ROBUST MODEL PREDICTIVE CONTROL FOR PROCESS CONTROL AND SUPPLY CHAIN OPTIMIZATION

ROBUST MODEL PREDICTIVE CONTROL FOR PROCESS CONTROL AND SUPPLY CHAIN OPTIMIZATION

By

XIANG LI, B.Eng., M.Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2009) McMaster University

(Chemical Engineering) Hamilton, Ontario

TITLE: Robust Model Predictive Control for Process Control and Supply Chain Optimization

AUTHOR: Xiang Li, B.Eng., M.Eng. (Zhejiang University, P. R. China)

SUPERVISOR: Professor T. E. Marlin

NUMBER OF PAGES: x, 295

Abstract

Model Predictive Control (MPC) is traditionally designed assuming no model mismatch and tuned to provide acceptable behavior when mismatch occurs. This thesis extends the MPC design to account for explicit mismatch in the control and optimization of a wide range of uncertain dynamic systems with feedback, such as in process control and supply chain optimization.

The major contribution of the thesis is the development of a new MPC method for robust performance, which offers a general framework to optimize the uncertain system behavior in the closed-loop subject to hard bounds on manipulated variables and soft bounds on controlled variables. This framework includes the explicit handling of correlated, time-varying or time-invariant, parametric uncertainty appearing externally (in demands and disturbances) and internally (in plant/model mismatch) to the control system. In addition, the uncertainty in state estimation is accounted for in the controller.

For efficient and reliable real-time solution, the bilevel stochastic optimization formulation of the robust MPC method is approximated by a limited number of (convex) Second Order Cone Programming (SOCP) problems with an industry-proven heuristic and the classical chance-constrained programming technique. A closed-loop uncertainty characterization method is also developed which improves real-time tractability by performing intensive calculations off-line.

The new robust MPC method is extended for process control problems by integrating a robust steady-state optimization method addressing closed-loop uncertainty. In addition, the objective function for trajectory optimization can be formulated as nominal or expected dynamic performance. Finally, the method is formulated in deviation variables to correctly estimate time-invariant uncertainty.

The new robust MPC method is also tailored for supply chain optimization, which is demonstrated through a typical industrial supply chain optimization problem. The robust MPC optimizes scenario-specific safety stock levels while satisfying customer demands for time-varying systems with uncertainty in demand, manufacturing and transportation. Complexity analysis and computational study results demonstrate that the robust MPC solution times increase with system scale moderately, and the method does not suffer from the curse of dimensionality.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Thomas Marlin, for his patient and insightful guidance, which not only has helped out this thesis, but also will be beneficial to my future career. I also thank him for giving me ample freedom to carry out my own idea, especially at the early stage of the research.

Besides my supervisor, I would like to thank other members of my research committee, Dr. Chris Swartz, Dr. Tamás Terlaky and Dr. Tim Davidson, for their valuable suggestions as well as hard questions that boost the progress of this research.

My sincere thanks also go to all the fellow graduate students in the MACC, for their friendship that make my life during the Ph.D. study warm and colorful. In particular, I would like to thank Yongsong, Zhiwen, Adam, Alberto, Zheng, Honglu, Jesus, Kevin, Danielle, Alexei, Rhoda, Rahul, Benoit. I would especially thank Weitao for helping me settle down and get familiar with Canadian life quickly during my early time in Hamilton.

In addition, I am grateful to a confidential industrial collaborator, for their offering important industrial data to facilitate the supply chain optimization research in this thesis.

Last but not the least, I would like to thank my parents, Bingrui and Zhijiang, for bringing me to this world, raising me up and always giving me unconditional support, and my wife, Hui, for her understanding and encouragement during the course of my Ph.D. research.

Table of Contents

Chapter 1

Introduction

This thesis focuses on the optimization of uncertain dynamic systems with feedback. Here "feedback" is the term from automatic control, which means the use of the measurement information of system outputs to determine future system inputs. While all physical systems are uncertain to some extent, this thesis concentrates on physical systems in which uncertainty significantly affects the behaviour of the control.

There exist a wide range of feedback control technologies. Here, we will concentrate on a method termed "Model Predictive Control" (MPC) for reasons explained subsequently. In addition, there are many methods for determining optimal performance for systems (usually, without feedback control), such as stochastic optimization. The developments in this thesis merge MPC and stochastic optimization in novel ways to provide unique advantages for the overall system performance.

The methods developed in this thesis are general, but they have been tailored to two applications of special interest in the chemical process industries. The first is process control, which applies automatic control to equipment in the process industries. The second in supply chain optimization (management), which involves coordinating the raw materials, manufacturing, transportation, sales, and storage of an integrated business. Both of these systems involve slow dynamics (relative to disturbances and control objectives) and significant uncertainties that provide challenges in attaining the desired performance.

The remainder of this chapter contains the following sections. Section 1.1 gives a general introduction of MPC methods (including robust MPC), and Sections 1.2 and 1.3 introduce process control and supply chain optimization, respectively. Then, Section 1.4 further defines the goal and scope of the research, and Section 1.5 shows an overview outline of the thesis. Finally, Section 1.6 defines some terms and conventions used in the thesis.

1.1 Model Predictive Control

1.1.1 Conventional, nominal MPC

Model predictive control (MPC) refers to a class of computer control algorithms that utilize an explicit process model to predict the future response of a plant (Qin and Badgewell, 2003). At each control interval, feedback information is used to update the model to reduce the effects of model mismatch. Then, the MPC algorithm optimizes future model behavior by computing a sequence of future manipulated variables. The first input in the optimal sequence is then sent to the plant, and the entire calculation is repeated at subsequent control intervals. The advantage of this "rolling horizon" implementation is that the new measurement information of the system at each control interval can be used to update the explicit process model for a better prediction of the future.

Figure 1.1 shows a typical MPC block diagram, where y denotes the controlled variables (or sometimes called system outputs in the thesis), which are to be maintained at their desired values, u denotes the manipulated variables (or sometimes called system inputs in the thesis), which are adjusted to drive the controlled variables to their desired values. The desired values of the controlled variables are called set points, shown as SP in the figure. An MPC controller includes at least an explicit model to predict the future and an optimizer to compute the optimal manipulated variables according to the prediction.

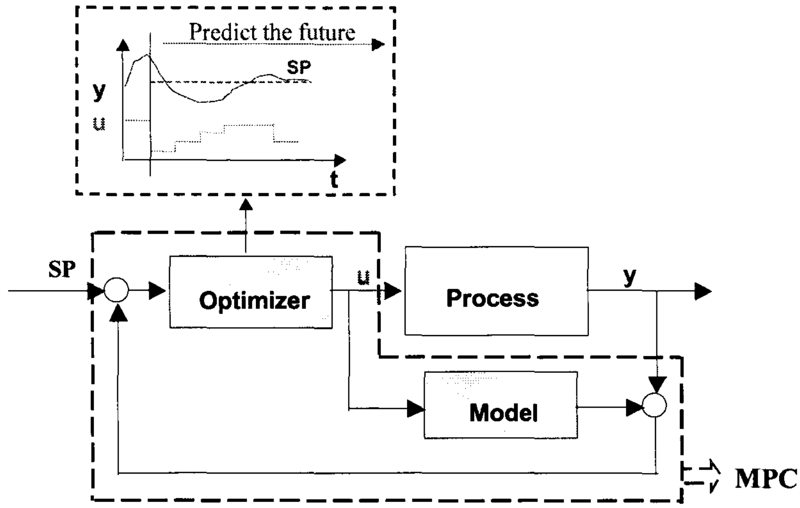


Figure 1.1 A typical block diagram of MPC

MPC has been widely adopted in industry as an effective control technology to deal with multivariable constrained control problems (Qin and Badgwell, 2003). Conventionally, MPC uses a linear deterministic model to predict the future, although a real system is usually nonlinear and uncertain. We call this type of MPC method "nominal MPC" in this thesis and call the deterministic model "nominal model".

1.1.2 Robust MPC, nonlinear MPC and adaptive MPC

The performance of nominal MPC will degrade if the real plant deviates from the nominal model used in the controller. The difference between the real plant and its nominal model is usually called the uncertainty, which is caused by the differences in structure, measurement error, and unmeasured disturbances in the system. Applying an MPC using only the nominal model and (de)tuning the controller is a typical approach to address the uncertainty, but it can lead to poor performance. Substantial performance improvement could be achieved by addressing the uncertainty explicitly in the control calculation, which leads to the concept of robust MPC. Unless otherwise specified, robust MPC refers to the MPC methods using linear nominal model with parametric uncertainties, which calculates the manipulated variables such that the future plant behavior satisfies specific (feasibility, stability or performance) criteria for not only the nominal plant realization, but also all the other possible plant realizations with the uncertainty in a specified region.

When the nonlinearity of the plant is significant and a good model of the non-linear process is available, describing the plant with a nominal model and the additional uncertainty may not be the appropriate controller to achieve a good performance. In this case, an explicit use of a nonlinear model would be appropriate to predict the future behavior. The resulting MPC method is called nonlinear MPC (Camacho and Bordons, 2007; Badgwell and Qin, 2001; Allgower et. al. 1999). Furthermore, if the uncertainty is explicitly addressed in nonlinear MPC calculation, the resulting method is called robust nonlinear MPC (e.g., Grancharova and Johansen, 2009; Zavala and Biegler, 2009). Although it has attracted much attention in the academic research, the nonlinear MPC has not been very widely applied in the industry, because some difficult problems, including nonlinear identification, nonlinear state estimation, nonconvex optimization of the transient behavior, etc. (Camacho and Bordons, 2007).

An alternative approach to address uncertainty is to adopt the idea from adaptive control, where the control law or controller tuning is changed with the real-time measurement information of the system (Sastry and Bodson, 1994). In the context of MPC, it usually refers to the MPC methods with real-time identification or selection of prediction models (e.g., Dougherty and Cooper, 2003). The limitation of adaptive MPC control is that it's challenging to satisfy the stability or even feasibility and other performance criteria, especially when the uncertainty changes frequently (Mayne, et al., 2000). So, an adaptive MPC method is usually designed with the integration of robust MPC techniques to address some uncertainty explicitly in the controller calculation, which is actually a robust adaptive MPC (e.g. Fukushima et al, 2007). Sometimes a nonlinear model is also used in the method, which gives robust adaptive nonlinear MPC (e.g. Rahideh et al., 2008; Adetola et al., 2009).

