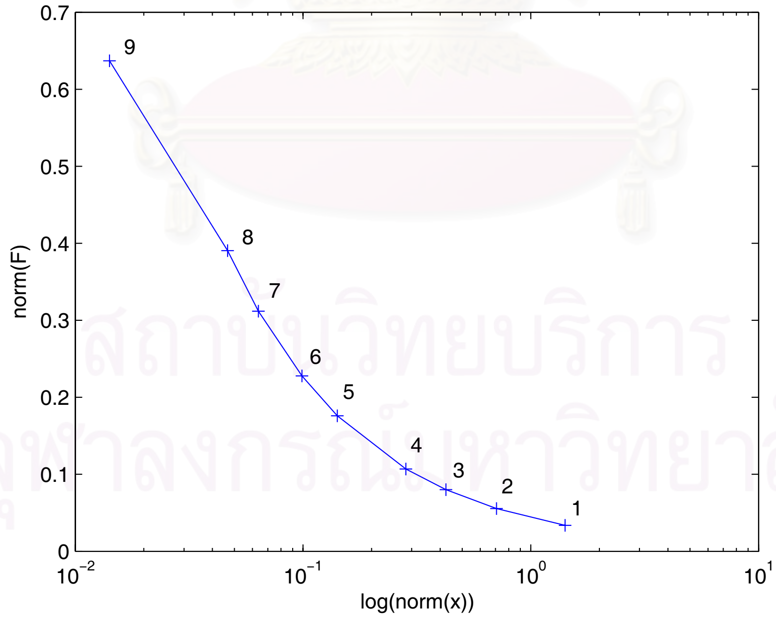


Figure 5.6: Norm of the off-line control law F.

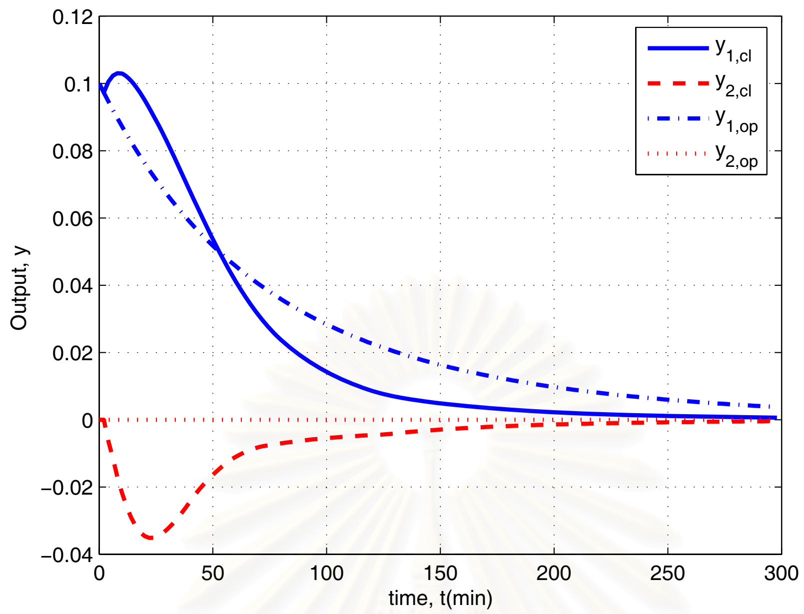


Figure 5.7: Closed-loop response of the distillation system.

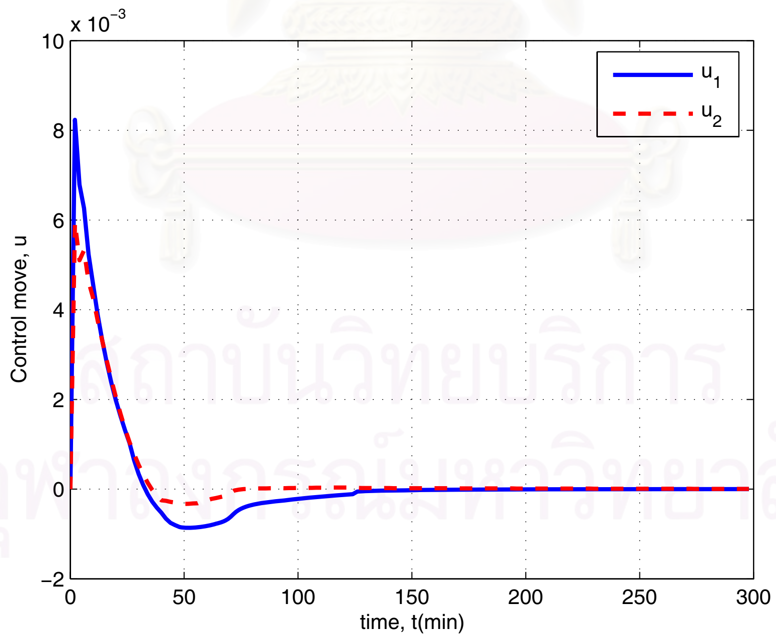


Figure 5.8: Control input of the distillation system.

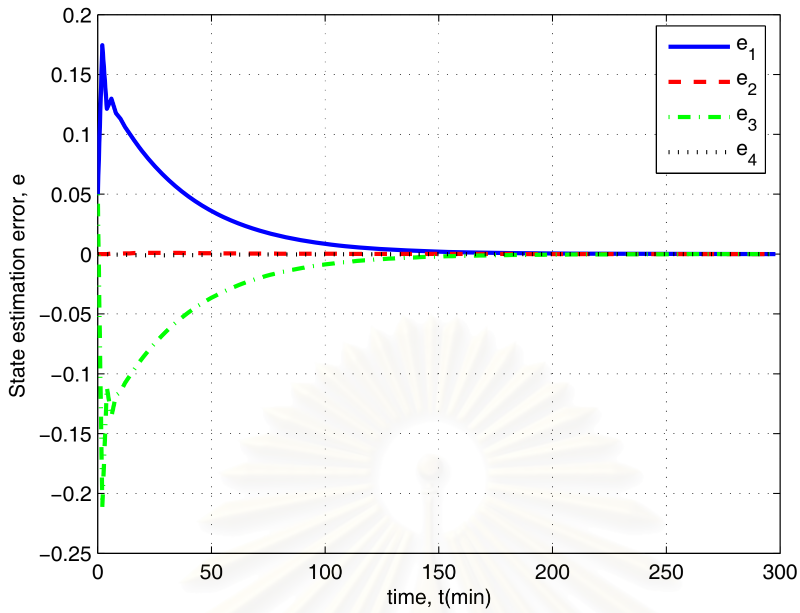


Figure 5.9: Estimation error of the distillation system.

5.3 Two-Mass-Spring System

Consider a two-mass-spring system as shown in Fig. 5.10. The state vector is given by

$$x = [x_1 \; x_2 \; x_3 \; x_4]^T,$$

where x_1 and x_2 are the positions of the first mass and second mass, respectively, x_3 and x_4 are the velocities associated with two masses. Using Euler first-order approximation with sampling time of 0.1 sec., we obtain a discrete-time linear time-varying model

$$x(k+1) = A(k)x(k) + Bu(k),$$

with

$$A(k) = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -0.1\frac{K(k)}{m_1} & 0.1\frac{K(k)}{m_1} & 1 & 0 \\ 0.1\frac{K(k)}{m_2} & -0.1\frac{K(k)}{m_2} & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \frac{0.1}{m_1} \\ 0 \end{bmatrix},$$

where m_1 and m_2 are the two masses and $K(k)$ is the spring constant. We choose system parameters and design specifications considered in [9], that are, $m_1 = m_2 = 1$, $\boldsymbol{\mathcal{Q}} = I$, $\boldsymbol{\mathcal{R}} = 1$ and $|u(k)| \leq 1$. Since there is one uncertain parameter $K(k)$, the polytopic uncertainty set $\boldsymbol{\Omega}$ has two vertices corresponding to the possible maximum and minimum value of $K(k)$. Subsequently, we apply the state feedback RCMPC algorithm using a PDLF as in Algorithm 4.1. To see the effectiveness of the developed technique, we compare the results with the ones obtained from the RCMPC algorithm

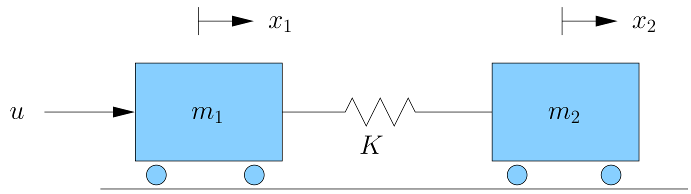


Figure 5.10: Two-mass-spring system.

using a single Lyapunov function and the static state feedback design. The experiments are comprised of two cases.

Case 1

Assume that $K(k) \in [0.5, 10]$. Fig. 5.11 shows the performance index γ achieved by two robust MPC algorithms. The index γ obtained by the RCMPC with a PDLF is smaller compared to the one obtained by the RCMPC with single Lyapunov function.The performance index obtained by the static state feedback law is computed only one time at $k = 0$ and has the value $\gamma = 279.65$.

Given an initial condition $x_0 = [1\ 1\ 0\ 0]^T$ and random samples of K, Fig. 5.12 and Fig. 5.13 show the time responses of the position and the velocity of the second mass, respectively. Time responses of the position and velocity of the first mass appear to be close to that of the second mass. Thus, the plots of x_1 and x_3 are omitted. In these figures, the solid line shows the response of the system with the RCMPC using a PDLF, the dashed line with the RCMPC using a single Lyapunov function, and the dotted line with the static state feedback control law. Apparently, the response corresponding to the RCMPC using a PDLF converges to zero fastest among these responses. In addition, Fig. 5.14 displays the control input of the closed-loop system. RCMPC with PDLF clearly utilizes the control input better than the other two methods. Fig. 5.15 depicts the norm of the state-feedback gains from three design methods. The state-feedback gain $F(k)$ is computed optimally at each time k satisfying the input constraint, as in the receding horizon controller. It is observed that the RCMPC using a PDLF gives smaller state-feedback gains compared to that of the RCMPC using single Lyapunov function. The static state-feedback controller does not recompute $F(k)$ at each sampling time, so it yields the most sluggish response. The actual CPU times required to compute the closed-loop responses for SLF and PDLF case were about 467.2 and 550.7 seconds, respectively while it took only 2.4 seconds for the static state feedback algorithm.

Case 2

Assume that $K(k) \in [0.5, K_{\max}]$. We vary $K_{\max}$ to show that the technique that uses a PDLF can control the system with a wider range of time-varying parametric uncertainty. Specifically, the synthesis conditions of the RCMPC algorithm using a single Lyapunov function become infeasible for $K_{\max} \geq 101$ whereas the synthesis conditions of the RCMPC algorithm using a PDLF is feasible for $K_{\max} = 200$.

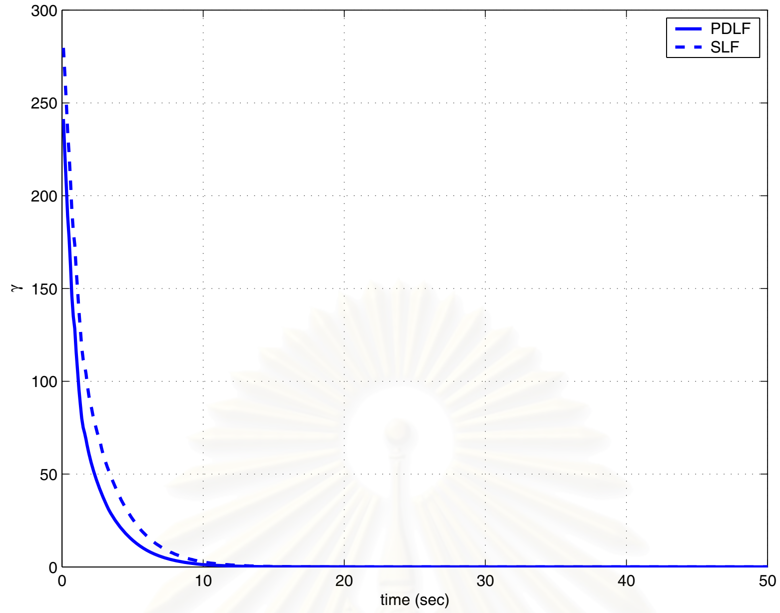


Figure 5.11: Performance index γ as a function of time.