In this thesis, we only address the robust MPC method using linear dynamic models with uncertain parameters. This choice is appropriate when (a) the system is not highly non-linear or non-linear models appropriate for real-time control are not available and (b) adapting models in real time is not appropriate because of lack of correlation of the plant behaviour with its recent past. Many process control and supply chain problems fit into this category, as indicated by the current dominant practice of applying linear nominal MPC with detuning.

1.2 Process Control

Process control is a sub-discipline of automatic control that involves tailoring methods for the operation of chemical processes with the goal of improving the safety and profitability of a process, while maintaining consistently high product quality (Marlin, 2000).

Process control usually deals with physicochemical systems such as reactors, heat exchangers, distillation columns, and so forth. The controlled variables and manipulated variables in process control depend on the specific process and control objectives. They can typically be temperature, pressure, flow rate, level, and composition. For example, for a binary distillation column that separates the light product from the feed flow, the controlled variables are usually the pressure, liquid levels, and compositions of the distillate and the bottoms products, and the manipulated variables are the condenser duty, reflux flow, reboiler duty, and product flow rates.

In process control, the cause of the uncertainty can be the uncertainty in the system variables that are independent from the control decisions. For example, in the binary distillation control system mentioned above, the feed flow rate and the feed composition can vary, which changes the relationship between the controlled and manipulated variables. The cause of the uncertainty can also be the approximation of a nonlinear process with linear model. This is because a linear model is only exactly accurate for a nonlinear process at a particular point, but the process operates in a region around a nominal operating point. The uncertainty of a process control system usually occurs as the net effects of the two causes mentioned above.

The most widely used process control methods are Proportional-Integral-Derivative (PID) control methods (e.g., Marlin, 2000) and MPC. The PID methods are appropriate for controlling a single output variable or for multivariable systems in which the interaction among control loops, which do not communicate in the controller calculations, does not significantly degrade dynamic performance. In contrast, the more complex MPC method is appropriate for systems with strong interaction among variables, for which centralized decision making will improve dynamic performance. This work deals exclusively with MPC control that optimizes the trajectory of the controlled and manipulated variables to their desired values.

The desired values of controlled variables are usually obtained according to an economic objective, which could change while disturbances enter the process. So, a steady-state optimization unit is usually integrated in many industrial process control systems that calculate the desired values of the controlled variables (and mostly also the desired values of the manipulated variables) before each controller execution (Qin and Badgwell, 2003). Due to its importance in process control practice, the steady-state optimization problem will also be addressed in this thesis.

1.3 Supply Chain Optimization

Supply chain optimization or supply chain management is "a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses and stores, so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, in order to minimize system-wide costs while satisfying service level requirements" (Simchi-Levi et al., 1999). Extensive work has been done for supply chain optimization in the management, operations research and industrial engineering communities, with a tremendous number of publications available. Readers can refer to Pinedo (2005), Chopra and Meindl (2004) and Nahmias (2001) for the review of the traditional supply chain optimization research.

The supply chain optimization applied to the process industries has been a hot research area recently (Shah, 2006; Neiro and Pinto, 2005), which includes making decisions for different levels of a supply chain, namely, strategic, tactical and operational (Neiro and Pinto, 2005). The operational problem, which refers to making decisions to drive some variables of the supply chain to the desired values (or force them to satisfy some objective), is analog to the process control problem. The decision variables (inputs) of the problem could be the production rate, orders to the upper stream units (Braun et al., 2003), or the advertising effort to increase the sales (Tzafestas et al., 1997). The controlled variables (outputs) could be the inventories, productions rates or material feedstock purchases, and they can be measured. Figure 1.2 illustrates the similar structure of process control and supply chain operation problems in the MPC framework. This motivates the concept of introducing feedback control technology to optimal supply chain operation problems, especially the MPC techniques (e.g. Braun et al., 2003; Perea-Lopez et al. 2001; Tzafestas et al., 1997).

The customer demands, which can change frequently and substantially and cannot be predicted accurately, are the most common disturbances in a supply chain system. The processing time for products, transportation times and the prices of the materials are other main sources of uncertainties (Lin et al., 2004). The most important impact of the uncertainties on a supply chain system is that supplies may not match demands, which results in back orders or lost sales. A back order is a customer order, which cannot be

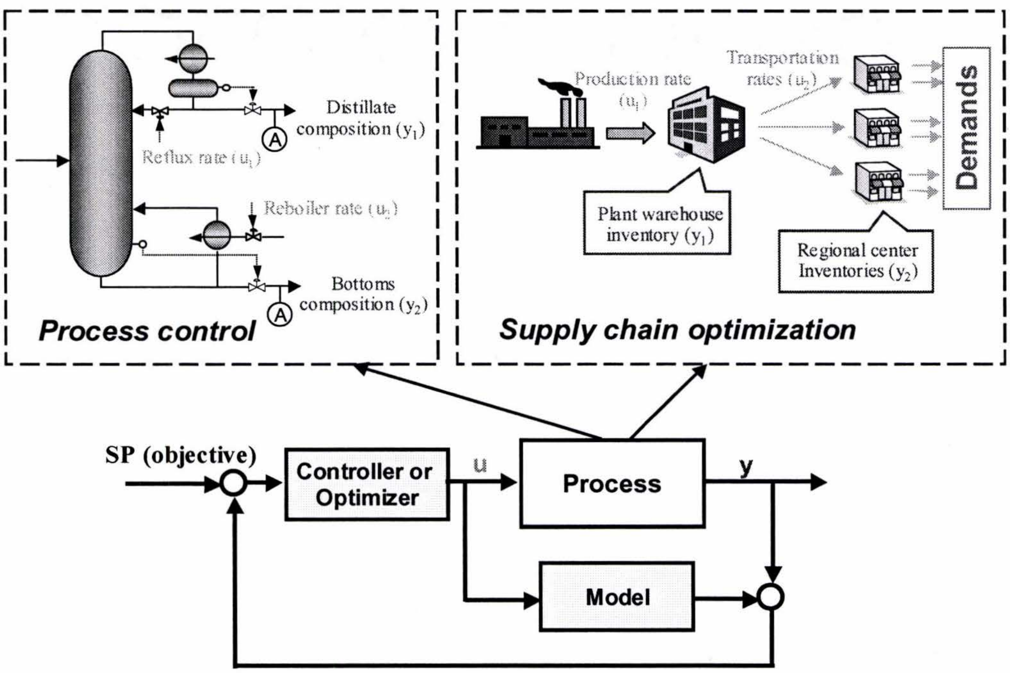


Figure 1.2 The MPC framework for process control and supply chain optimization

filled immediately and for which the customer is prepared to wait for some time (BusinessDictionary, 2009). Naturally, back orders lead to customer dissatisfaction, may incur price penalties, and if persistent, will lead to lost sales. The percentage of items back ordered and the number of the back order days (until the back ordered-products are ultimately delivered) are important measures of the quality of a firm's customer service. Since the uncertainties are usually very large in a supply chain system, we cannot overlook them in supply chain optimization if we want to achieve a satisfactory customer service level. As we shall see, robust MPC, which can address uncertainties explicitly, is an excellent tool for operational supply chain optimization problems.

1.4 Goal and Scope of the Research

The goal of the research is to develop a robust MPC method that has fast and reliable solution as well as good control performance for control and optimization of uncertain systems with feedback, such as occurs in process control and supply chain optimization. Particularly, the method should offer a framework addressing the following issues:

Plant characteristics:

- Linear or nonlinear but can be approximated by linear model with uncertain parameters;
- Multi-input multi-output;
- With strict physical limits on manipulated variables;
- With bounds on controlled variables.

Uncertainty modelling:

- Time-vary or time-invariant uncertainty allowed;
- Correlated, parametric uncertainties of the unmeasured disturbances and noises, the plant/model mismatch and the disturbance plant/model mismatch.
- An estimate of the uncertainty of all transient variables in the system under control

Feedback scheme:

- State feedback (i.e., all the system states are measured) or output feedback (i.e., some of the system states are not measured);
- State-estimation addressing uncertainty explicitly.

Optimization formulation:

1) Hard bounds on the manipulated variables and soft bounds on the controlled variables;
2) Flexible objective function that can contain nominal or expected performance;
3) Reliable and efficient solution for real-time computation;
4) Both steady-state and trajectory optimization considered.

Figure 1.3 shows a more detailed block diagram of the MPC system addressed in the thesis with the variables and parameters explicitly appearing in the robust MPC formulation, where u denotes the manipulated variables decided by the MPC controller (trajectory optimization), y_p denotes the real (uncertain) controlled variables, y_m denotes the measured controlled variables, d_m are the measured information of the true disturbance d_p, w_d and v denote the measurement noises of d_p and y_p respectively, $G_w w_z$ denotes the effects of the noise w_z on the system states through linear channel G_w. All the noises are assumed to be white noises. Also, $\hat{x}$ and $\hat{e}$ denote the estimated states and feedback information for the dynamic system model, b denotes the estimated bias for the steady-state system model, y_{sp} and u_{sp} denote the set points of the controlled and manipulated variables for the trajectory optimization, which are obtained in the steady-state optimization with reference values y_r and u_r that can be determined by an upper-level optimization or prior experience. Note that the steady-state optimization unit may not exist in some applications (e.g., supply chain optimization), and the state-estimation unit is not needed if all of the system states are measured, which is unusual for process control but could be the case for some supply chain systems.