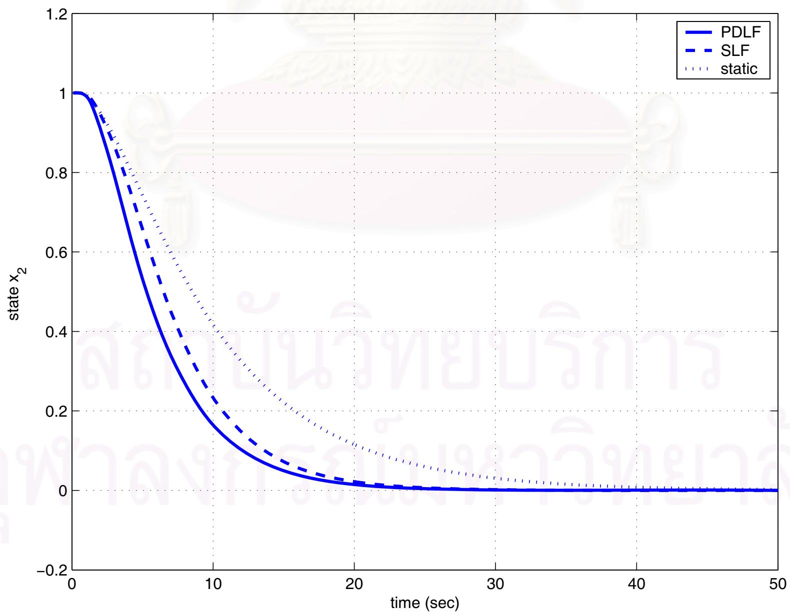


Figure 5.12: Time response of the second-mass position, x_2.

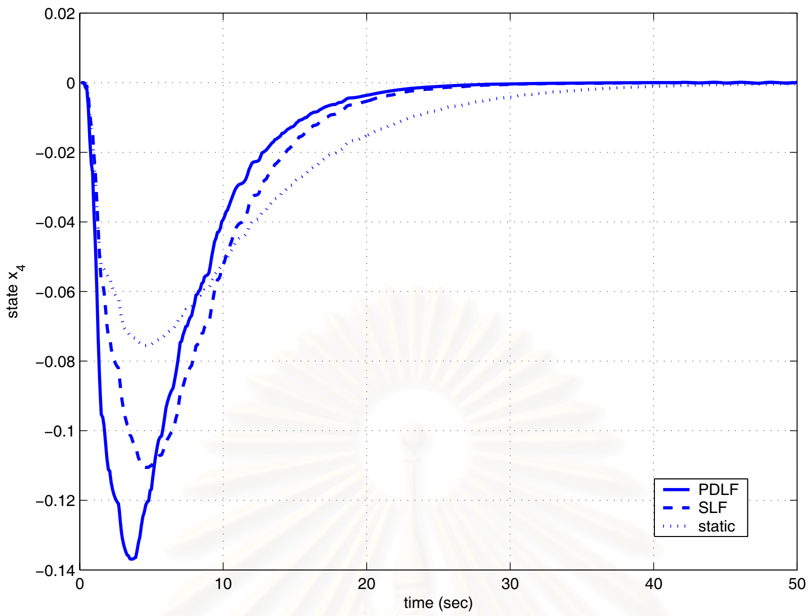


Figure 5.13: Time response of the second-mass velocity, x_4.

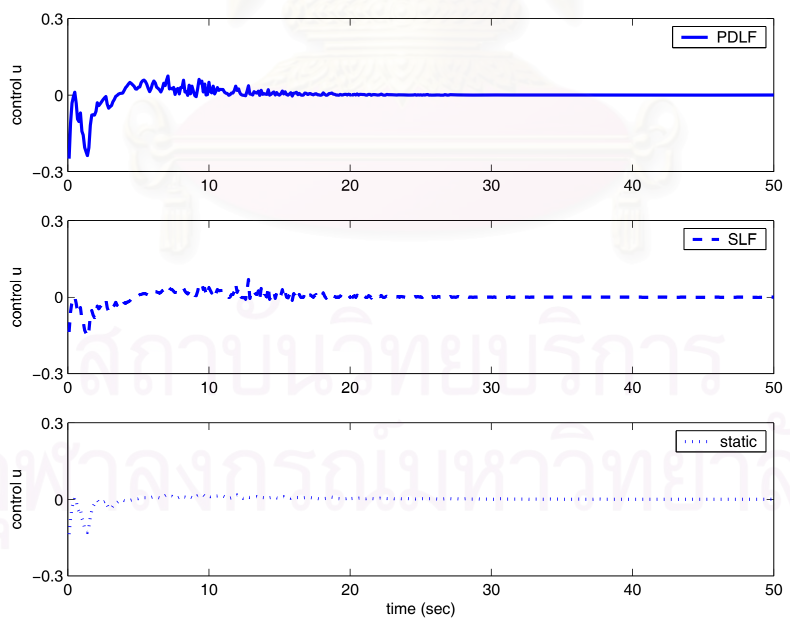


Figure 5.14: Control input u.

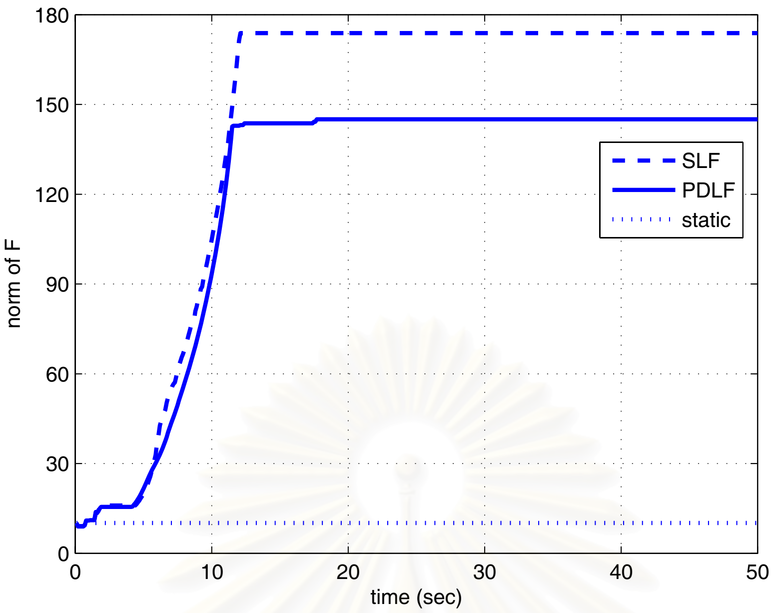


Figure 5.15: Norm of state-feedback gains F as a function of time.

5.4 Non-Isothermal CSTR

Consider the following linearized model derived for a single, non-isothermal CSTR [20]

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx, \end{aligned}$$

where x is a vector of the reactor concentration and temperature, u is the constrained coolant flow, and y is the reactor temperature (see Fig. 5.16). The matrices A, B and C are given by:

$$A = \begin{bmatrix} \frac{F}{V} - k_0 e^{\frac{E}{RT_s}} & -\frac{E}{RT_s^2} k_0 e^{-\frac{E}{RT_s}} C_{As} \\ \frac{-\Delta H_{rxn} k_0 e^{\frac{E}{RT_s}}}{\rho C_p} & -\frac{F}{V} - \frac{UA}{V\rho C_p} - \Delta H_{rxn} \frac{E}{\rho C_p R T_s^2} k_0 e^{-\frac{E}{RT_s}} C_{As} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ -2.098 \times 10^5 \frac{T_s - 365}{V\rho C_p} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

where $F = 1$ m^3/min, $V = 1$ m^3, $k_0 = 10^9 - 5 \times 10^9$ min^{-1}, $\frac{E}{R} = 8.330.1$ K, $-\Delta H_{rxn} = 10^7 - 5 \times 10^7$ cal/mol, $\rho = 10^6$ g/m^3, $UA = 5.34 \times 10^6$ cal/K, and $C_p = 1$ cal/(gK). Note that A and B depend on operating conditions. We will concentrate on this linearized model at steady state $T_s = 394$ K and $C_{As} = 0.265$ kmol/m^3 under the uncertain parameters k_0 and $-\Delta H_{rxn}$. The model is discretized using a sampling time of 0.15 min and given in terms of perturbation variables as follows

$$x(k+1) = \begin{bmatrix} 0.85 - 0.0986\alpha(k) & -0.0014\alpha(k) \\ 0.9864\alpha(k)\beta(k) & 0.0487 + 0.01403\alpha(k)\beta(k) \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ -0.912 \end{bmatrix} u(k),$$
$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

where $1 \leq \alpha(k) = k_0/10^9 \leq 5$ and $1 \leq \beta(k) = -\Delta H_{rxn}/10^7 \leq 5$. The polytopic uncertainty set has four vertices $\Omega = \text{Co}\{A_1, A_2, A_3, A_4\}$. The nominal model is the average of the four vertices. The input constraint is $|u(k+i|k)| \leq 0.5$ m^3/min.

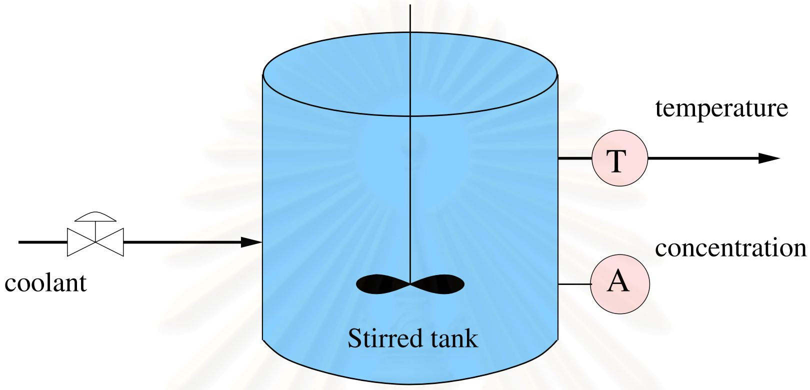


Figure 5.16: Single non-isothermal CSTR.

We will apply the output feedback RCMPC law to this system with two cases: using a single Lyapunov function (Theorem 3.4) and using a parameter-dependent Lyapunov function (Theorem 4.5). Both methods employ an off-line approach for controller design which gives a sequence of explicit control laws. We specify the design parameters

$$\mathcal{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathcal{R} = 2 \times 10^{-5} \quad \text{and} \quad \rho = 0.95,$$

which satisfy the robust criteria (3.22) and (4.39). We choose a sequence of ten state vectors along the x_1 axis

$$x_1^{set} = [1, 0.5, 0.3, 0.2, 0.15, 0.1, 0.07, 0.05, 0.035, 0.01].$$

Fig. 5.17(a) shows the ellipsoids defined by Q_i^{-1} for all ten states for the case of using a single Lyapunov function. Fig. 5.17(b) shows the ellipsoids for the case of using a PDLF. Similarly, Fig. 5.18 plots the norm of the off-line state feedback matrices F_i along the x_1 axis in logarithmic scale. In this particular example, the norm of matrices F_i are almost the same. In Fig. 5.19, the performance index γ corresponding to the given states is shown for both cases. The γ obtained by using a PDLF is much smaller than that obtained by using a single Lyapunov function, as expected.

Given an initially perturbed state $x(0) = \begin{bmatrix} 0.1 \\ 2 \end{bmatrix}$, Fig. 5.20 and Fig. 5.21 show the responses of the system. In both figures, the solid line corresponds to the use of a PDLF whereas the dotted line corresponds to the use of a single Lyapunov function. It is clear that the proposed method give better performance than the existing method. Also, the estimated errors for both cases are plotted in Fig. 5.22. The estimated state in the PDLF case converges to the actual state faster than that in the SLF case. The total times required to compute the RCMPC algorithms with PDLF and SLF techniques are 20.7 seconds and 10.1 seconds, respectively.

5.5 Conclusions

The objective of examples 5.1 and 5.2 is to illustrate the design and implementation of the RCMPC method with a SLF. Simulation results showed that the method is applicable both for the polytopic uncertain model and norm-bound uncertain model with both control structures, namely state feedback and output feedback. In fact, the PDLF technique had been applied to the example of an angular positioning system to compare with the SLF technique. However, since the uncertain parameter has little effect on the system behavior, simulation results showed not much improvement of the proposed method, and, therefore, were omitted here. In the example of the distillation column, some relaxations had been made to change the system model to polytopic paradigm so that the PDLF technique could be employed. Unfortunately, the solution of the convex optimization problem (4.33) does not exist. It motivates the need for synthesizing a RCMPC law with PDLF that is valid for norm-bound uncertain model.

Examples 5.3 and 5.4 give a good demonstration how the time response of the system can be improved if using the PDLF technique presented in chapter 4. This is not surprising since the quadratic Lyapunov function used to guarantee the robust stability and robust performance varies with the change of system parameters. In the method that uses a SLF, the Lyapunov matrix is always constant for all time-varying parametric uncertainties, and hence, resulting in conservative results.

In all simulations, the uncertain parameters were generated by using the command 'rand' in MATLAB, i.e., they are uniformly distributed random numbers. We conjecture that the system performance when using the RCMPC law with a PDLF will be much improved if we can generate the uncertain parameters with various rate of variation.

On the other hand, a weakness of quadratic stability is that it guards against arbitrarily fast parameter variations. As a result, this condition tends to be very conservative for constant or slowly-varying parameters. In case of time-invariant uncertain systems, a modified algorithm of RCMPC with multiple Lyapunov functions was derived in the work by Cuzzola et al. [22]. A method to construct a PDLF that takes into account the bounded rate of uncertain parameters, thus providing a smooth transition between time-invariant parameters and arbitrarily fast parameter variations may suggest further investigation.

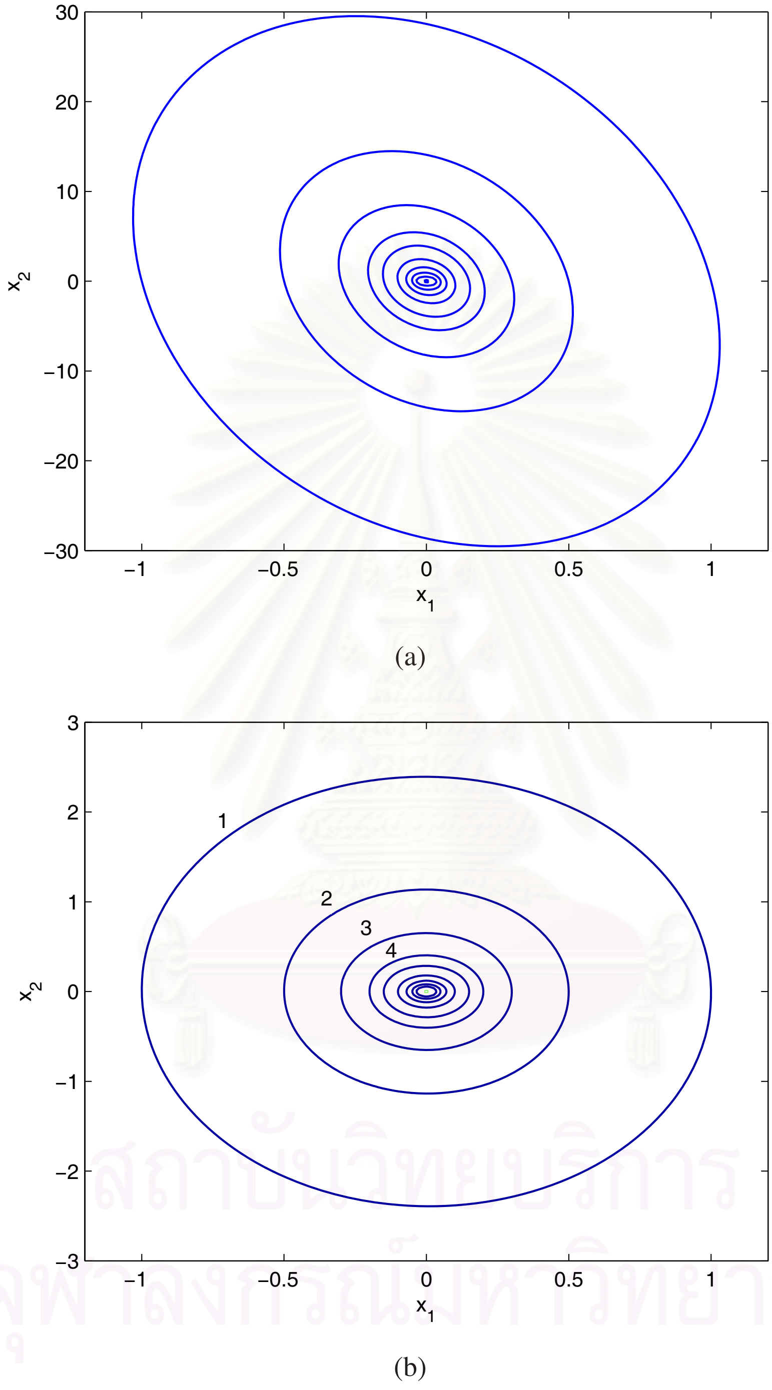


Figure 5.17: Ellipsoids defined by Q_i^{-1} for given ten states: (a) using a single Lyapunov function; (b) using a parameter-dependent Lyapunov function.

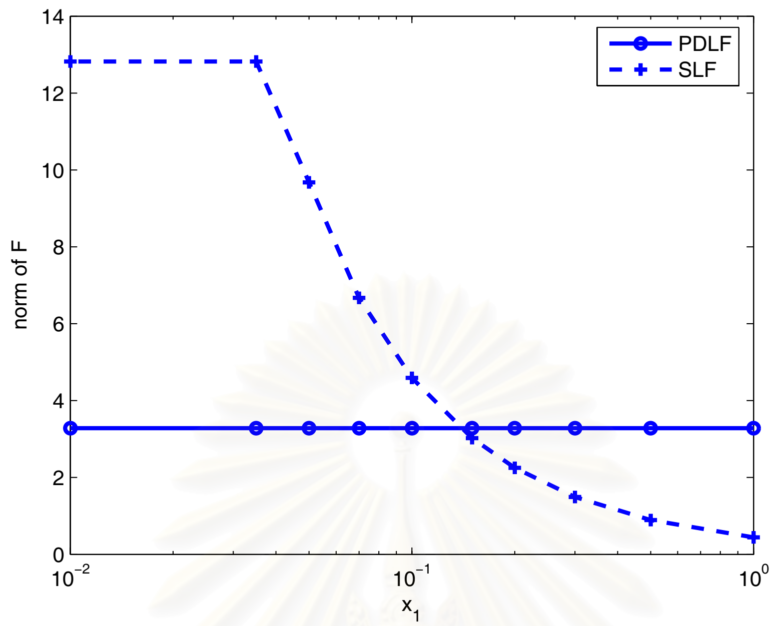


Figure 5.18: Norm of off-line state feedback matrices F_i for given ten states.

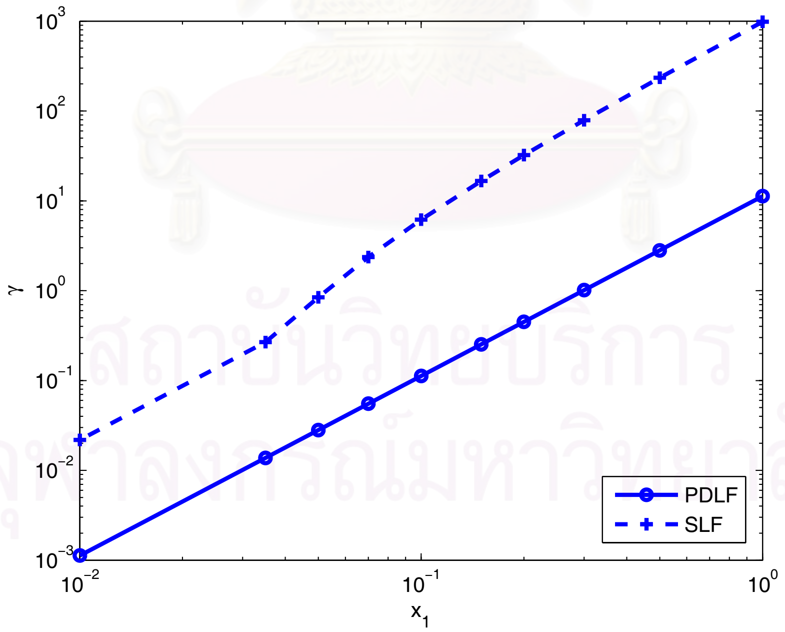


Figure 5.19: Performance index γ for given ten states.

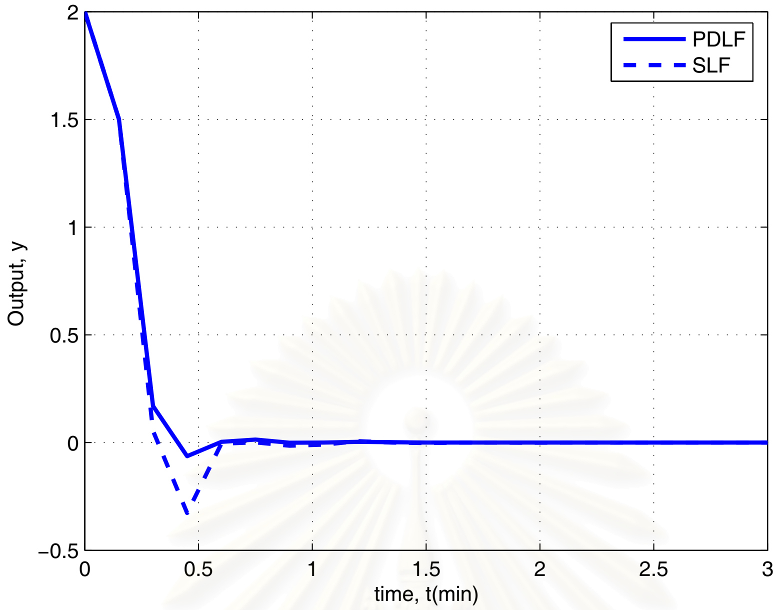


Figure 5.20: Time response of the reactor temperature, y.