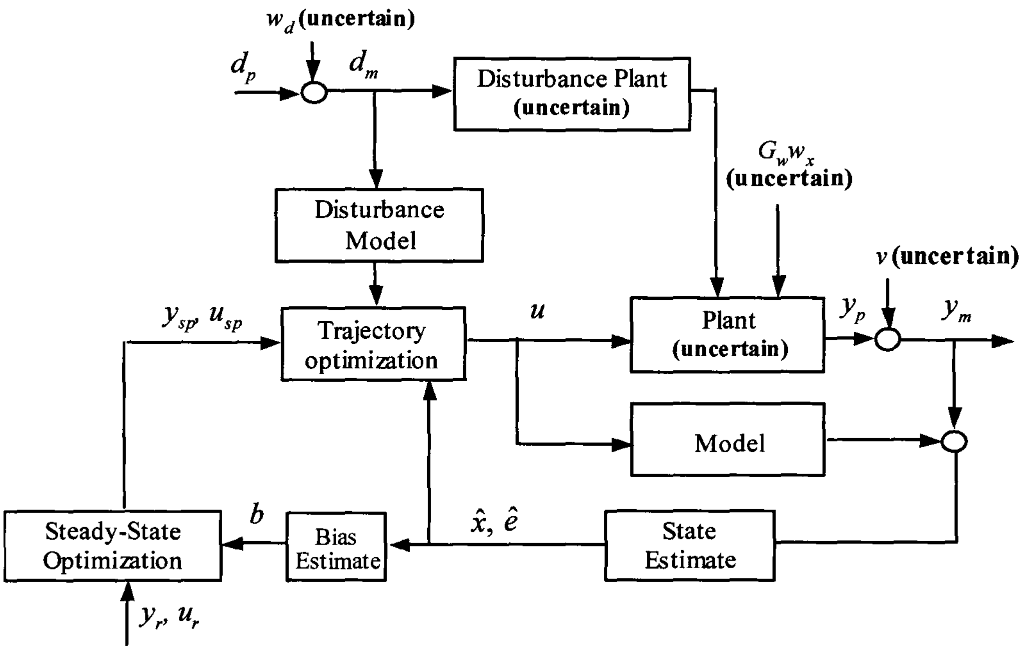


Figure 1.3 Detailed block diagram of the MPC system addressed in the thesis

1.5 Thesis Outline

The remaining part of the thesis is organized as described in the following paragraphs and is shown schematically in Figure 1.4.

Chapter 2 reviews the state-of-the-art robust MPC methods and points out the technology gaps between these methods and the research goals. It also reviews the state-of-the art supply chain optimization under uncertainty.

Chapter 3 develops a new robust MPC framework for the optimization of uncertain system with feedback. This framework addresses various sources of uncertainties of the closed-loop system with hard input bounds. An efficient and reliable optimization solution method and an uncertainty characterization are developed for the real-time implementation of the new robust MPC method.

Optimization of uncertain system with feedback

Goal (Chapter 1, Section 1.3)

Develop a **robust MPC** method that has fast and reliable solution as well as good control performance for the optimization of uncertain systems with feedback. The method addresses linear models with commonly occurring sources of uncertainty and hard limits on manipulated inputs.

Prior Technology (Chapter 2)

State-of-the-art:

1. Parametric, correlated uncertainty
2. Stochastic opt. → SOCP or SDP
3. Individual formulations for either model mismatch or disturbance uncertainty
4. Closed-loop prediction

Gaps:

1. No steady-state opt. with CL uncertainty
2. No unified formulation for model mismatch and disturbance uncertainty with hard bounds
3. Formulation involves simplifications that limit control performance

General Formulation and Solution Method (Chapter 3)

Key contributions

1. A unified formulation addressing: (a) model mismatch and disturbance uncertainties, (b) measured disturbances and the disturbance model mismatch, (c) hard input bounds in closed-loop prediction, and (d) tractable solution for real-time implementation
2. Dynamic deviation model for time-invariant uncertainty
3. State estimation and output feedback under uncertainty
4. Efficient on-line calculation for uncertainty characterization

Case study

1. CSTR control system 1 & 2

Applications of the new robust MPC

Model Predictive Control

(Reach steady state within horizon, **Chapter 4**)

Key contributions

1. Robust steady-state optimization with deviation model for time-invariant uncertainty
2. Objective function for expected performance

Case study

1. Binary distillation control system 1 & 2
2. CSTR control system 3 & 4

Supply Chain Optimization

(No steady-state settling, **Chapter 5**)

Key contributions

1. Tailored state-space model
2. Modified LP to suit the method
3. Addressing non-normally distributed disturbance uncertainty

Case study

1. An industrial supply chain optimization project

Summary and Future Work (Chapter 6)

Figure 1.4 An overview outline of the remaining part of the thesis

Chapter 4 extends the new robust MPC method for the application to process control problems, where a new robust steady-state optimization method is developed to obtain optimal set points for the trajectory optimization and a flexible but convex objective function is developed to include either nominal or expected performance.

Chapter 5 applies the new robust MPC method to an industrial supply chain optimization problem, where both a tailored system model and a tailored robust MPC formulation are developed for the problem. The complexity of this problem and results of computational experiments are reported.

Chapter 6 summarizes the research results and contributions and suggests future research topics.

1.6 Terminology and Conventions

We assume the readers have basic chemical process control and mathematical programming background, so not all the terms in the thesis are explained. We give explanations of some important terms as follows for the ease of the discussions in the remaining part of thesis.

- **Control horizon** (or sometimes called **input horizon** in the thesis) refers to the future time periods (or number of time intervals in discrete control systems) during which the dynamic system behaviours are calculated in MPC.
- **Controlled variables** (or sometimes called **outputs** in the thesis) refer to the variables of a system to be regulated to satisfy some goal (e.g., maintained at specific level or a function of them being optimized).
- **Controller execution period** refers to the time interval between two successive controller decisions.
- **Disturbances** to a system refer to the variables that cannot be adjusted by the controller but affect the controlled variables.
- **Disturbance plant/model mismatch** refers to that the real disturbance plant associating the disturbances with the controlled variables is not exactly described with its model.

- **Manipulated variables** (or sometimes called inputs in the thesis) refer to the variables of a system to be adjusted to influence the controlled variables.
- **Plant** and **process** are essentially synonymous terms. They all refer to any system to be controlled.
- **Plant/model mismatch** refers to that the real plant associating the manipulated variables with the controlled variables is not exactly described with its model.
- **Prediction horizon** (or sometimes called **output horizon** in the thesis) refers to the future time periods (or number of time intervals in discrete control systems) during which the controller decisions are calculated in MPC.
- **Reference** refers to the desired value of the controlled variables (or sometimes include the manipulated variables) at the steady state used in the steady-state optimization. It is obtained from an upper-level optimizer or experience of plant personnel.
- **Saturation** of a manipulated variable refers to the situation where a hard bound posed on this variable is active.
- **Set point** refers to the desired value of the controlled variables (and sometimes the manipulated variables as well) at the steady state used in the trajectory control (optimization).
- **Simulation period** of an MPC method refers to the time interval of the discrete prediction model used by the MPC.
- **Steady-state optimization** refers to the optimization of the steady-state settling point of the system.
- **Trajectory optimization (control)** in MPC refers to the optimization (control) of the dynamic behavior of the system.

The following conventions will be observed in the thesis. Matrices are denoted with uppercase English letters. Scalars and vectors are denoted with lowercase English or Greek letters, except that the extended vectors containing the elements of a vector over prediction or control horizon are denoted with bold English or Greek letters.

Chapter 2

Literature Review

This chapter reviews the state-of-the-art robust MPC methods and the state-of-the-art methods to address uncertainty explicitly in supply chain optimization. The advantages and the disadvantages of the various existing methods are discussed, and the gap between the state-of-the-art technologies and the research goals are outlined.

Since this thesis focuses on explicit handling of uncertainties, this chapter does not review the classical, nominal MPC. Readers can refer to Camacho and Bordons (1999), Macejowski (2000) for more details of the nominal MPC methods. For readers not already familiar with MPC, a brief introduction is given in Section 3.1.

Also, this chapter does not review the supply chain optimization methods that do not address uncertainty explicitly. Readers can refer to Pinedo (2000), Nahmias (2001), Chopra and Meindl (2004) for the review of the extensive work on the "nominal supply chain optimization" methods.

2.1 The State-of-the-art Robust MPC Methods

This section reviews the state-of-the-art methods for robust MPC. First, in Section 2.1.1, we discuss the different approaches to build the uncertain prediction model for robust MPC calculation. This issue is key for a robust MPC method, because it not only determines how good the method can be, but also affects the complexity of the formulation and the tractability of the solution of the method. Then in Section 2.1.2, we discuss the approaches to address other important aspects of robust MPC that has been outlined in Chapter 1. In Section 2.1.3, we summarize some typical robust MPC methods and point out the gap between the state-of-the-art and the research goals. Finally, in Section 2.1.3, we discuss the issue of robust steady-state optimization, which is not addressed in most of the existing robust MPC research but very important for industrial process control systems.

2.1.1 Different approaches for formulating uncertain model

Open-loop or closed-loop prediction?

Nominal MPC predicts the future with a nominal process model that is assumed to be perfect, so it assumes that the forecast of future model errors is perfect and the control sequence obtained at the current time step are unchanged in the future (if the prediction horizon is infinite or sufficiently large). Under these assumptions, the open-loop prediction that does not explicitly consider the effects of the future feedback control is equivalent to the actual closed-loop behavior of the manipulated and controlled variables. So all the existing nominal MPC optimize the open-loop dynamics instead of the equivalent (but more complex) closed-loop dynamics of the system for the simplicity of the formulation. The degrees of freedom of the optimization problem are the future control sequence.

A simple approach to address uncertainty explicitly in MPC calculation is to use an uncertain process model (instead of nominal model) in the same open-loop prediction framework as in nominal MPC. This approach has been adopted in some robust MPC methods, e.g., Badgwell (1997), Schwarm and Nikolaou (1999), Li et al. (2002). However, the open-loop prediction is not equal to the closed-loop prediction in the context of robust MPC, and it may overestimate the uncertainty of the closed-loop system dynamics due to the omission of the future correcting actions of the feedback controller. (See Chapter 3 for more discussion on the limitation of open-loop prediction.) This point has been widely recognized (e.g. Bemporad, 1998) and most of the existing robust MPC methods adopt closed-loop prediction, i.e. using a closed-loop prediction model that includes both a model for describing the uncertain process and a model for describing the future controller actions.

Modeling of closed-loop dynamics – The rigorous approach

The key to building a closed-loop model is modelling the controller behaviour in the future. The rigorous way is to exploit the "Principle of Optimality", which is the basis of the well-known dynamic programming theory (Bellman, 1957). This principle states that in an optimal sequence of decisions, each subsequence must also be optimal, i.e., in the context of MPC (with infinite or sufficiently large horizon), an optimal control sequence obtained at one time step must include the optimal control sequence obtained in any future time steps. Therefore, we can describe the controller actions in robust MPC using a dynamic programming framework, i.e., we can model the controller behavior at end of the horizon (which is easy) first and derive its behavior at the other time steps in a backward mode according to the "Principle of Optimality". Refer to Lee and Yu (1997), Sakizlis et al. (2004) for more details on formulating robust MPC with this idea.

However, the scale of a dynamic programming problem is exponential with respect to the number of possible system states at each decision stage, which makes even small problems suffer the "curse of dimensionality" (Bertsekas, 2000). This widely-recognized drawback of dynamic programming prevents its direct application on most of the real-time problems.