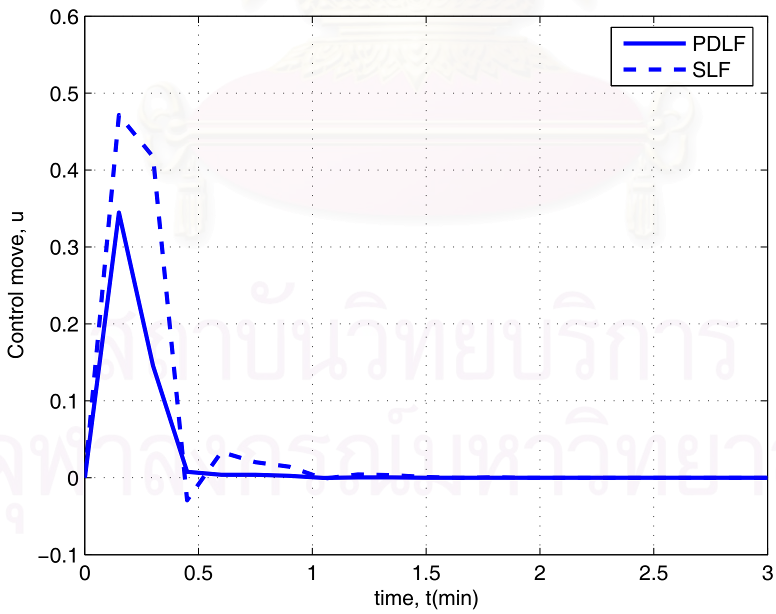
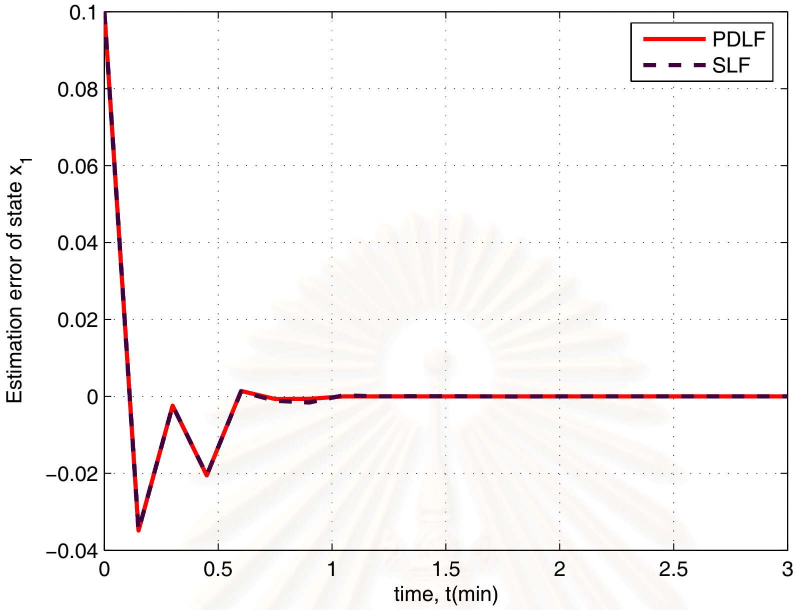
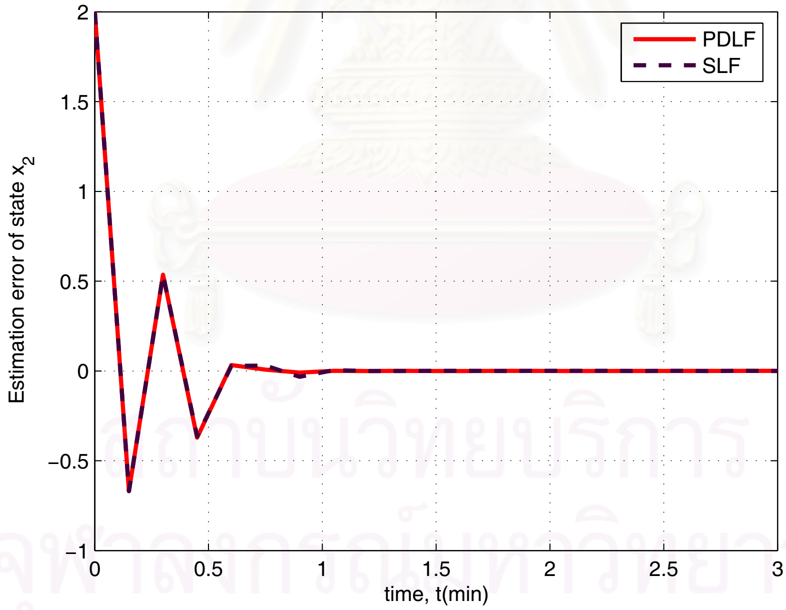


Figure 5.21: Control input of the CSTR system, u.



(a)



(b)

Figure 5.22: Estimation error of the CSTR system: (a) error of state x_1; (b) error of state x_2.

Chapter 6

CONCLUSIONS

6.1 Summary of Results

The RCMPC synthesis for uncertain LTV systems was extensively discussed in this thesis. Based on previous work done by Kothare et al. [9], Wan and Kothare [20, 21] and others, an approach for the design of a predictive control that ensures robust stability in the presence of input and output constraints was given. However, different from the pre-existing method that uses a single Lyapunov function, the RCMPC algorithm presented in this thesis employs a PDLF which corresponds to the vertices of the polytopic uncertainty. Several applications both to mechanical and chemical systems show that the proposed design technique provides improved performance and less conservative results. Moreover, searching for an RCMPC law is equivalent to solving a number of LMI problems which current available softwares can handle with extreme efficiency. To summarize the thesis, we highlight main topics in the following.

In chapter 2, a basic knowledge with some important concepts and tools to be used throughout the thesis was presented. The principle of Model Predictive Control was introduced in section 2.1. The mathematical representations of the uncertainties were discussed in section 2.3. Section 2.4 gave a brief description of LMI theory and Lyapunov stability for discrete-time systems was revised by section 2.4.

Chapter 3 introduced the design method for RCMPC, which employs a single Lyapunov function to guarantee robust stability of the closed-loop system. As described in section 3.1, the state feedback control law can be obtained by solving the on-line convex optimization over LMI constraints. This approach was then used in section 3.2 to initialize another off-line strategy making use of the concept of asymptotically stable invariant ellipsoid. Moreover, output feedback scheme can be developed by combining the off-line state feedback RCMPC with a state estimator. This combination leads to the need of off-line solution of problems involving LMIs, as discussed in section 3.3.

In parallel to chapter 3, RCMPC with a PDLF was discussed with details in chapter 4. This method is applicable for polytopic uncertain systems and provides less conservative results. Section 4.1 presented the state feedback RCMPC with a PDLF. The Lyapunov function is quadratic in the system state and depends in a polytopic way on the system uncertain parameters. After that, an off-line approach for state feedback RCMPC with a PDLF was introduced in section 3.2 . Section 4.3 offered an observer-based RCMPC scheme that ensures the robust stability of the augmented closed-loop system.

Finally, four different examples illustrated the RCMPC procedures in chapter 5, together with an extensive analysis. The first two examples based on an angular positioning system and a distillation column were given to demonstrate RCMPC policy with a SLF. In section 5.3, an example of a two-mass-spring system was presented. The effectiveness of the state feedback RCMPC with a PDLF was compared to the one with a SLF, and to the static state feedback law. The synthesis of the output feedback RCMPC with a PDLF was then applied to the example of a non-isothermal CSTR, with its behavior demonstrated and analyzed.

Some evaluations and discussions were made at the end of every chapter. The thesis ends with some extensions and potential improvements of the proposed technique.

6.2 Recommendations

6.2.1 Possible Extensions

The presentation in this thesis has been restricted to the infinite horizon regulator with zero target. There are some possible extensions to several standard problems encountered in practice.

Reference trajectory tracking

In optimal tracking problems, the system output is required to track a reference trajectory $y = C_t x_t(k)$, where the reference state x_t is computed from the equation

$$x_t(k+1) = A_t x_t(k), \quad x_t(0) = x_{t0}.$$

The choice of $J_{LQ}(k)$ for the robust trajectory tracking objective (3.1) is

$$\begin{aligned} J_{LQ}(k) = & \sum_{i=0}^{\infty} \big\{ [Cx(k+i|k) - C_t x_t(k+i)]^T \boldsymbol{\mathcal{Q}} [Cx(k+i|k) - C_t x_t(k+i)] \\ & + u(k+i|k)^T \boldsymbol{\mathcal{R}} u(k+i|k) \big\}. \end{aligned}$$

The plant dynamics can be augmented by the reference trajectory dynamics to reduce the robust trajectory tracking problem (with input and output constraints) to the standard form as in sections 3.1 and 4.1.

Constant setpoint tracking

Even though all the formulations in this thesis are intended for uncertain LTV systems, they can be applied to uncertain linear time-invariant case, admitting some conservatism. For LTI systems, the desired equilibrium state may be a constant point $x_s,\ u_s$ (called the *setpoint*) in state space, different from the origin. Consider (2.6), which is now assumed to represent an uncertain LTI system, i.e. $[A \ \ B] \in \boldsymbol{\Omega}$ are *constant* unknown matrices. Suppose that the system output y is required to track the target vector y_s, by moving the system to the setpoint $x_s,\ u_s$, where

$$x_s = Ax_s + Bu_s, \quad y_s = Cx_s.$$

Assume that x_s, u_s and y_s are feasible, i.e. they satisfy the imposed constraints. The choice of $J_{LQ}(k)$ for the robust setpoint tracking objective in the optimization (3.1) is

$$\begin{aligned} J_{LQ}(k) = \sum_{i=0}^{\infty} \big\{ & [Cx(k+i|k) - Cx_s]^T \boldsymbol{\mathcal{Q}} [Cx(k+i|k) - Cx_s] \\ & + [u(k+i|k) - u_s]^T \boldsymbol{\mathcal{R}} [u(k+i|k) - u_s] \big\}. \end{aligned}$$

This problem can be reduced to the standard form as in sections 3.1 and 4.1 by defining a shifted state $\hat{x}(k) = x(k) - x_s$, a shifted input $\hat{u}(k) = u(k) - u_s$, and a shifted output $\hat{y}(k) = y(k) - y_s$.

Disturbance rejection

In all practical applications, some disturbances invariably enters the system and hence it is meaningful to study its effect on the closed-loop response. Let an unknown disturbance $e(k)$, having the property $\lim_{k\to\infty} e(k) = 0$ enter the system as follows

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= Cx(k), \\ \big[A(k) \quad B(k)\big] &\in \boldsymbol{\Omega}. \end{aligned} \tag{6.1}$$

A simple example of such a disturbance is any energy-bounded signal $(\sum_{i=0}^{\infty} e(i)^T e(i) < \infty)$. Assume that the state of the system $x(k)$ is measurable, we would like to solve the optimization problem (3.1). If the predicted states of the system satisfy the equation

$$\begin{aligned} x(k+i+1|k) &= A(k+i)x(k+i|k) + B(k+i)u(k+i|k), \\ \big[A(k+i) \quad B(k+i)\big] &\in \boldsymbol{\Omega}. \end{aligned} \tag{6.2}$$

Then, as discussed in sections 3.1 and 4.1, we can derive an upper bound of the robust performance objective (3.1). The problem of minimizing this upper bound with a state feedback control law $u(k+i|k) = F(k)x(k+i|k),\ i > 0$, at the same time satisfying constraints on the control input and plant output, can be reduced to a linear objective minimization as in Theorem 3.2.

Systems with delays

Consider the following uncertain discrete-time LTV system with delay elements, described by the equations

$$\begin{aligned} x(k+1) &= A_0(k)x(k) + \sum_{i=1}^{m} A_i(k)x(k-\tau_i) + B(k)u(k-\tau), \\ y(k) &= Cx(k), \end{aligned} \tag{6.3}$$

with

$$\big[A_0(k) \quad A_1(k) \quad \dots \quad A_m(k) \quad B(k)\big] \in \boldsymbol{\Omega}.$$

Assume without loss of generality that the delays in the system satisfy $0 < \tau < \tau_1 < \cdots < \tau_m$. At sampling time $k \geq \tau$, we would like to design a state feedback control law $u(k+i-\tau|k) =$

$Fx(k+i-\tau|k),\ i \geq 0$, to minimize the following modified infinite horizon robust performance objective

$$J_{WC}(k) = \max_{[A_0(k+i)\ \dots\ A_m(k+i)\ \ B(k+i)]\in\mathbf{\Omega},\ i\geq 0} J_{LQ}(k), \tag{6.4}$$

where

$$J_{LQ}(k) = \sum_{i=0}^{\infty} \left\{ x^T(k+i|k)\boldsymbol{\mathcal{Q}}x(k+i|k) + u^T(k+i-\tau|k)\boldsymbol{\mathcal{R}}u(k+i-\tau|k) \right\},$$

subject to input and output constraints. Defining the augmented state

$$w(k) = \begin{bmatrix} x(k)^T & x(k-1)^T & \dots & x(k-\tau)^T & \dots & x(k-\tau_1)^T & \dots & x(k-\tau_m)^T \end{bmatrix}^T,$$

which is assumed to be measurable at each time $k \geq \tau$, we can derive an upper bound on the robust performance objective (6.4) as in section 3.1. The problem of minimizing this upper bound with the state feedback law $u(k+i-\tau|k) = Fx(k+i-\tau|k),\ k \geq \tau,\ i \geq 0$, subject to constraints on the control input and plant output, can then be reduced to a linear objective minimization as in Theorem 3.2. Note, however, that the appropriate choice of the function $V(w(k))$ satisfying an inequality of the form (3.3) is

$$\begin{aligned} V(w(k)) = {} & x(k)^T P_0 x(k) + \sum_{i=1}^{\tau} x(k-i)^T P_\tau x(k-i) + \sum_{i=\tau+1}^{\tau_1} x(k-i)^T P_{\tau_1} x(k-i) \\ & + \dots + \sum_{i=\tau_{m-1}+1}^{\tau_m} x(k-i)^T P_{\tau_m} x(k-i) \\ = {} & w(k)^T P w(k), \end{aligned}$$

where P is appropriately defined in terms of $P_0, P_\tau, P_{\tau_1}, \dots, P_{\tau_m}$. The motivation for this modified choice of function V comes from Feron et al. [31], where such a V is defined for continuous time systems with delays, and is referred to as a modified Lyapunov-Krasovskii (MLK) functional.