The most popular approach to relieve the "curse of dimensionality" is to approximate the state space with a smaller number of states and the more complicated cost functions with simpler functions, which gives the idea of approximate dynamic programming (Bertsekas and Tsitsiklis, 1998; Lee and Lee, 2004). Refer to Lee et al. (2000), Lee and Lee (2001) and Kaisare et al. (2003) for details of applying approximate dynamic programming to MPC control problems. This approach relies on extensive off-line sampling to validate the approximation; for large or even medium scale problems, it either makes off-line sampling computationally intractable or limits the off-line sampling in a smaller subregion so that the real-time application is only viable within this subregion. So this approach will not work if the real-time operation is outside the subregion of offline sampling, as would occur when the inequality constraint bounds in the system change over the time.

The other approach to address the "curse of dimensionality" is to change the high-dimensional aspects of the calculation from on on-line computation to on off-line computation through parametric programming. The aim of the parametric programming approach is to obtain the optimal solution as a function of the parameters (Dua et al., 2002); once the realizations of the uncertain parameters are known in real-time, the decision can be made by directly evaluating the function of the parameters, which takes little time. The parametric programming approach does not eliminate the "curse of dimensionality"; it still suffers from it in the off-line calculation that may be computationally intractable even for medium scale problems. Also, determining the correct active set is a challenging problem that must be solved in real time. So this approach has the same limitation as approximate dynamic programming.

Modeling of closed-loop dynamics – The approximating approaches

Due to the high computational complexity induced by rigorous modeling approach, many robust MPC methods model the future controller behaviour with an approximating control law that has a simpler structure, so that the results robust MPC formulation is simpler and tractable for real-time applications. In this case, the degrees of freedom of the optimization problem are not the future control sequence, but the parameters of the approximating control law (e.g. the feedback gain of the control law). Linear or affine feedback control laws are widely used for this approximation. For the convenience of discussion, we denote the controller decisions (or the manipulated variables) at the i^{th} time step in the horizon as u_i, the states and controlled variables at the i^{th} time step in the horizon as x_i, y_i respectively. Then, we can summarize the linear and affine control laws used in the different robust MPC methods into five types:

Type 1: $u_i = Kx_i$ (e.g., Kothare et al., 1996), where K denotes the constant feedback gain matrix throughout the horizon, but it needs to be evaluated at the beginning of each robust MPC execution period. This control law is a simple proportional expression, and it can be used to develop a convex robust MPC formulation that guarantees robust stability (see Section 2.1.2 for more details); however, it can not describe the input saturation in the future horizon, because in general (unless K is a zero matrix) it requires u_i to be different for different x_i (under different plant realization).

Type 2: $u_i = K_{offline}x_i + c_i$ (e.g., Kouvaritakis et al., 2000), where $K_{offline}$ denotes the constant feedback gain throughout the horizon that is obtained offline, c_i denotes the variable perturbed term that is evaluated at the beginning of each robust MPC execution period. The degrees of freedom of the resulting robust MPC optimization problem are the perturbed term c_i (whose effects on u_i is independent of the plant realization), while the feedback gain $K_{offline}$ is constant (whose effects on u_i is dependent on the plant realization). This optimization problem is much easier to solve than the problem based on Type 1 control law. However, Type 2 control law loses the flexibility to adjust the feedback gain in the real time, and it cannot describe the input saturation in the future horizon as well.

Type 3: $u_i = Kx_i + c_i$ (e.g., Bemporad, 1998), where K and c_i denote the same variables defined in the above Type 1 and Type 2, and they are evaluated at the beginning of each robust MPC execution period. Obviously, the type of control law includes the Type 1 and Type 2 control law, and it can describe some special input saturation situations (e.g. the input saturation is held throughout the horizon). However, it cannot model the input saturation occurring during the transient response, because it requires the same gain matrix K for the whole prediction horizon. Also, it leads to a much more complicated optimization problem for robust MPC and the efficient solution can be obtained only for special cases. For example, Bemporad (1998) developed robust MPC method based on Type 3 control law for the problems where the only source of uncertainty is the unmeasured disturbances.

Type 4: $u_i = K_ix_i + c_i$ (e.g., Goulart et al., 2006), where K_i denotes the variable feedback gains. Both K_i and c_i are evaluated at the beginning of each robust MPC execution period. Obviously, this type of affine state feedback law includes the Type 3 feedback law (so that it also includes Type 1 and Type 2 feedback laws), and the input saturation can be well described by this type of control law (by forcing the corresponding elements in K_i to be zero). However, using this control law in closed-loop prediction results in the robust MPC formulation being highly nonconvex, which is difficult to solve in the real-time (Goulart et al., 2006).

Goulart et al. (2006) proved that, if the process model is perfect and the uncertainty only comes from only the unmeasured disturbances, Type 4 control law is identical to a unmeasured disturbance feedback, $u_i = K_i^* w_i + c_i^*$ (where w_i denotes the unmeasured disturbances at the i^{th} time step in the horizon). To use this unmeasured disturbance feedback in the closed-loop prediction leads to a convex optimization robust MPC formulation, which can be solved efficiently and reliably in the real-time (see more details in Section 2.1.2). This clever mathematical transformation can be understood as the following: when the unmeasured disturbances are the only source of uncertainty, they equal to the difference between the measured and predicted states, so an affine state feedback is equivalent to an affine unmeasured disturbance feedback. However, in general an unmeasured disturbance feedback is not theoretically sound for control problems, because we have no exact information about future unmeasured disturbances.

Therefore, Type 4 control law is a better approximation than Type 1-Type 3 control laws, but it can lead to a practical robust MPC formulation only when the source of uncertainty is solely the unmeasured disturbances.

Type 5: $u_i = K_i y_i + c_i$ (e.g., Van Hessem and Bosgra, 2006), which is similar to Type 4 control law but uses output feedback instead of state feedback. Its properties are similar to that of Type 4 control law.

There are other approximating control laws in addition to the linear or affine feedback laws discussed above. For example, Warren (2004) used unconstrained nominal MPC with variable output references to approximate the robust MPC in the closed-loop prediction. The degrees of freedom of the resulting robust MPC optimization problem are the references of the controlled variables at different time steps in the prediction horizon. This problem is convex and easy to solve in the real time. However, the input saturation was not modeled. Warren (2004) reduced the model inaccuracy due to the input saturation by partitioning the original uncertainty region into several small subregions and solving for different output references for uncertainties in the different subregions.

2.1.2 Approaches for other aspects of robust MPC

This section introduces the approaches to address other aspects of robust MPC in the research goads outlined in Chapter 1. We discuss these aspects one by one.

Source of uncertainty

As mentioned in Chapter 1, the source of uncertainty in the system can be the plant/model mismatch, the measured disturbance plant/model mismatch and the unmeasured disturbances and noise. Any robust MPC method has to address one or more of these uncertainties. Generally, the uncertainty in the plant/model mismatch is more difficult to handle than the uncertainty in the measured disturbance plant/model mismatch and the unmeasured disturbances, because the effects of the former on the system depend on the decisions, while the effects of the latter are independent of the decisions.

Most robust MPC methods only address part of these sources of uncertainties. Some address plant/model mismatch only, e.g. Kothare et al. (1996), Badgwell (1997), and Kouvaritakis et. al. (2000). Some others address unmeasured disturbances only, e.g. Bemporad (1998), Goulart et al. (2006), Van Hessem and Bosgra (2006).

Warren (2004)'s method can handle plant/model mismatch and measured disturbance plant/model mismatch as well as time-invariant unmeasured disturbances and noise, but these uncertainties are not addressed in a unified framework. Lee and Yu (1997)'s addressed plant/model mismatch only in their method that is based on approximate dynamic programming; however, their method can also address other sources of uncertainty (although the resulting robust MPC formulation is more complicated).

Temporal manner of uncertainty

The uncertainty in a system could be time-invariant or time-varying. If a robust MPC method can handle time-varying uncertainty, it can naturally handle time-invariant uncertainty (although additional work may need to be done to exploit the time-invariant characteristics for better control performance). Examples of the methods capable of handling time-varying uncertainty are Kothare et al. (1996), Lee and Yu (1997), Goulart et al. (2006). Examples of the methods only addressing time-invariant uncertainty are Badgwell (1997), Warren (2004).

Description of uncertainty

The way to describe the uncertainty is important for a robust MPC method because it not only determines how accurate the uncertainty is described, but also affects the complexity of the robust MPC calculations. Since this thesis only addresses the parametric uncertainty, we only discuss the description of uncertain parameters here. Typically, there are four types of approaches to describe the uncertain parameters:

1) **Multi-plant (Scenario based description):** This approach samples representative realizations of the uncertain plant or disturbances and addresses these realizations only in the robust MPC formulation. Advantages of this method are that it is easy to formulate the optimization problem and the resulting robust formulation has the same linearity/nonlinearity property as the nominal formulation (e.g. if the nominal formulation is linear, the robust formulation is linear too). However, it is usually difficult to choose the representative samples from all the possible uncertainty realizations, and even if these samples are correctly selected, the number of these samples is usually large, and the resulting problem is too large to solve. Some robust MPC methods use this description for the uncertainty in the plant/model mismatch, e.g. Badgwell (1997).

2) **Polytopic uncertainty region:** This approach assumes the uncertainty lies within a polytopic uncertainty region. This region is usually in the form of a convex hull of a series of sample realizations, and one can address this region by addressing these sample realizations. So this method leads to a similar formulation as the method using multi-plant description. Many robust MPC methods use this uncertainty description, e.g. Bemporad (1998), Kouvaritakis et al. (2000), and Sakizlis et al. (2004).

3) **Structured uncertainty region:** The term "structured uncertainty" here denotes the uncertain parameters having significant correlations among one another. Ellipsoidal uncertainty region is a typical structured uncertainty region, where the size and shape of the ellipsoid indicates the correlations among the parameters. It can be described

by bounds on the norms of uncertain vectors or matrices. This description usually makes the robust MPC formulation more difficult to solve than the nominal formulation, e.g. if the nominal formulation is linear, the robust formulation becomes nonlinear (see more details in the later discussion on on-line optimization). However, in many process control problems the uncertainty parameters are strongly correlated, so the structured uncertainty description is more desirable than the above two description approaches. Therefore, many robust MPC methods are designed to be able to adopt both the structured uncertainty and the polytopic uncertainty descriptions, e.g. Kothare et al. (1996), Lee and Yu (1997), Goulart et al. (2006).