Construction of invariant sets

The concept of invariant set is important in the off-line approach. There are various ways to construct the invariant sets depending on the objectives. For example, one may construct an ellipsoid by requiring it to encompass a 'ball' of radius r (see the algorithm in the work by Wan and Kothare [32]).

Alternatively, one can also define a polytope and calculate $\{Y, Q\}$ at all the vertices of the polytope. The linear combination of $\{Y, Q\}$ will provide a feasible solution of $\{Y, Q\}$ for any state within the polytope. For further reading, the interested reader is referred to the similar algorithm proposed by Wan and Kothare [33].

One may even try to define a region where linear combination of $\{Y, Q\}$ always provides optimal control law $F(k) = YQ^{-1}$, and partition the state space into different regions with linear combination between different $\{Y, Q\}$'s (see the paper by Bemporad et al. [34]).

6.2.2 Future Work

The controller design using a PDLF for linear systems described by a norm-bound uncertain model (2.8)-(2.9) can be a subject of future research. The main idea is to consider the following Lyapunov function

$$V(x(k|k), \lambda(k)) = x(k|k)^T P(\lambda(k)) x(k|k), \quad P(\lambda(k)) = P(\lambda(k))^T > 0,$$

where

$$P(\lambda(k)) = \begin{bmatrix} I \\ \Delta_a(\lambda(k)) \end{bmatrix}^T P_a(k) \begin{bmatrix} I \\ \Delta_a(\lambda(k)) \end{bmatrix},$$
$$\Delta_a(\lambda(k)) = (I - \Delta(\lambda(k))D)^{-1} \Delta(\lambda(k))C,$$

for all $\Delta(\lambda(k)) \in \Delta$. This function can be an upper bound of the robust performance (3.2). The problem is then redefined to the trace minimization of P_a. The motivation of this method is to apply to the example of the distillation column that was mentioned in chapter 5.

REFERENCES

[1] D. W. Clarke, "Application of generalized predictive control to industrial processes," *IEEE Control Syst. Mag.*, 122, (1998): 49–55.

[2] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: application to industrial processes," *Automatica*, 14, 2, (1978): 413–428.

[3] J. Richalet, "Industrial applications of model based predictive control," *Automatica*, 29, 5, (1993): 1252–1274.

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

[5] A. Bemporad and M. Morari, "Robust model predictive control: a survey," *Robustness in Indentification and Control*, 245, (1999): 207–226.

[6] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, England, 2002.

[7] T. A. Badgwell, "Robust model predictive control of stable linear systems," *Int. J. Control*, 68, 4, (1997): 797–818.

[8] A. Zheng, "Robust stability analysis of constrained model predictive control," *J. of Process Control*, 9, (1999): 271–278.

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

[10] Y. H. Lu and Y. Arkun, "Quasi min-max MPC algorithms for LPV systems," *Automatica*, 36, 4, (2000): 527–540.

[11] Z. Q. Zheng and M. Morari, "Stability of model predictive control with mixed constraints," *IEEE Trans. Aut. Control*, 40, 10, (1995): 1818–1823.

[12] J. H. Lee, M. Morari, and C. E. Garcia, "State space interpretation of model predictive control," *Automatica*, 30, 4, (1994): 707–717.

[13] J. C. Doyle, "Guaranteed margins for LQG regulators," *IEEE Trans. Aut. Control*, 23, 4, (1978): 756–757.

[14] J. C. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems," *Automatica*, 34, 8, (1989): 831–847.

[15] A. Zheng and M. Morari, "Robust control of linear time varying systems with constraints," *Int. J. Robust and Nonlinear Control*, 10, 13, (2000): 1063–1078.

[16] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice Hall, 1998.

[17] D. Henrion, S. Tarbouriech, and G.Garcia, "Output feedback robust stabilization of uncertain linear systems with saturating controls: An LMI approachh," *IEEE Trans. Aut. Control*, 44, 11, (1999): 2230–2237.

[18] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Interior-point polynomial methods in convex programming*. Philadelphia, PA: Studies in Applied Mathematics, SIAM, 1994.

[19] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *J. of Process Control*, 10, (2000): 363–385.

[20] Z. Wan and M. V. Kothare, "Robust output feedback model predictive control using off-line linear matrix inequalities," *J. of Process Control*, 12, (2002): 763–774.

[21] Z. Wan and M. V. Kothare, "An efficient off-line formulation of robust model predictive control using linear matrix inequalities," *Automatica*, 39, (2003): 837–846.

[22] F. A. Cuzzola, J. C. Geromel, and M. Morari, "An improved approach for constrained robust model predictive control," *Automatica*, 38, (2002): 1183–1189.

[23] W. Mao, "Robust stabilization of uncertain time-varying discrete systems and comments on "an improved approach for constrained robust model predictive control"," *Automatica*, 39, (2003): 1109–1112.

[24] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computer and Chemical Engineering*, 23, (1999): 667–682.

[25] R. W. Liu, "Convergent system," *IEEE Trans. Aut. Control*, 13, (1968): 384–391.

[26] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI control toolbox: for use with MATLAB*. The Mathworks, Inc., Natick, MA, 1995.

[27] M. C. de Olivera, J. Bernussou, and J. C. Geromel, "A new discrete-time robust stability condition," *Syst. Control Letters*, 37, (1999): 261–265.

[28] J. Daafouz and J. Bernussou, "Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties," *Syst. Control Letters*, 43, (2001): 355–359.

[29] M. V. Kothare and M. Morari, "Multiplier theory for stability analysis of anti-windup control systems," *Automatica*, 35, 5, (1999): 917–928.

[30] M. L. Luyben and W. L. Luyben, *Essentials of Process Control*. McGraw-Hill, 1997.

[31] E. Feron, V. Balakrishnan, and S. Boyd, "Design of stabilizing state feedback for delay systems via convex optimization," in *Proc. IEEE Conf. on Decision and Control*, (1992): 147–148.

[32] Z. Wan and M. V. Kothare, “Efficient robust constrained model predictive control with a time varying terminal constraint set,” *Syst. Control Letters*, 48, (2003): 375–383.

[33] Z. Wan and M. V. Kothare, “A two-level model predictive control formulation for stabilization and optimization,” in *Proc. American Control Conf.*, (2003): 5294–5299.

[34] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, 38, (2002): 3–20.

APPENDICES

Appendix

Matlab Source Code for Simulations of Controllers Design

A Program for Simulation of an Angular Positioning System

angular_response.m

```
%***************************************************************************************
% angular_response.m
% To simulate the response of an angular positioning system
% Using on-line state feedback RCMPC in comparison with static state feedback
%
% Tu Anh Do
%
% Files needed: rstatefeedc.m
% Update: 12/08/2005
%***************************************************************************************
close all;
clear all;
tic          % start clock
% Declare the vertices of polytopic model
Bsys = [0; 0.0787];
Amod1 = [1    0.1
         0    0.99];
Bmod1 = Bsys;
Amod2 = [1    0.1
         0     0 ];
Bmod2 = Bsys;

umax = 2;           % input constraint
Q1 = [1  0;  0  0]; % state weighting matrix
R = 0.00002;        % input weighting matrix
N = 40;             % number of iterations

% Compute x(k),u(k) and F(k) using state feedback RCMPC
xk = [0.05; 0];  % initial state
dataout = zeros(N,3);
for k = 0:(N-1)
    F = rstatefeedc(Amod1,Bmod1,Amod2,Bmod2,Q1,R,xk,umax);
    uk = F*xk;                       % compute u(k)
    dataout(k+1,:)=[k xk(1) uk];     % store the data
    if (xk(1)> 0.0001) || (xk(2)> 0.0001)
        Fk =F;
    end
    alpha = 0;
    while alpha < 0.1
        alpha = 10*rand;                 % uncertain parameter
    end
    Asys = [1        0.1
            0   1-0.1*alpha];            % dymamic matrix of system
    xk = (Asys+Bsys*Fk)*xk;              % compute x(k+1)
```

```
    normf(k+1)= norm(Fk);                    % compute norm of matrix F
    k   = k+1;
end
dataout(k+1,:)=[k xk(1) uk];
toc      % stop clock

% Compute x(k),u(k) and F(k) using static state feedback
xks = [0.05; 0];
F_static = rstatefeedc(Amod1,Bmod1,Amod2,Bmod2,Q1,R,xks,umax);
dataouts = zeros(N,3);
for k = 0:(N-1)
     uks = F_static*xks;                    % compute u(k)
     dataouts(k+1,:)=[k xks(1) uks];
     alpha = 0;
     while alpha < 0.1
         alpha = 10*rand;
     end
     Asys = [1        0.1
             0   1-0.1*alpha];
     xks = (Asys+Bsys*F_static)*xks;       % compute x(k+1)
     normfs(k+1)= norm(F_static);           % compute norm of matrix F_static
     k   =k+1;
end
dataouts(k+1,:)=[k xks(1) uks];

time = dataout(:,1)/10;
figure (1);                                 % Plot the response x(1)
plot(time,dataout(:,2),'b'); hold on;
plot(time,dataouts(:,2),'b:');
xlabel('time (sec)');
ylabel('\theta (rad)');
figure (2)                                  % Plot the input u
plot(time,dataout(:,3)); hold on;
plot(time,dataouts(:,3),':');
xlabel('time (sec)');
ylabel('u (volts)');
figure (3);                                 % Plot the norm of matrix F
plot(time(1:N),normf); hold on;
plot(time(1:N),normfs,':');
xlabel('time (sec)');
ylabel('norm of F');
```

rstatefeedc.m

```
%****************************************************************************************
% rstatefeedc.m
% Used in file: angular_response.m
% To determine the feedback matrix F of the state feedback RCMPC law
% by solving an LMI optimization
%
% Tu Anh Do
% Update: 12/08/2005
%****************************************************************************************
function F = rstatefeedc(A1,B1,A2,B2,Q1,R,xk,um)

n = length(A1); % number of state variables
p = 1;          % number of inputs
% Define the variables for the minimization problem
```

```
gamma = sdpvar(1,1);
Q = sdpvar(n,n);
Y = sdpvar(p,n);
% Declare the left-hand side of the LMIs
Q1s = sqrtm(Q1);
Rs = sqrtm(R);
I1 = eye(n);
I2 = eye(p);
F1 = [1 xk';xk Q];
F21 = [   Q        Q*A1'+Y'*B1'    Q*Q1s      Y'*Rs
       A1*Q+B1*Y      Q           zeros(n)    [0;0]
        Q1s*Q       zeros(n)     gamma*I1    [0;0]
        Rs*Y         [0 0]        [0 0]     gamma*I2];
F22 = [   Q        Q*A2'+Y'*B2'    Q*Q1s      Y'*Rs;
       A2*Q+B2*Y      Q           zeros(n)    [0;0] ;
        Q1s*Q       zeros(n)     gamma*I1    [0;0] ;
        Rs*Y         [0 0]        [0 0]     gamma*I2];
F3 = [um^2*I2   Y            % for input constraint
        Y'      Q];
% Set up the constraints for the minimization problem
constraint = set(Q>0)+set(F1>=0)+set(F21>=0)+set(F22>=0)+set(F3>=0);
% Solve the minimization problem
solvesdp(constraint,gamma);
% The minimizer
Y = double(Y);
Q = double(Q);
gamma = double(gamma);

% The state feedback matrix
F = Y*inv(Q);
```

B Program for Simulation of a Distillation Column

distillation.m

```
%*************************************************************************************
% distillation.m
% To simulate the response of a distillation column
% Using output feedback RCMPC
% by combining an off-line state feedback RCMPC together with an estimator
%
% Tu Anh Do
%
% Files needed: rstatefeed_offline.m, observer.m
% Update: 25/11/2005
%*************************************************************************************
close all; clear all;
tic          % start clock
% Declare the norm-bound uncertain model
T = 2;  %sampling time
sigma = 0.01; %standard deviation for the gain and time constant in the TF
theta1 = max(abs(1/(1+sigma)-1), abs(1/(1-sigma)-1));
theta2=max(abs((1-sigma)/(1+sigma)-1), abs((1+sigma)/(1-sigma)-1));
A  = [1-(T/54)      0          0          0
          0      1-(T/48)      0          0
          0         0      1-(T/114)      0
          0         0          0      1-(T/42)];
B = [34*T/54        0
```

```
       31.6*T/78          0
          0          -44.7*T/114
          0          -45.2*T/42  ];
Bp = [1   0   0   0   1   0   0   0
      0   1   0   0   0   1   0   0
      0   0   1   0   0   0   1   0
      0   0   0   1   0   0   0   1];
Cq = theta1*[T/54    0      0      0
                0   T/78    0      0
                0     0   T/114    0
                0     0     0    T/42
                0     0     0      0
                0     0     0      0
                0     0     0      0
                0     0     0      0 ];
Dqu = theta2*[  0           0
                0           0
                0           0
                0           0
             34*T/54        0
             31.6*T         0
                0      -44.7*T/114
                0      -45.2*T/42  ];
C = [1   0   1   0
     0   1   0   1];

umax = 0.05;            % input constraints
% Design parameters
Qw = diag([1 1 1 1]);
Rw = 0.00002*diag([1 1]);
rho = 0.95;
% Sequence of N states (N = 9)
x_1 = [1 0.5 0.3 0.2 0.1 0.07 0.045 0.033 0.01]; x_2 =  zeros(1,9);
x_3 = x_1;
x_4 = x_2;
N = 9; v = zeros(N,4);
for k = 1:N
    v(k,:) = [x_1(k)  x_2(k)  x_3(k)  x_4(k)];
    k = k+1;
end
% Set up a lookup table of (Qi,Fi)
dataout1 = zeros(4,4*N); % to store matrices Qi
dataout2 = zeros(2,4*N); % to store matrices Fi
for i =1:N
    [Q,F] = rstatefeed_offline(A,B,Bp,Cq,Dqu,Qw,Rw,v(i,:)' ,umax);
    dataout1(:,(4*i-3):4*i) = Q;
    dataout2(:,(4*i-3):4*i) = F;
end
% Obtain the estimator gain
Lp = observer(A,C,rho);
% Simulation
xk = [0.05; 0; 0.05; 0]; x_estk = [0 ; 0; 0 ; 0];
x_augk=[xk;x_estk];
Bpnorm = [Bp; zeros(4,8)];
for j = 1:1:150
    V = rand(3);
    Delta = diag(V(1:8));
    %open-loop
```

```
    qk = Cq*xk;
    pk = Delta*qk;
    y = C*xk;
    xk = A*xk+Bp*pk;
    y = C*xk;
    dataout3(j,:)=[y(1) y(2)];
    %closed-loop
    if x_estk == [0;0;0;0]
        n = N;
    else
        for m = 1:N
        if ( x_estk'*inv(dataout1(:,(4*m-3):4*m))*x_estk <= 1),continue, end
        n = m;
        break;
        end
    end
    if n < N
        Qn = dataout1(:,(4*n-3):4*n);
        Fn = dataout2(:,(4*n-3):(4*n));
        Qnn = dataout1(:,(4*(n+1)-3):4*(n+1));
        Fnn = dataout2(:,(4*(n+1)-3):4*(n+1));
        alpha = (1-(x_estk'*inv(Qnn)*x_estk))/(x_estk'*(inv(Qn) - inv(Qnn))*x_estk);
        Fk =  alpha*Fn+(1-alpha)*Fnn ;
    else
        Fk = dataout2(:,(4*N-3):(4*N));
    end
    uk = Fk*x_augk(5:8);
    yk = C*x_augk(1:4);
    dataout4(j,:) = [ j uk(1) uk(2) yk(1) yk(2)];
    dataout5(j,:) = [(x_augk(1)-x_augk(5)) (x_augk(2)-x_augk(6))
                     (x_augk(3)-x_augk(7)) (x_augk(4)-x_augk(8))];
    Anorm = [A      B*Fk
             Lp*C   A+B*Fk-Lp*C];
    Cqnorm = [Cq Dqu*Fk];
    q_augk = Cqnorm*x_augk;
    p_augk = Delta*q_augk;
    x_augk = Anorm*x_augk + Bpnorm*p_augk;
    x_estk = x_augk(5:8);
end
toc      % stop clock
% Plots
time = dataout4(:,1)*2 -2;
figure (1);            %plot the output y
plot(time,dataout4(:,4),'k'); hold on;
plot(time,dataout4(:,5),'r');hold on;
plot(time,dataout3(:,1),'k--'); hold on;
plot(time,dataout3(:,2),'r--');
xlabel('time, t(min)');
ylabel('Output, y ');
figure (2);            % plot the input u
plot(time,dataout4(:,2),'b'); hold on;
plot(time,dataout4(:,3),'b--');
xlabel('time (min)');
ylabel('control move, u');
% Plot the error
figure (3);            % plot the estimation error
plot(time,dataout5(:,1),'b'); hold on plot(time,dataout5(:,2),'r');
hold on plot(time,dataout5(:,3),'g'); hold on
```

```
plot(time,dataout5(:,4),'k');
xlabel('time, t(min)');
ylabel('State estimation error, e ');
```

rstatefeed_offline.m

```
%***************************************************************************************
% rstatefeed_offline.m
% Used in file: distillation.m
% To determine the  matrices Q and F of the output feedback RCMPC law
% by solving an LMI optimization
%
% Tu Anh Do
% Update: 25/11/2005
%***************************************************************************************
function [Q,F] = rstatefeed_offline(A,B,Bp,Cq,Dqu,Qw,Rw,xk,um)

n = length(A);     % number of state variables
p = 2;             % number of inputs
% Define the variables for the minimization problem
gamma = sdpvar(1,1);
Q = sdpvar(n,n);
Y = sdpvar(p,n);
X=sdpvar(p,p);
lambda = sdpvar(8,8);
% Declare the left-hand side of the LMIs
Qs = sqrtm(Qw);
Rs = sqrtm(Rw);
I1 = eye(n);
I2 = eye(p);

F1 = [1 xk';xk Q];

F2= [      Q            Y'*Rs         Q*Qs        Q*Cq'+Y'*Dqu'    Q*A'+Y'*B'
        Rs*Y        gamma*I2      zeros(2,4)      zeros(2,8)       zeros(2,4)
        Qs*Q        zeros(4,2)    gamma*I1        zeros(4,8)       zeros(4)
    Cq*Q+Dqu*Y      zeros(8,2)    zeros(8,4)      lambda           zeros(8,4)
      A*Q+B*Y       zeros(4,2)    zeros(4)        zeros(4,8)    Q-Bp*lambda*Bp'];
F3 = [X     Y              % input constraints
      Y'    Q];
X1 = X(1,1); X2 = X(2,2);
% Set up the constraints for the minimization problem
constraint = set(Q>0)+set(F1>=0)+set(F2 >=0)+set(F3>=0)
             +set(X1<=um^2)+set(X2<=um^2)+set(lambda>0);
% Solve the minimization problem
solvesdp(constraint,gamma);
% The minimizer
Y = double(Y);
gamma = double(gamma);
lambda =double(lambda);
Q=double(Q);

% The state feedback matrix
F = Y*inv(Q);
```

observer.m

```
%***************************************************************************************
```

```
% observer.m
% Used in file: distillation.m
% To determine the gain matrix Lp of the estimator
% by solving an LMI feasibility problem
%
% Tu Anh Do
% Update: 25/11/2005
%***************************************************************************************
function Lp = observer(A,C,rho)

n = length(A);     % number of state variables
p = 2;             % number of inputs
% Define the variables for the minimization problem
Q = sdpvar(n,n);
Y = sdpvar(n,p);
% Declare the left-hand side of the LMIs
F  = [rho^2*Q  Q*A-Y*C
      A'*Q-C'*Y'  Q];
constraint = set(Q>0)+set(F >=0);
% Solve the feasibility problem
solvesdp(constraint);
% The solution
Y = double(Y); Q = double(Q);

% The estimator gain
Lp = inv(Q)*Y;
```

C Program for Simulation of a Two-Mass-Spring System

mass_spring.m

```
%***************************************************************************************
% mass_spring.m
% To simulate the response of a two-mass-spring system
% Using state feedback RCMPC with a PDLF
% in comparison with a single Lyapunov function and static state feedback
%
% Tu Anh Do
%
% Files needed: rmpcpoly_mass.m, rmpcpoly_mass_Lya.m.
% Update: 18/04/2006
%***************************************************************************************
close all;
clear all;
m = 1;                    % unit masses
N = 500;                  % number of iterations
Kmin = 0.5; Kmax = 10;
for i=1:N
    K = 0;                % spring constant
    while K < Kmin
        K = Kmax*rand;    % K belongs to [Kmin Kmax]
    end
    Ka(i) = K;
  end
% Declare the vertices of the polytopic model
A1  = [   1              0           0.1      0
          0              1           0        0.1
      -0.1*Kmin/m    0.1*Kmin/m      1        0
```

```
        0.1∗Kmin/m     −0.1∗Kmin/m          0          1  ];
B1 = [0; 0; 0.1/m; 0]; A2  = [   1                    0               0.1
0
           0                    1              0       0.1
       −0.1∗Kmax/m      0.1∗Kmax/m          1          0
        0.1∗Kmax/m     −0.1∗Kmax/m          0          1  ];
B2 = B1;
% Input constraint
umax = 1;
% Weighting matrices in the objective function
Qw = eye(4); Rw = 1;
x = [1; 1; 0; 0];  % initial state

% A single Lyapunov function case
for i =1:N
    Asys  = [    1              0          0.1     0
                 0              1            0     0.1
              −0.1∗Ka(i)   0.1∗Ka(i)         1       0
               0.1∗Ka(i)  −0.1∗Ka(i)         0       1  ];
    Bsys = B1;
    % Compute u(k),x(k) and norm of F(k)
    [F,gamma] = rmpcpoly_mass(A1,B1,A2,B2,Qw,Rw,x,umax);
    u = F∗x;
    normF = norm(F);
    if (i>2)&&(normF − dataout(i−1,5)) < 10^(−2)
            normFd = dataout(i−1,5);
    else
            normFd = normF;
    end
    dataout(i,:)=[i x(2) u gamma normFd  x(4)];
    x = (Asys+Bsys∗F)∗x;
end

% A Parameter−dependent Lyapunov function  case
xn = [1; 1; 0; 0];  % initial state
for i =1:N
    Asys  = [    1              0          0.1     0
                 0              1            0     0.1
              −0.1∗Ka(i)   0.1∗Ka(i)         1       0
               0.1∗Ka(i)  −0.1∗Ka(i)         0       1  ];
    Bsys = B1;
    % Compute u(k),x(k) and norm of F(k)
    [Fn,gamman] = rmpcpoly_mass_Lya(A1,B1,A2,B2,Qw,Rw,xn,umax);
    un = Fn∗xn;
    normFn = norm(Fn);
    if (i>2)&&(normFn − dataoutn(i−1,5)) < 10^(−2)
            normFdis = dataoutn(i−1,5);
    else
            normFdis = normFn;
    end
    dataoutn(i,:)=[i xn(2) un gamman normFdis xn(4)];
    xn = (Asys+Bsys∗Fn)∗xn;
end

% Static state feedback
xs = [1; 1; 0; 0];  % initial state
[Fs,gammas] = rmpcpoly_mass(A1,B1,A2,B2,Qw,Rw,xs,umax);  for i =1:N
    Asys  = [    1              0          0.1     0
```

```
                0              1            0      0.1
           -0.1*Ka(i)   0.1*Ka(i)          1       0
            0.1*Ka(i)  -0.1*Ka(i)          0       1 ];
    Bsys = B1;
    % Compute u(k),x(k) and norm of F(k)
    us = Fs*xs;
    normFs = norm(Fs);
    dataouts(i,:)=[i xs(2) us gammas normFs xs(4)];
    xs = (Asys+Bsys*Fs)*xs;
end

time = dataout(:,1)/10;
figure (1);        % plot the state x2(k)
plot(time,dataout(:,2),'--'); hold on;
plot(time,dataoutn(:,2));
hold on;
plot(time,dataouts(:,2),':');
xlabel('time (sec)');
ylabel('state x_2 ');
figure (2);        % plot the state x4(k)
plot(time,dataout(:,6),'--'); hold on;
plot(time,dataoutn(:,6));
hold on;
plot(time,dataouts(:,6),':');
xlabel('time (sec)');
ylabel('state x_4 ');
figure (3);         % plot the input u
plot(time,dataout(:,3),'--'); hold on;
plot(time,dataoutn(:,3));
hold on;
plot(time,dataouts(:,3),':');
xlabel('time (sec)');
ylabel('control u');
figure (4);         % plot the index gamma
plot(time,dataout(:,4),'--'); hold on;
plot(time,dataoutn(:,4));
hold on;
plot(time,dataouts(:,4),':');
xlabel('time (sec)');
ylabel('\gamma');
figure (5);         % plot the norm of matrix F
plot(time,dataout(:,5),'--'); hold on;
plot(time,dataoutn(:,5));
hold on;
plot(time,dataouts(:,5),':');
xlabel('time (sec)');
ylabel('norm of F');
```