4) **Multivariate continuous distribution:** This approach characterizes the uncertainty with multivariate distribution of continuous variables, where the correlations among the uncertain variables are described with their covariance matrix. So this approach can also be used to describe the correlated uncertainties, and the resulting robust MPC stochastic optimization problem can be transformed (equivalently or approximately) into a (more complicated) deterministic problem with a given confidence level by chance-constrained programming technique (Charnes and Cooper, 1958; Kleywegt and Shapiro, 2001) (see more details in the later discussion on on-line optimization). Some robust MPC methods use this approach to describe correlated uncertainty, e.g., Warren (2004), Van Hessem and Bosgra (2006).

Feedback Scheme

Most of the robust MPC methods assume that all the states of the system can be measured directly at the beginning of each time step, which means the system has full state feedback. However, in many real process systems, not all the states can be measured; the controller has to infer the states from the limited measurements of the outputs. In this case, one needs to use an output feedback scheme to estimate the states for the robust MPC calculation. The theory for output feedback and the state estimation has been well established for nominal problems (see Appendix A for more details), and some robust MPC methods have adopted this theory, e.g. Van Hessem and Bosgra (2006). However, applying a nominal estimation method will deteriorate the performance of robust MPC because the uncertain error in the nominal estimation of the states is not explicitly addressed. There are some robust state estimation methods that are developed to guarantee the convergence of the state estimation to the real states, e.g. Mangoubi (1998), Xie et al. (1994), but these methods do not obtain the uncertain estimation error at each controller execution for the use in the controller calculation. According to the author's knowledge, the uncertainty in the state estimation error is not addressed in any of the existing robust MPC methods explicitly in each controller calculation.

Note that the state-estimation may not be needed if an input-output model (instead of state-space model) is used in a robust MPC methods model and all the controlled variables are assumed to be measurable (e.g., Warren, 2004; Wang and Rawlings 2004).

Objective function

When optimizing the dynamics of a system, the most commonly used objective function is the sum of the squared differences between the controlled and manipulated variables and their desired values over the horizon, which is quadratic and with appropriate tuning, convex. With the presence of uncertainty, however, this function becomes uncertain.

A natural choice for the objective of robust MPC is the expected value of the dynamic performance function. This idea has been adopted in the robust optimization research in Darlington et al. (2000) and Darlington et al. (1999).

Another choice is the worst-case value of the dynamic performance. This idea has been widely adopted in robust MPC research (e.g. Kothare et al., 1996; Kouvaritakis et al., 2000; Lee and Yu, 1997). This is because using this objective makes the robust stability easier to be guaranteed. However, optimizing worst case dynamic performance may lead to conservative control. So, some robust MPC methods optimize nominal performance, e.g. Goulart et al. 2006, Warren (2004). Badgwell (1997) proposed to include both the nominal performance and an additional stabilizing term to guarantee robust stability for time-invariant uncertainty in plant/model mismatch with multi-plant description.

An ideal objective function would evaluate the entire dynamic response for all realizations of the uncertain parameters, from which expected value and other characteristics could be determined. However, the statistical information on parameter uncertainty would not typically support these calculations, and the resulting algorithm would be intractable.

Robust stability

Stability is an important property for a closed-loop control system because if it cannot be achieved, the performance will always be unacceptable. In the context of MPC, the concept of Lyapunov stability is usually used to define the stability of the closed-loop system, and it can be guaranteed by forcing a Lyapunov function to decrease at each time step (details on Lyapunov stability theory can be found in Haddad and Chellaboina, 2008). In many robust MPC methods (e.g. Kothare et al. 1996; Kouvaritakis et al., 2000), the worst-case dynamic performance, which is a Lyapunov function, is the objective function that the controller minimizes. Interesting readers can refer to the summary papers of Bemporad and Morari (1999) and Mayne et al. (2000) for more discussions on robust stability of robust MPC.

This thesis is primarily aiming at the optimization, instead of the stabilization, of the dynamics of a system; therefore, the controller is not designed with a robust stability guarantee. We note that common industrial practice using commercial nominal MPC software for process control does not implement the constraints guaranteeing nominal stability; instead, ad hoc approaches are integrated in the control calculation (Qin and Badgwell, 2003), which have been successful to ensure stability in practice, especially for open-loop stable system and integrating system. Also, stability has not been reported to be an issue in supply chain optimization.

On-line optimization problem

The optimization problem to be solved on-line is determined by how the robust MPC is formulated to address all the aspects discussed above, especially the way to model the closed-loop dynamics. When the closed-loop dynamics are modelled rigorously using the dynamic programming framework (e.g., Lee and Yu 1997), the resulting optimization problem is a dynamic programming problem, which can be solved by approximate dynamic programming technique (e.g. Lee and Lee, 2004) on-line or the parametric programming technique off-line (e.g. Sakizlis et al., 2004). However, as discussed in Section 2.1.1, the techniques offer computational challenges for large problems.

When the closed-loop dynamics are modelled approximately as discussed in earlier part of this section, the closed-loop model is linear with uncertain parameters. Then the resulting robust MPC formulation contains linear inequalities with uncertain parameters. This formulation can be basically transformed into deterministic Quadratic Program (QP) or Second Order Cone Program (SOCP) depending on the source and description of uncertainty. Also, if the robust MPC optimizes the worst dynamic performance for the stability guarantee, it can be transformed into Semi-Definite Programming (SDP) (or called Linear Matrix Inequality) (LMI) problems. Examples of robust MPC solving QP are Badgwell (1997) and Bemporad (1998); examples solving SOCP are Warren (2004) and Goulart et al. (2006); examples of solving SDP are Kothare et al. (1996) and Kouvaritakis et al. (2000).

QP, SOCP and SDP are all convex optimization problems where the local optimal objective value is the same as the global optimal value (Boyd and Vandenberghe, 2004). In general, QP is easier to solve than SOCP, and SOCP is easier to solve than SDP (Lobo, et al., 1998); but all of these problems can be solved in polynomial time using the interior point method (or called barrier method) (Nocedal and Wright, 1999; Boyd and Vandenberghe, 2004). Many state-of-the-art optimization solvers are featured with interior point method, e.g. CPLEX (ILOG Inc., 2008), IPOPT (Wachter and Biegler, 2002), and SeDuMi (Sturm, 1999).

2.1.3 Summary of the "representative" robust MPC methods

In Sections 2.1.1 and 2.1.2 we discussed the typical approaches to address the different aspects of robust MPC in the literature. In this section we show some representative robust MPC methods with their approaches to address these different aspects summarized in Tables 2-1 and 2-2, which gives an overview of the state-of-the-art in the robust MPC research with a different perspective. Here the word "representative" is used to describe the methods featured with typical approaches to address some important aspects of robust MPC, and each of the methods representatives a typical approach for formulating the problem. Actually, there are some other methods that are different from all the methods in the tables; we do not show them here because their ways to address the different aspects of robust MPC have been covered by one or more of the methods in the tables, or the special features they have are not included in the goals of research.

Tables 2-1 and 2-2 indicate the gap between the state-of-the-art robust MPC research results and our research goals. Specifically, the following issues are not well addressed:

Table 2-1: Summary of robust MPC methods: modeling of closed-loop uncertainty

Authors	**Controller model in closed-loop prediction**	**Source and temporal manner of uncertainty**	**Uncertainty description**
Kothare et al. (1996)	$u_i = Kx_i$	Time-varying plant/model mismatch	Structured or polytopic region
Lee and Yu [1] (1997)	Dynamic program	Time-varying plant/model mismatch	Structured or polytopic region
Badgwell (1997)	No controller (open-loop prediction)	Time-invariant plant/model mismatch	Multi-plant
Bemporad (1998)	$u_i = Kx_i + c_i$	Time-varying disturbances	Ploytopic region
Kouvaritakis et al. (2000)	$u_i = K_{offline} x_i + c_i$	Time-varying plant/model mismatch	Ploytopic region
Sakizlis et al. (2004)	Dynamic program	Time-varying disturbances	Polytopic region
Warren (2004)	Unconstrained nominal MPC	Time-invariant plant/model mismatch and time-varying disturbances	Multivariate and continuous distribution
Goulart et al. (2006)	$u_i = K_i x_i + c_i$	Time-varying disturbances	Structured or polytopic region
Van Hessem and Bosgra (2006)	$u_i = K_i y_i + c_i$	Time-varying disturbances	Multivaraite and continuous distribution

Note: [1] Their method can be extended to include all the other sources of uncertainties.

Table 2-2: Summary of robust MPC methods: feedback and optimization formulation

Authors	**Feedback scheme**	**Objective Function**	**Robust stability included?**	**Online optimization problem**
Kothare et al. (1996)	State feedback	Upper bound of the worst performance	Yes	SDP
Lee and Yu (1997)	State feedback	The worst performance	Yes	A huge number of QP [4]
Badgwell (1997)	State feedback	Nominal performance+ stabilizing term	Yes	QP
Bemporad (1998)	State feedback	Nominal performance	No	QP
Kouvaritakis et al. (2000)	State feedback	Upper bound of the worst performance	Yes	SDP
Sakizlis et al. (2004)	State feedback	Nominal or expected performance	Yes[3]	Evaluating parametric solution of QP and LP (obtained offline)
Warren (2004)	Output feedback without state estimation [1]	Nominal performance	No	SOCP or QP [5]
Goulart et al. (2006)	State feedback	Nominal performance	No	SOCP or QP [6]
Van Hessem and Bosgra (2006)	Nominal output feedback [2]	Nominal performance	No	SOCP

Note: [1] The author uses the input-output model instead of state-space model to describe the process; [2] Here "nominal" means the uncertainty in the state estimation is not considered explicitly; [3] The robust stability is guaranteed only for constant uncertain disturbances; [4] The problem can also be solved by approximate dynamic programming technique (see Lee and Lee (2004) for more details); [5] The formulation is QP if uncertainty only appears in disturbance; otherwise it is SOCP; [6] The formulation is SOCP for structured uncertainty and QP for polytopic uncertainty.

1) The trade-off between the rigorousness of the formulation and the tractability of the solution. The closed-loop prediction model used in the formulation is either too complex for a tractable solution of a large problem (e.g., the approaches using dynamic programming), or lack accuracy in modeling the controller behaviors (e.g., approximating linear feedback control law can not model input saturation).
2) An output feedback and state-estimation scheme that exploits uncertainty explicitly. All the methods in the tables consider state feedback only or output feedback without including uncertainty in the state estimation explicitly.
3) An objective function that can include expected performance and the variances of the controlled variables. All the methods in the tables optimize either the worst performance only or the nominal performance only.
4) There is no unified framework that can well address all the aspects listed. Each method in the tables has advantages in addressing some issues but is limited in addressing other issues.