rmpcpoly_mass.m

```
%************************************************************************************
% rmpcpoly_mass.m
% Used in file: mass_spring.m
% To determine the feedback matrix F of the state feedback RCMPC law
% with a single Lyapunov function
% by solving an LMI optimization
%
% Tu Anh Do
```

```
% Update: 25/04/2006
%***************************************************************************************
function [F,gamma] = rmpcpoly_mass(A1,B1,A2,B2,Qw,Rw,xk,um)

n = 4;          % number of state variables
p = 1;          % number of inputs
% Define the variables for the minimization problem
gamma = sdpvar(1,1);
Q = sdpvar(n,n);
Y = sdpvar(p,n);
% Declare the left-hand side of the LMIs
Qws = sqrtm(Qw);
Rws = sqrtm(Rw);
I1 = eye(n);
I2 = eye(p);

F1 = [1 xk';xk Q];
F21 = [    Q          Q*A1'+Y'*B1'     Q*Qws  Y'*Rws
       A1*Q+B1*Y          Q            zeros(n)    zeros(n,p)
         Qws*Q         zeros(n)       gamma*I1    zeros(n,p)
         Rws*Y         zeros(p,n)     zeros(p,n)    gamma*I2  ];
F22 = [    Q          Q*A2'+Y'*B2'     Q*Qws        Y'*Rws
       A2*Q+B2*Y          Q            zeros(n)    zeros(n,p)
         Qws*Q         zeros(n)       gamma*I1    zeros(n,p)
         Rws*Y         zeros(p,n)     zeros(p,n)    gamma*I2  ];
F3 = [um^2*I2   Y              % for input constraints
        Y'       Q];
% Set up the constraints for the minimization problem
constraint = set(Q>0)+set(F1>=0)+set(F21>0)+set(F22>0)+set(F3>=0);
% Solve the minimization problem
solvesdp(constraint,gamma);
% The minimizer
Y = double(Y); Q = double(Q); gamma = double(gamma);

% The state feedback matrix
F = Y*inv(Q);
```

rmpcpoly_mass_Lya.m

```
%***************************************************************************************
% rmpcpoly_mass_Lya.m
% Used in file: mass_spring.m
% To determine the feedback matrix F of the state feedback RCMPC law
% with a parameter-dependent Lyapunov function
% by solving an LMI optimization
%
% Tu Anh Do
% Update: 25/04/2006
%***************************************************************************************
function [F,gamma] = rmpcpoly_mass_Lya(A1,B1,A2,B2,Qw,Rw,xk,um)

n = 4;          % number of state variables
p = 1;          % number of inputs
% Define the variables for the minimization problem
gamma = sdpvar(1,1);
Q1 = sdpvar(n,n);
Q2 = sdpvar(n,n);
```

```
G = sdpvar(n,n);
Y = sdpvar(p,n);
% Declare the left-hand side of the LMIs
Qws = sqrtm(Qw);
Rws = sqrtm(Rw);
I1 = eye(n);
I2 = eye(p);

F11 = [1 xk';xk Q1];
F12 = [1 xk';xk Q2];

F21 = [ G+G'-Q1     G'*A1'+Y'*B1'   G'*Qws        Y'*Rws
        A1*G+B1*Y        Q1           zeros(n)     zeros(n,p)
        Qws*G          zeros(n)       gamma*I1     zeros(n,p)
        Rws*Y          zeros(p,n)     zeros(p,n)   gamma*I2  ];
F22 = [ G+G'-Q1     G'*A1'+Y'*B1'   G'*Qws        Y'*Rws
        A1*G+B1*Y        Q2           zeros(n)     zeros(n,p)
        Qws*G          zeros(n)       gamma*I1     zeros(n,p)
        Rws*Y          zeros(p,n)     zeros(p,n)   gamma*I2  ];
F23 = [ G+G'-Q2     G'*A2'+Y'*B2'   G'*Qws        Y'*Rws
        A2*G+B2*Y        Q1           zeros(n)     zeros(n,p)
        Qws*G          zeros(n)       gamma*I1     zeros(n,p)
        Rws*Y          zeros(p,n)     zeros(p,n)   gamma*I2  ];
F24 = [ G+G'-Q2     G'*A2'+Y'*B2'   G'*Qws        Y'*Rws
        A2*G+B2*Y        Q2           zeros(n)     zeros(n,p)
        Qws*G          zeros(n)       gamma*I1     zeros(n,p)
        Rws*Y          zeros(p,n)     zeros(p,n)   gamma*I2  ];
F31 = [um^2*I2     Y              % input constraints
        Y'      G+G'-Q1];
F32 = [um^2*I2     Y
        Y'      G+G'-Q2];
% Set up the constraints for the minimization problem
constraint = set(F11>=0)+set(F12>=0)+set(F21>0)+set(F22>0)
             +set(F23>0)+set(F24>0)+set(F31>=0)+set(F32>=0);
% Solve the minimization problem
solvesdp(constraint,gamma);
% The minimizer
Y = double(Y); G = double(G); gamma = double(gamma);

% The state feedback matrix
F = Y*inv(G);
```

D Program for Simulation of a Non-Isothermal CSTR

reactor.m

```
%*************************************************************************************
% reactor.m
% To simulate the response of a non-isothermal CSTR
% Using output feedback RCMPC
% by combining an off-line state feedback RCMPC together with an estimator
% Compare 2 cases: a single Lyapunov function and PDLF
% Tu Anh Do
%
% Files needed: rmpcpoly_reactor.m, rmpcpoly_Lya_reactor, observer_reactor.m
% Update: 06/06/2006
%*************************************************************************************
close all;
```

```
clear all;
tic        % start clock

T = 0.15;                  % sampling time
al_min = 1; al_max = 5;    % uncertainty set
be_min = 1; be_max = 5;
N = 21;                    % number of iterations
for i=1:N alpha = 0;
    while alpha < al_min
        alpha = al_max*rand; % alpha belongs to [al_min al_max]
    end
    alvec(i) = alpha;
end for i=1:N beta = 0;
    while beta < be_min
        beta = be_max*rand;  % beta belongs to [be_min be_max]
    end
    bevec(i) = beta;
end
n = 2;                     % number of state variables
p = 1;                     % number of inputs
% Declare the vertices of the polytopic model
A1  = [ 0.85-0.0986*al_min             -0.0014*al_min
       0.9864*al_min*be_min    0.0487+0.01403*al_min*be_min];
B1 = [    0
       -0.912];
A2  = [ 0.85-0.0986*al_min             -0.0014*al_min
       0.9864*al_min*be_max    0.0487+0.01403*al_min*be_max];
B2 = B1; A3  = [ 0.85-0.0986*al_max           -0.0014*al_max
       0.9864*al_max*be_min    0.0487+0.01403*al_max*be_min];
B3 = B1; A4  = [ 0.85-0.0986*al_max           -0.0014*al_max
       0.9864*al_max*be_max    0.0487+0.01403*al_max*be_max];
B4 = B1;
% The nominal plant
A0 = (A1+A2+A3+A4)/4; B0 = B1;
% The time-varying system
Csys = [0   1];
Bsys = B1;
% Input constraints
umax = 0.5;
% Design parameters
Qw = [1 0; 0 1];
Rw = 0.00002;
rho = 0.1;
% Sequence of Ks state vectors
Ks = 10; x_1 = [1 0.5 0.3 0.2 0.15 0.1 0.07 0.05 0.035 0.01]; x_2 =
zeros(1,Ks); v = zeros(Ks,2); for k = 1:Ks
    v(k,:) = [x_1(k)  x_2(k)];
end
% Set up a lookup table of (Qi,Fi)
dataout1 = zeros(n,n*Ks);    % to store matrices Qi
dataout2 = zeros(p,n*Ks);    % to store matrices Fi
dataout3 = zeros(1,Ks);      % to store gamma
for i =1:Ks
    [F,Q,gamma] = rmpcpoly_reactor(A1,B1,A2,B2,A3,B3,A4,B4,Qw,Rw,v(i,:)',umax);
    dataout1(:,(n*i-(n-1)):n*i) = Q;
    dataout2(:,(n*i-(n-1)):n*i) = F;
    dataout3(1,i) = gamma;
end
```

```
% Obtain the estimator gain
Lp = observer_reactor(A0,Csys,rho);

% With a single Lyapunov function
xk = [0.1; 2]; x_estk = [0 ; 0]; x_augk = [xk;x_estk]; for i = 1:N
    if x_estk == [0;0]
        index = Ks;
    else
        for j = 1:Ks
            if ( x_estk'*inv(dataout1(:,(n*j-(n-1)):n*j))*x_estk <=1)
                index = j;
            else
                break;
            end
        end
    end
    if index < Ks
        Qn = dataout1(:,(n*index-(n-1)):n*index);
        Fn = dataout2(:,(n*index-(n-1)):n*index);
        Qnn = dataout1(:,(n*(index+1)-(n-1)):n*(index+1));
        Fnn = dataout2(:,(n*(index+1)-(n-1)):n*(index+1));
        alphak = (1-(x_estk'*inv(Qnn)*x_estk))/(x_estk'*(inv(Qn) - inv(Qnn))*x_estk);
        Fk = alphak*Fn+(1-alphak)*Fnn ;dataalpha1(i)=alphak;
    else
        Fk = dataout2(:,(n*Ks-(n-1)):(n*Ks));
    end

    datanorm1(i)= norm(Fk);
    uk = Fk*x_augk(n+1:2*n);
    yk = Csys*x_augk(1:n);
    dataout4(i,:) = [ i uk yk index x_augk(1) x_augk(3)];
    dataout5(i,:) = [(x_augk(1)-x_augk(3)) (x_augk(2)-x_augk(4))];
    Asys = [ 0.85-0.0986*alvec(i)            -0.0014*alvec(i)
           0.9864*alvec(i)*bevec(i)   0.0487+0.01403*alvec(i)*bevec(i)];
    Apoly = [ Asys          Bsys*Fk
            Lp*Csys   A0+B0*Fk-Lp*Csys];
    x_augk = Apoly*x_augk ;
    x_estk = x_augk(n+1:2*n);
end

% With a PDLF
% Set up a lookup table of (Qi,Fi)
dataout1new = zeros(n,n*Ks);   % to store matrices Qi
dataout2new = zeros(p,n*Ks);   % to store matrices Fi
dataout3new = zeros(1,Ks);     % to store gamma
for i =1:Ks
    [Fnew,Qm,gammanew] = rmpcpoly_Lya_reactor(A1,B1,A2,B2,A3,B3,A4,B4,Qw,Rw,v(i,:)',umax);
    dataout1new(:,(n*i-(n-1)):n*i) = Qm;
    dataout2new(:,(n*i-(n-1)):n*i) = Fnew;
    dataout3new(1,i) = gammanew;
end
xk = [0.1; 2];
x_estk = [0 ; 0];
x_augk = [xk;x_estk];

for i = 1:N
    if x_estk == [0;0]
        index = Ks;
```

```
    else
        for j = 1:Ks
            if ( x_estk'*inv(dataout1new(:,(n*j-(n-1)):n*j))*x_estk <=1)
                index = j;
            else
                break;
            end
        end
    end
    if index < Ks
        Qn = dataout1new(:,(n*index-(n-1)):n*index);
        Fn = dataout2new(:,(n*index-(n-1)):n*index);
        Qnn = dataout1new(:,(n*(index+1)-(n-1)):n*(index+1));
        Fnn = dataout2new(:,(n*(index+1)-(n-1)):n*(index+1));
        alphak = (1-(x_estk'*inv(Qnn)*x_estk))/(x_estk'*(inv(Qn) - inv(Qnn))*x_estk);
        Fk = alphak*Fn+(1-alphak)*Fnn ; dataalpha2(i)=alphak;
    else
        Fk = dataout2new(:,(n*Ks-(n-1)):(n*Ks));
    end

    datanorm2(i)= norm(Fk);
    uk = Fk*x_augk(n+1:2*n);
    yk = Csys*x_augk(1:n);
    dataout4new(i,:) = [i uk yk index x_augk(1) x_augk(3)];
    dataout5new(i,:) = [(x_augk(1)-x_augk(3)) (x_augk(2)-x_augk(4))];
    Asys = [ 0.85-0.0986*alvec(i)            -0.0014*alvec(i)
           0.9864*alvec(i)*bevec(i)  0.0487+0.01403*alvec(i)*bevec(i)];
    Apoly = [ Asys          Bsys*Fk
             Lp*Csys   A0+B0*Fk-Lp*Csys];
    x_augk = Apoly*x_augk ;
    x_estk = x_augk(n+1:2*n);
end
toc     % stop clock

% Plots
time = dataout4(:,1)*T-T;
figure (1);   % plot the output y
plot(time,dataout4new(:,3));
hold on;
plot(time,dataout4(:,3),':');
xlabel('time, t(min)');
ylabel('Output, y ');
figure (2);   % plot the input u
plot(time,dataout4new(:,2)); hold on;
plot(time,dataout4(:,2),':');
xlabel('time, t(min)');
ylabel('Control move, u');
figure(3)
plot(time,datanorm2,'r'); hold on;
plot(time,datanorm1);
xlabel('time, t(min)');
ylabel('norm of F');
figure(4)
plot(1:Ks,dataout3new); hold on;
plot(1:Ks,dataout3,':');
xlabel('number');
ylabel('\gamma');
figure(5)
```

```
plot(time,dataout5new(:,1)); hold on;
plot(time,dataout5(:,1),':'); hold on;
xlabel('time, t(min)');
ylabel('Estimated error of state x 1');
figure(6)
plot(time,dataout5new(:,2),'r'); hold on;
plot(time,dataout5(:,2),'r:');
xlabel('time, t(min)');
ylabel('Estimated error of state x 2');
```