2.1.4 Robust steady-state optimization

Industrial MPC control systems usually include a steady-state optimization unit that is executed immediately before each controller execution (Qin and Badgwell, 2003). It is formulated to find a feasible "settling point" or steady state of the system that is close to the reference values of the controlled and manipulated variables that are determined by an upper-level optimizer or by plant personnel. The desired steady state is called the set point of the system, which is used by the MPC controller to regulate the dynamics of the system. The steady-state optimization is important because disturbances entering the system or new input information from the operator may change the location of the optimal steady state.

A nominal steady-state optimization may give infeasible set points with the presence of uncertainty. In this case, a robust steady-state optimization method that addresses uncertainty explicitly in the calculation is required. Although it is important for a system with significant uncertainty, robust steady-state optimization has not been addressed in most of the robust MPC research. Some results on robust steady-state optimization can be found in Kassmann et al. (2000) where open-loop parametric uncertainty described with multivariate continuous distribution is addressed, or Wang and Rawlings (2004) where open-loop parametric uncertainty described with polytopic uncertainty region is addressed. Due to the explicit integration of the uncertainty in the formulation, these two methods reduces the chance of generating infeasible steady-states, but they may be overly conservative on uncertainty estimation of steady-state because of the omission of the controller action in the closed-loop (as we discussed before for the uncertainty estimation of dynamics). According to the author's knowledge, no research has been published for a robust steady-state optimization method addressing closed-loop uncertainty.

2.2 Supply Chain Optimization Under Uncertainty

This section reviews the state-of-the-art methods to address uncertainty explicitly in supply chain optimization. The review includes two major topics: a) In Section 2.2.1, we discuss the techniques for optimization under uncertainty; b) In Section 2.2.2, we discuss the control strategies to address uncertainty with feedback. The optimization methods with deterministic models, such as mathematical programming (see the review paper Biegler and Grossmann, 2004), constraint programming (see the review paper Lustig and Puget, 2001), are not reviewed here.

2.2.1 Methods for optimization under uncertainty

Stochastic Programming

Stochastic programming is an approach for modeling and solving optimization problems that involve uncertainty (Shapiro and Philpott, 2007), where an expected cost/performance of the system is optimized with the known distribution of uncertainty. A variability measure (e.g., the variances of some key system variables) can also be included in the objective function to capture the notion of risk (Sen and Higle, 1999). There are typically two types of formulations of stochastic programming:

1) **Chance-constrained programming**: The idea of chance-constrained programming is to satisfy the constraints with a specified confidence level (a lower threshold for the

probability of the satisfaction of the constraints) (Charnes and Cooper, 1958; Kleywegt and Shapiro, 2001). The problem can be formulated so that the confidence level can be observed for each individual constraint or all the all the constraints together. The idea of solving a chance-constrained program is to transform it into a deterministic optimization problem, which is in general not easy because of the need of the integration of multivariate distribution functions (Li et al., 2008). However, if the parameters obey multivariate normal distribution and the confidence level is observed for each individual constraint, the problem can be transformed into a deterministic SOCP that is convex and can be solved using standard software (Lobo, et al., 1998).

2) **Stochastic programming with recourse**: Another type of formulation is based on a "wait-and-see" analysis for multi-stage decision-making (Sen and Higle, 1999; Shapiro and Philpott, 2007). The formulation mimics the following decision-making procedure: the decision maker takes some action in the first stage, after which a random event occurs affecting the outcome of the first-stage decision. A recourse decision can then be made in the second stage that optimizes the remaining problem, but cannot change the first stage decisions. The optimal policy from such a formulation is a single group of first-stage decisions and collections of recourse decisions (decision rules) defining the actions that should be taken in response to random outcomes in the future stages. The stochastic recourse formulation is usually solved by addressing representative samples of the uncertain parameters, which makes the problem suffer from "curse of dimensionality", especially as the number of stages expands beyond two, as is the case in the systems considered in this research (Kleywegt and Shapiro, 2001).

Robust Optimization

Robust optimization is another approach for modeling and solution of optimization problems that involve uncertainty (Ben-Tal and Nemirovski, 2002), where the worst cost/performance of the system is optimized with the uncertainty described by uncertainty region. Similar to stochastic programming, robust optimization is solved by transforming the original problem into a deterministic problem. Several types of robust optimization problems can be transformed into convex deterministic optimization problems such as LP, SOCP and SDP (See Ben-Tal and Nemirovski, 1999; Ben-Tal and Nemirovski, 2002) for more details.

Dynamic Programming

Dynamic programming is an approach for the modeling of dynamic and stochastic decision problems as well as the solution of these problems. Kleywegt and Shapiro (2001) pointed out that dynamic programming offers another framework and solution approach for multistage stochastic programming. As discussed in Section 2.1.1, the advantage of dynamic programming lies in its ability to rigorously model a sequential-decision making problem with explicit uncertainty. However, it is only applicable for small problems due to the "curse of dimensionality". Approximating solution techniques for dynamic programming problems, such as approximate dynamic programming (Lee and Lee, 2004), have been developed to achieve better efficiency of on-line calculation by using less complicated approximating formulation or limiting the application to smaller subregion of closed-loop uncertainty. But as we discussed in Section 2.2.1, these methods will not work if the real-time operation is outside the subregion of offline sampling or the constraint values on the system change over the time, which is usual for process control and supply chain systems.

Parametric Programming

Parametric programming obtains the optimal basis as a function of the parameters off-line (Dua et al., 2002) and evaluates the parametric solution according to the known realizations of the uncertain parameters on-line. Obviously, parametric programming makes the on-line calculation very fast, but the off-line procedure to obtain the parametric solution is much more complicated. Examples of the application of parametric programming on small problems can found in Ryu et al. (2007), Ryu and Pistikopoulos (2007). According to the author's knowledge, no applications on medium or large problems have been published.

2.2.2 Control strategies for the uncertain system with feedback

As stated in Chapter 1, the supply chain operation problems are analog to process control problems. In both type of problems new information of the system is available periodically at each decision-making stage, which can be exploited to regulate the dynamics of the system so that the uncertainty is addressed in an implicit way. In process control, different approaches to exploit the new information appear as the different feedback schemes and controller algorithms, which can also be used for supply chain operation problems.

The systematic research on the application of control strategies for supply chain optimization can be traced back to late 1950s, when Forrester (1958; 1961) introduced his pioneering work on so-called "industrial dynamics". This methodology, later referred as "system dynamics", used a feedback perspective to model, analyze and improve industrial dynamics systems such as production-inventory systems. The philosophy of this methodology forms the basis for the application of control technologies to supply chain optimization problems. For more details of the "system dynamics" philosophy and its various applications, see Sterman (2000).

As it is widely used in process control, PID control methods have been applied to supply chain optimization, such as Proportional (P) control (e.g. Perea-Lopez et al. 2001), Proportional-Integral (PI) control (e.g. Lin et al., 2004). It was shown that the "bullwhip effect" (Lee et al., 1997), which denotes the phenomenon that the variability of the demand at a downstream node is amplified at a upper stream node, can be relieved or reduced by proper tuning of the controllers (e.g., Perea-Lopez et al. 2001; Lin et al., 2004).

However, the classical PID control methods have inherent limitations that could prevent their application to real supply chain systems: a) They can not handle the constraints on the system explicitly; b) They basically pose a decentralized control structure that does not share the information between the different control loops. The system may not achieve the best overall performance through these "local controllers". Perea-Lopez et al. (2003) showed though a supply chain optimization case study that the centralized controller, which makes decisions according to all the available information in the system, performs better than the decentralized controller.

As stated in Chapter 1, MPC is an effective means for multivariate constrained control (Maciejowski, 2002) that has been widely applied in industry (Qin and Badgwell, 2003). It is a natural choice for the centralized control of supply chain systems with constraints. Therefore, the applications of MPC to supply chain optimization has been paid much attention recently, e.g., Tzafestas et al. (1997), Braun et al. (2003), and Perea-Lopez et al. (2003). Seferlis and Giannelos (2004) employed a two-layer optimization framework for supply chain optimization, where an MPC controller is used as the upper layer controller for the entire supply chain system and PID controllers are used as the lower layer controllers to maintain the safety stocks at different nodes of the system.

However, the performance of MPC control can be degraded by the uncertainties, especially for supply chain system where uncertainties are usually significant. For example, back orders may occur due to the inaccurate demand forecast or the mismatch between the supply chain system and the nominal model used by MPC.

To prevent significant performance degradation caused by uncertainties, Wang et al. (2007) introduced an upper level stochastic optimizer that executes infrequently and provides constraint back-off parameters to the lower level MPC controller. However, this approach cannot respond quickly to changes in the control structure, such as when a manipulated variable is temporarily placed on manual or is placed in operation after having been in manual.

A better way to address the uncertainties is to include them in the controller calculation explicitly at every controller execution period. In the context of MPC, this means to use the robust MPC for supply chain optimization. For example, Warren (2004) successfully applied robust MPC to a generalized production planning problem. However, very little work has been published on such applications, which may be due to the lack of a flexible robust MPC method that includes the various necessary features required for the application to real supply chain systems (e.g. ability to address different sources of uncertainties, efficient real time calculation, etc.).

2.3 Summary

The purpose of this chapter is to outline the gaps between the state-of-the-art technologies and the research goals by reviewing the existing robust MPC methods and the existing methods to address uncertainties in supply chain optimization. We summarize all the discussions in this chapter as follows.

A practical robust MPC method has to build the formulation in the way that the optimization problem to be solved on-line is tractable, and it also has to model the closed-loop dynamics accurately enough for good control performance. In addition, different features should be included in the robust MPC formulation for different goals of control (or optimization). Therefore, a good robust MPC method should be able to keep a good trade-off between the accuracy and features of the formulation and the efficiency of the solution. Among all the aspects of robust MPC discussed in this chapter, the modeling of closed-loop dynamics, or more specifically, the modelling of the controller behaviour in the prediction horizon, is key to a robust MPC method, because it not only determines the accuracy of the formulation, but also impacts on the efficiency of the solution.

After examining the "representative" robust MPC methods, we can conclude the following major drawbacks of the robust MPC technologies according to the research goals:

- Lack of a closed-model that addresses the input saturation in the prediction and yields tractable on-line solution, for different sources of uncertainties;
- Lack of a unified framework that well addresses all the aspects of interest;
- Lack of some features in the formulation, which may be important for particular problems: a) an output feedback and state-estimation scheme that exploits uncertainty explicitly; b) a flexible objective function that can include expected performance and the variances of controlled variables; c) a robust steady-state optimization method addressing closed-loop uncertainty.

The techniques for optimization under uncertainty and the control strategies are the two categories of methodologies to address uncertainties in supply chain optimization, in which the uncertainties are addressed explicitly in the optimization formulation or implicitly by exploiting the periodical feedback information. A more powerful tool for robust supply chain optimization can be developed by merging the two types of methodologies in a unified framework of robust MPC, which can essentially improve the performance of the optimization of uncertain supply chain system with periodical feedback information.

Therefore, a new robust MPC method needs to be developed in this research, which should offer a unified framework that provides a good approximation of closed-loop dynamics, includes all the important features of interest as well as yields efficient on-line solution. The following chapter develops the general framework of the new robust MPC method, and tailored formulations using this framework are developed for process control problems in Chapter 4 and for supply chain optimization problems in Chapter 5.

Chapter 3

A General Framework for Robust MPC

In this chapter, a general framework is presented for the new robust MPC method to optimize an uncertain dynamic system with feedback. Recall that the general problem is to achieve good dynamic performance with robust feasibility for a dynamic system with uncertain parameters.

This framework is developed based on the nominal MPC formulation using state-space model, which is introduced in Section 3.1. The extension to robust MPC is explained in detail in Sections 3.2 to 3.4. The original robust formulation is a bilevel stochastic optimization problem where the inner optimization problems model the behavior of the MPC controller in the closed-loop. Methods for solving this problem could involve either unrealistically large numbers of variables and equations with a scenario-based uncertainty description or highly nonconvex with continuous parametric uncertainty, so it is computationally intractable for real-time applications. Here, we consider the continuous parametric uncertainty in the robust MPC and develop a series of reformulations and approximations that yield tractable computation for real-time applications. Figure 3.1 shows the "road map" of the development of the new robust MPC method, which outlines the key steps of the development and their locations in this chapter.

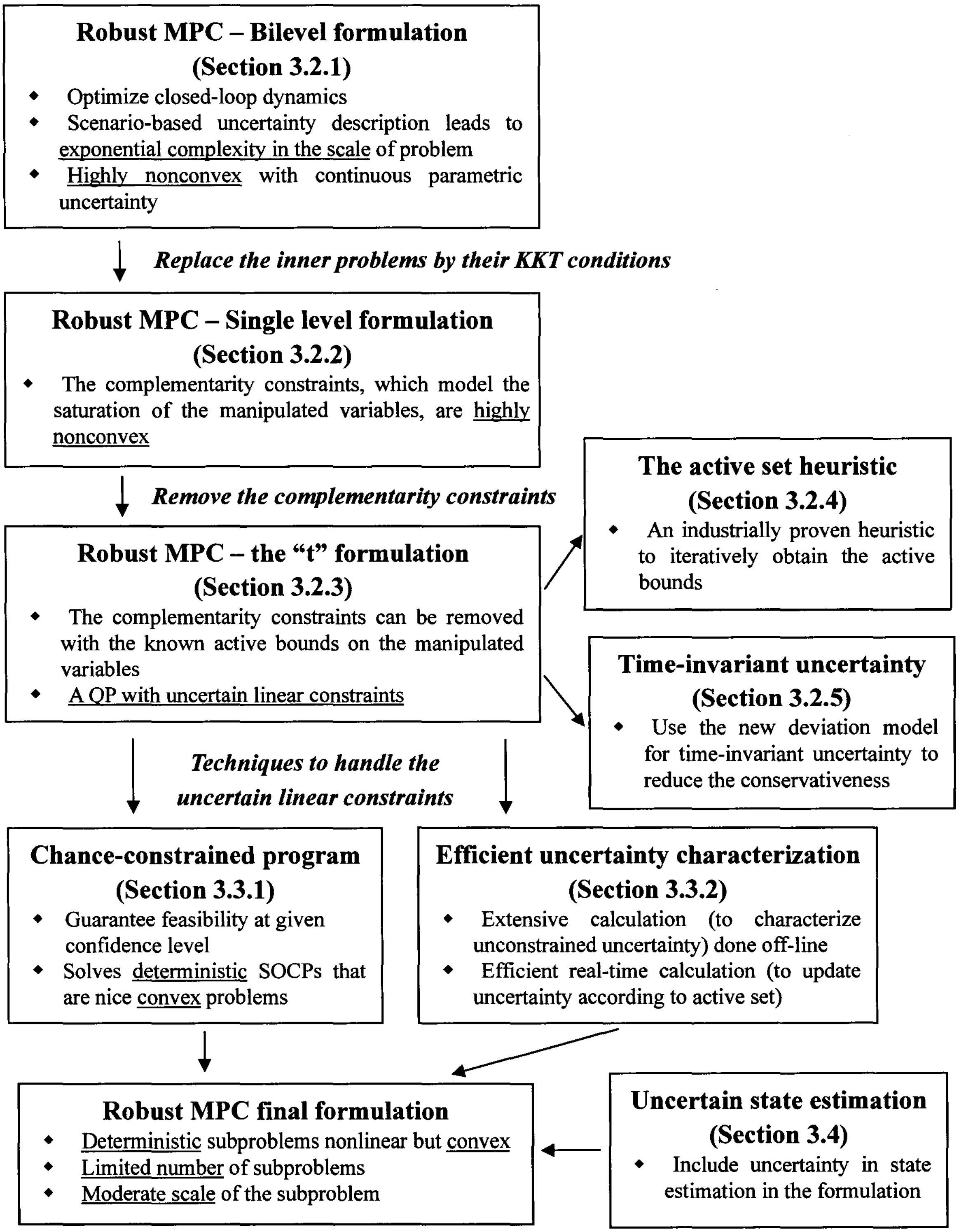


Figure 3.1 "Road map" for developing the new robust MPC method

The new robust MPC method is summarized in Section 3.5 with detailed steps for its implementation. The simulation studies are reported in Section 3.6 for a number of simple applications that clearly show the benefits of using the new robust MPC method, and the conclusions of the chapter are summarized in Section 3.7.

3.1 Nominal MPC formulation – Basis of the Robust MPC

We introduce the nominal MPC in this section, because it forms a basis for the discussion of the new robust MPC. The conventional nominal MPC determines the current control action by solving, at each sampling time interval, a finite horizon open-loop optimal control problem, using the current state of the process as the initial state for a dynamic optimization. The open-loop optimization is written in this thesis as follows:

NMPC:

$$\min_{u_k, s_{k+1}} \sum_{k=0}^{p-1} (y_{k+1} - y_{sp})^T Q (y_{k+1} - y_{sp}) + \sum_{k=0}^{n-1} (u_k - u_{sp})^T R (u_k - u_{sp}) + \sum_{k=0}^{n-1} \Delta u_k^T W \Delta u_k + \sum_{k=0}^{p-1} s_{k+1}^T W_s s_{k+1} \tag{3.1a}$$

$$\text{s.t.} \quad \left.\begin{array}{ll} x_{k+1} = Ax_k + Bu_k + B_e e_k + B_d d_{m,k} & (3.1b) \\ y_{k+1} = Cx_{k+1} & (3.1c) \\ e_{k+1} = e_k & (3.1d) \\ u_{\min,k} \le u_k \le u_{\max,k} & (3.1e) \\ y_{\min,k+1} - s_{k+1} \le y_{k+1} \le y_{\max,k+1} + s_{k+1} & (3.1f) \\ s_{k+1} \ge 0 & (3.1g) \end{array}\right\} k = 0, \cdots, p-1$$

$$\Delta u_k = u_k - u_{k-1}, \qquad k = 0, \cdots, n-1 \tag{3.1h}$$

$$u_k = u_{k-1}, \qquad k = n, \cdots, p-1 \tag{3.1i}$$

$$x_0 = \hat{x}_0, \quad e_0 = \hat{e}_0 \tag{3.1j}$$

where n is called control horizon (or input horizon), p is called prediction horizon (or output horizon), $y_k \in R^{n_y}$ contains the output variables at the kth time step, $u_k \in R^{n_u}$ contains the manipulated variables at the kth time step, $d_{m,k} \in R^{n_d}$ contains the measured disturbance variables at the kth time step, $x_k \in R^{n_x}$ contains the state variables at the kth time step, $e_k \in R^{n_e}$ denotes the feedback vector which contains the estimated unmeasured disturbances, $\hat{x}_0 \in R^{n_x}$, $\hat{e}_0 \in R^{n_e}$ denote the estimated states and unmeasured disturbances at the current time step, $u_{\max,k}, u_{\min,k} \in R^{n_u}$ are the upper and lower bounds on the manipulated variables at the kth time step, $y_{\max,k}, y_{\min,k} \in R^{n_y}$ are the upper and lower bounds on the controlled variables at the kth time step, $s_k \in R^{n_y}$ denotes the slack variables of the controlled variables y_k in the bounds. $y_{sp} \in R^{n_y}$ denotes the desired values of controlled variables, which are usually called the set points of the control system; $u_{sp} \in R^{n_u}$ denotes the desired values of manipulated variables, and we call these the set points of the manipulated variables in this thesis. $Q \in R^{n_y \times n_y}$ is the weighting matrix for controlled variables, $R \in R^{n_u \times n_u}$ is the weighting matrix for manipulated variables, $W \in R^{n_u \times n_u}$ is the weighting matrix (or move suppression matrix) on the change of the manipulated variables and $W_s \in R^{n_y \times n_y}$ is the weighting matrix of the slack variables.

The mathematical program (3.1a-3.1j) is a Quadratic Program (QP, Boyd and Vandenberghe, 2004). The objective function (3.1a) includes the distance between the predicted controlled and manipulated variables and their desired values, the penalty on the changes of manipulated variables and the slack variables that penalize violations of soft bounds. The process model (3.1b-3.1c) describes the open-loop behavior of the system, where we assume an unmeasured disturbance e affects the system states through B_e and in this thesis, it is assumed to be constant throughout the prediction horizon by equation (3.1d). Equation (3.1i) enforces the limited control horizon; the manipulated variables do not change after the control horizon. Equation (3.1e) denotes the hard bounds on the manipulated variables, which can never be violated in the physical system. Equations (3.1f-3.1g) describe the soft bounds on the controlled variables, which may be violated in the solution but their violation is penalized in the objective function. The benefit of using the soft bounds for the controlled variables is that it makes the mathematical program feasible even when the physical system is infeasible (i.e., when some controlled variables have to violate the bounds). Therefore, the controller will continue to function when faced with a violation that cannot be avoided due to, for example, a large disturbance and feedback dead time.