rmpcpoly_reactor.m

```
%*************************************************************************************
% rmpcpoly_reactor.m
% Used in file: reactor.m
% To determine the  matrices Q and F of the output feedback RCMPC law
% by solving an LMI optimization
% With a single Lyapunov function
%
% Tu Anh Do
% Update: 25/04/2006
%*************************************************************************************
function [F,Q,gamma] =
rmpcpoly_reactor(A1,B1,A2,B2,A3,B3,A4,B4,Qw,Rw,xk,um)

n = 2;          % number of state variables
p = 1;          % number of inputs
% Define the variables for the minimization problem
gamma = sdpvar(1,1);
Q = sdpvar(n,n);
Y = sdpvar(p,n);
% Declare the left-hand side of the LMIs
Qws = sqrtm(Qw);
Rws = sqrtm(Rw);
I1 = eye(n);
I2 = eye(p);

F1 = [1 xk';xk Q];
F21 = [   Q        Q*A1'+Y'*B1'    Q*Qws'      Y'*Rws'
      A1*Q+B1*Y        Q          zeros(n)    zeros(n,p)
       Qws*Q        zeros(n)      gamma*I1    zeros(n,p)
       Rws*Y       zeros(p,n)    zeros(p,n)   gamma*I2  ];
F22 = [   Q        Q*A2'+Y'*B2'    Q*Qws'      Y'*Rws'
      A2*Q+B2*Y        Q          zeros(n)    zeros(n,p)
       Qws*Q        zeros(n)      gamma*I1    zeros(n,p)
       Rws*Y       zeros(p,n)    zeros(p,n)   gamma*I2  ];
F23 = [   Q        Q*A3'+Y'*B3'    Q*Qws'      Y'*Rws'
      A3*Q+B3*Y        Q          zeros(n)    zeros(n,p)
       Qws*Q        zeros(n)      gamma*I1    zeros(n,p)
       Rws*Y       zeros(p,n)    zeros(p,n)   gamma*I2  ];
F24 = [   Q        Q*A4'+Y'*B4'    Q*Qws'      Y'*Rws'
      A4*Q+B4*Y        Q          zeros(n)    zeros(n,p)
       Qws*Q        zeros(n)      gamma*I1    zeros(n,p)
       Rws*Y       zeros(p,n)    zeros(p,n)   gamma*I2  ];
F3 = [um^2*I2   Y            % for input constraints
        Y'      Q];
% Set up the constraints for the minimization problem
constraint = set(Q>0)+set(F1>=0)+set(F21>0)+set(F22>0)
```

```
            +set(F23>0)+set(F24>0)+set(F3>=0);
% Solve the minimization problem
solvesdp(constraint,gamma);
% The minimizer
Y = double(Y);
Q = double(Q);
gamma = double(gamma);

% The state feedback matrix
F = Y*inv(Q);
```

rmpcpoly_Lya_reactor.m

```
%*************************************************************************************
% rmpcpoly_Lya_reactor.m
% Used in file: reactor.m
% To determine the  matrices Q and F of the output feedback RCMPC law
% by solving an LMI optimization
% With a PDLF
%
% Tu Anh Do
% Update: 25/04/2006
%*************************************************************************************
function [F,Qm,gamma] =
rmpcpoly_Lya_reactor(A1,B1,A2,B2,A3,B3,A4,B4,Qw,Rw,xk,um)

n = 2; % number of state variables
p = 1;          % number of inputs
% Define the variables for the minimization problem
gamma = sdpvar(1,1);
Q1 = sdpvar(n,n);
Q2 = sdpvar(n,n);
Q3 = sdpvar(n,n);
Q4 = sdpvar(n,n);
G = sdpvar(n,n);
Y = sdpvar(p,n);
% Declare the left-hand side of the LMIs
Qws = sqrtm(Qw);
Rws = sqrtm(Rw);
I1 = eye(n);
I2 = eye(p);

F11 = [1 xk';xk Q1];
F12 = [1 xk';xk Q2];
F13 = [1 xk';xk Q3];
F14 = [1 xk';xk Q4];

F21 = [ G+G'-Q1     G'*A1'+Y'*B1'    G'*Qws'      Y'*Rws'
       A1*G+B1*Y        Q1           zeros(n)     zeros(n,p)
        Qws*G         zeros(n)       gamma*I1     zeros(n,p)
        Rws*Y         zeros(p,n)     zeros(p,n)   gamma*I2  ];
F22 = [ G+G'-Q1     G'*A1'+Y'*B1'    G'*Qws'      Y'*Rws'
       A1*G+B1*Y        Q2           zeros(n)     zeros(n,p)
        Qws*G         zeros(n)       gamma*I1     zeros(n,p)
        Rws*Y         zeros(p,n)     zeros(p,n)   gamma*I2  ];
F23 = [ G+G'-Q1     G'*A1'+Y'*B1'    G'*Qws'      Y'*Rws'
       A1*G+B1*Y        Q3           zeros(n)     zeros(n,p)
        Qws*G         zeros(n)       gamma*I1     zeros(n,p)
```

```
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F24 = [ G+G'-Q1    G'*A1'+Y'*B1'   G'*Qws'        Y'*Rws'
     A1*G+B1*Y      Q4             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F25 = [ G+G'-Q2    G'*A2'+Y'*B2'   G'*Qws'        Y'*Rws'
     A2*G+B2*Y      Q1             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F26 = [ G+G'-Q2    G'*A2'+Y'*B2'   G'*Qws'        Y'*Rws'
     A2*G+B2*Y      Q2             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F27 = [ G+G'-Q2    G'*A2'+Y'*B2'   G'*Qws'        Y'*Rws'
     A2*G+B2*Y      Q3             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F28 = [ G+G'-Q2    G'*A2'+Y'*B2'   G'*Qws'        Y'*Rws'
     A2*G+B2*Y      Q4             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F29 = [ G+G'-Q3    G'*A3'+Y'*B3'   G'*Qws'        Y'*Rws'
     A3*G+B3*Y      Q1             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F210 = [ G+G'-Q3   G'*A3'+Y'*B3'   G'*Qws'        Y'*Rws'
     A3*G+B3*Y      Q2             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F211 = [ G+G'-Q3   G'*A3'+Y'*B3'   G'*Qws'        Y'*Rws'
     A3*G+B3*Y      Q3             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F212 = [ G+G'-Q3   G'*A3'+Y'*B3'   G'*Qws'        Y'*Rws'
     A3*G+B3*Y      Q4             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F213 = [ G+G'-Q4   G'*A4'+Y'*B4'   G'*Qws'        Y'*Rws'
     A4*G+B4*Y      Q1             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F214 = [ G+G'-Q4   G'*A4'+Y'*B4'   G'*Qws'        Y'*Rws'
     A4*G+B4*Y      Q2             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F215 = [ G+G'-Q4   G'*A4'+Y'*B4'   G'*Qws'        Y'*Rws'
     A4*G+B4*Y      Q3             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];
F216 = [ G+G'-Q4   G'*A4'+Y'*B4'   G'*Qws'        Y'*Rws'
     A4*G+B4*Y      Q4             zeros(n)       zeros(n,p)
     Qws*G          zeros(n)       gamma*I1       zeros(n,p)
     Rws*Y          zeros(p,n)     zeros(p,n)     gamma*I2  ];

F31 = [um^2*I2     Y              % input constraints
     Y'       G+G'-Q1];
F32 = [um^2*I2     Y
     Y'       G+G'-Q2];
```

```
F33 = [um^2*I2        Y
          Y'        G+G'-Q3];
F34 = [um^2*I2        Y
          Y'        G+G'-Q4];

% Set up the constraints for the minimization problem
constraint = set(F11>=0)+set(F12>=0)+set(F13>=0)+set(F14>=0)
             +set(F21>0)+set(F22>0)+set(F23>0)+set(F24>0)+set(F25>0)
             +set(F26>0)+set(F27>0)+set(F28>0)+set(F29>0)
             +set(F210>0)+set(F211>0)+set(F212>0)+set(F213>0)+set(F214>0)+set(F215>0)
             +set(F216>0)+set(F31>=0)+set(F32>=0)+set(F33>=0)+set(F34>=0);
% Solve the minimization problem
solvesdp(constraint,gamma);
% The minimizer
Y = double(Y);
Q1 = double(Q1);
Q2 = double(Q2);
Q3 = double(Q3);
Q4 = double(Q4);
G = double(G);
gamma = double(gamma);

% Find the maximal Q
t = sdpvar(1,1);
Qm = sdpvar(n,n);
M0 = Qm - t*eye(n);
M1 = Q1 - Qm;
M2 = Q2 - Qm;
M3 = Q3 - Qm;
M4 = Q4 - Qm;

constraint_Q= set(M0>0)+set(M1>=0)+set(M2>=0)+set(M3>=0)+set(M4>=0);
solvesdp(constraint_Q,-t);
Qm = double(Qm);
t = double(t);

% The state feedback matrix
F = Y*inv(G);
```

observer_reactor.m

```
%***************************************************************************************
% observer_reactor.m
% Used in file: reactor.m
% To determine the gain matrix Lp of the estimator
% by solving an LMI feasibility problem
%
% Tu Anh Do
% Update: 25/04/2006
%***************************************************************************************
function Lp = observer_reactor(A,C,rho)

n = 2;          % number of state variables
p = 1;          % number of inputs
% Define the variables for the minimization problem
Q = sdpvar(n,n); Y = sdpvar(n,p);
% Declare the left-hand side of the LMIs
F  = [ rho^2*Q    Q*A-Y*C
```

```
        A'*Q-C'*Y'      Q     ];
constraint = set(Q>0)+set(F >=0);
% Solve the minimization problem
solvesdp(constraint);
% The minimizer
Y = double(Y); Q = double(Q);

% The estimator gain
Lp = inv(Q)*Y;
```

Biography

Tu Anh Thi Do was born in Hanoi, Vietnam in 1980. She received her bachelor's degree in electrical engineering from Hanoi University of Technology, Vietnam in 2003. She has been granted a scholarship by the AUN/SEED-Net (www.seed-net.org) to pursue her master degree in electrical engineering at Chulalongkorn University, Thailand, since 2004. Her graduate study is completed under Control Systems Research Laboratory, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University. Her research work is in the area of model predictive control and robust control.