Note that:

1) The weighing matrices Q, R, W, W_s are properly tuned such that the QP problem is strictly convex (the Hessian of the objective function is positive definite). Then, the NMPC problem involves convex optimization in which the objective function value of a local optimum is ensured of being the value of the global optimum.
2) The unmeasured disturbance model described by equations (3.1b) and (3.1d) is from a more general framework introduced by Muske and Badgwell (2002), where the unmeasured disturbances can be deemed as the additional system states. For offset-free control, the unmeasured disturbance model should be designed such that the augmented system, equations (3.1b-3.1d), is detectable (Muske and Badgwell, 2002). We will check this for the unmeasured disturbance model used in all the case studies in the thesis. Then the system states x and the unmeasured disturbance states e can be estimated at the beginning of each time step by

$$\hat{x}_0 = \hat{x}_{0/-1} + L_x(y_{m,0} - C\hat{x}_{0/-1}) \tag{3.2}$$

$$\hat{e}_0 = \hat{e}_{0/-1} + L_e(y_{m,0} - C\hat{x}_{0/-1}) \tag{3.3}$$

where $y_{m,0} \in R^{n_y}$ denotes the measurements of the outputs at the current time step, $\hat{x}_{0/-1} \in R^{n_x}, \hat{e}_{0/-1} \in R^{n_e}$ denote the states and unmeasured disturbances at the current time step that is estimated at the last time step, $\hat{x}_0 \in R^{n_x}$, $\hat{e}_0 \in R^{n_e}$ denote the states and unmeasured disturbances at the current time step that is also estimated at the current time step, $L_x \in R^{n_x \times n_y}$, $L_e \in R^{n_e \times n_y}$ denote the discrete steady-state Kalman filter (Kalman, 1960) gains for x and e respectively. See Appendix A for more detail on output feedback and linear state estimation used in deriving the unmeasured disturbance model and the equations (3.2) and (3.3).

3) The open-loop optimization problem (3.1a-3.1j), in the context of nominal MPC with infinite (or sufficiently large) horizon, is equivalent to a closed-loop optimization problem that considers the effect of future control actions on the system behavior (Mayne et al., 2000), because it is assumed that no mismatch exists between the process and the controller model. However, solving the single, open-loop optimization does not optimize the closed-loop trajectories when the uncertainties are considered in the formulation explicitly (see the next section for details).
4) If there is time-delay between a manipulated variable and a controlled variable (or state variable) we write the process model in the canonical form (3.1b-3.1c) by introducing additional states to the system. See Appendix B for more detail.

The open-loop optimization problem (3.1) can also be written in the following form using the extended vectors:

$$\begin{aligned}\min_{\mathbf{u},\mathbf{s}}\ & (\mathbf{y}-\mathbf{y}_{sp})^T \widetilde{Q}(\mathbf{y}-\mathbf{y}_{sp}) + (\mathbf{u}-\mathbf{u}_{sp})^T \widetilde{R}(\mathbf{u}-\mathbf{u}_{sp}) \\ & + (I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1})^T \widetilde{W}(I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1}) + \mathbf{s}^T \widetilde{W}_s \mathbf{s}\end{aligned} \tag{3.4a}$$

$$\text{s.t.}\quad \mathbf{x} = \widetilde{A}_x \hat{x}_0 + \widetilde{A}_e \hat{e}_0 + \widetilde{B}\mathbf{u} + \widetilde{B}_d \mathbf{d}_m \tag{3.4b}$$

$$\mathbf{y} = \widetilde{C}\mathbf{x} \tag{3.4c}$$

$$\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max} \tag{3.4d}$$

$$\mathbf{y}_{\min} - \mathbf{s} \le \mathbf{y} \le \mathbf{y}_{\max} + \mathbf{s} \tag{3.4e}$$

$$\mathbf{s} \ge 0 \tag{3.4f}$$

where the bold symbols denote the extended vectors that contain the related variables in the control or prediction horizon, specifically,

$$\mathbf{x}=\begin{pmatrix}x_1\\x_2\\\vdots\\x_p\end{pmatrix}\in R^{n_x p},\ \mathbf{u}=\begin{pmatrix}u_0\\u_1\\\vdots\\u_{n-1}\end{pmatrix}\in R^{n_u n},\ \mathbf{d}_m=\begin{pmatrix}d_{m,0}\\d_{m,1}\\\vdots\\d_{p-1}\end{pmatrix}\in R^{n_d p},\ \mathbf{y}=\begin{pmatrix}y_1\\y_2\\\vdots\\y_p\end{pmatrix}\in R^{n_y p},$$

$$\mathbf{s}=\begin{pmatrix}s_1\\s_2\\\vdots\\s_p\end{pmatrix}\in R^{n_y p},\ \mathbf{y}_{sp}=\begin{pmatrix}y_{sp}\\y_{sp}\\\vdots\\y_{sp}\end{pmatrix}\in R^{n_y p},\ \mathbf{u}_{sp}=\begin{pmatrix}u_{sp}\\u_{sp}\\\vdots\\u_{sp}\end{pmatrix}\in R^{n_u n},\ \mathbf{y}_{\max}=\begin{pmatrix}y_{\max}\\y_{\max}\\\vdots\\y_{\max}\end{pmatrix}\in R^{n_y p},$$

$$\mathbf{y}_{\min}=\begin{pmatrix}y_{\min}\\y_{\min}\\\vdots\\y_{\min}\end{pmatrix}\in R^{n_y p},\ \mathbf{u}_{\max}=\begin{pmatrix}u_{\max}\\u_{\max}\\\vdots\\u_{\max}\end{pmatrix}\in R^{n_u n},\ \mathbf{u}_{\min}=\begin{pmatrix}u_{\min}\\u_{\min}\\\vdots\\u_{\min}\end{pmatrix}\in R^{n_u n},$$

the weighing matrices in the objective function

$$\tilde{Q}=\begin{pmatrix}Q&&\\&\ddots&\\&&Q\end{pmatrix}\in R^{(n_y p)\times(n_y p)},\ \tilde{R}=\begin{pmatrix}R&&\\&\ddots&\\&&R\end{pmatrix}\in R^{(n_u n)\times(n_u n)},$$

$$\tilde{W}=\begin{pmatrix}W&&\\&\ddots&\\&&W\end{pmatrix}\in R^{(n_u n)\times(n_u n)},\ \tilde{W}_s=\begin{pmatrix}W_s&&\\&\ddots&\\&&W_s\end{pmatrix}\in R^{(n_y p)\times(n_y p)},$$

and the coefficient of the open-loop model

$$\widetilde{A}_x = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^p \end{pmatrix} \in R^{(n_x p)\times n_x}, \quad \widetilde{A}_e = \begin{pmatrix} B_e \\ AB_e + B_e \\ \vdots \\ (\sum_{i=0}^{p-1} A^i)B_e \end{pmatrix} \in R^{(n_x p)\times n_u},$$

$$\widetilde{B} = \begin{pmatrix} B & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ A^{n-1}B & A^{n-2}B & \cdots & B \\ A^n B & A^{n-1}B & \cdots & AB+B \\ \vdots & \vdots & \vdots & \vdots \\ A^{p-1}B & A^{p-2}B & \cdots & (\sum_{i=0}^{p-n} A^i)B \end{pmatrix} \in R^{(n_x p)\times(n_u n)},$$

$$\widetilde{B}_d = \begin{pmatrix} B_d & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ A^{n-1}B_d & A^{n-2}B_d & \cdots & 0 \\ A^n B_d & A^{n-1}B_d & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{p-1}B_d & A^{p-2}B_d & \cdots & B_d \end{pmatrix} \in R^{(n_x p)\times(n_d p)}, \quad \widetilde{C} = \begin{pmatrix} C & & \\ & \ddots & \\ & & C \end{pmatrix} \in R^{(n_y p)\times(n_x p)}.$$

In addition,

$$I_{\Delta 1} = \begin{pmatrix} I & & & \\ -I & I & & \\ & & \ddots & \\ & & -I & I \end{pmatrix} \in R^{(n_u n)\times(n_u n)} \text{ and } I_{\Delta 2} = \begin{pmatrix} I \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in R^{(n_u n)\times n_u}$$

are the matrices such that $\Delta\mathbf{u} = (I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1})$ and $\Delta\mathbf{u} = \left(\Delta u_0^T, \cdots, \Delta u_{n-1}^T\right)^T \in R^{n_u n}$.

More discussion on MPC can be found in Maciejowski (2002) and its industrial applications in Qin and Badgwell (2003). Nominal MPC has proved extremely successful and is widely applied, but it cannot guarantee feasibility for uncertain processes. Thus, the Robust MPC is presented in the next section.

3.2 The New Robust MPC Formulation

3.2.1 Robust MPC with closed-loop model

In this research, we develop the robust MPC formulation based on the nominal MPC formulation (3.1) in the last section. We first discuss a simple, but incorrect, formulation that provides insight into the correct formulation. Then, the correct formulation is presented. We conclude that the correct formulation is not computationally tractable, and modifications are presented in subsequent sub-sections to achieve a controller algorithm that is both theoretically sound and tractable.

A straightforward way to formulate the robust MPC problem (i.e. to explicitly address uncertainty in the system) is to replace the nominal process model (3.1b-3.1c) with an uncertain process model in the open-loop optimization framework in problem (3.1). A scenario method (e.g., Sen and Higle, 1999) for introducing the uncertain model would be to use a number of uncertain models, in which the optimal manipulations are implemented, and require all models to predict feasible solutions. (Naturally, even a large but finite number of models will not guarantee feasibility for parametric uncertainty that is continuous, but they could provide an adequate approximation.) The resulting formulation obtains a series of deterministic manipulated variables such that the nominal performance is the optimal and different realizations of controlled variables are kept within (or driven close to) the feasible region. Figure 3.2 shows the prediction of a sample single input-single output (SISO) system (the first CSTR system studied in Section 3.6) using a set of uncertain models and this open-loop formulation for uncertainty in MPC.

We can see that while the controlled variables are different for different realizations of the process, the manipulated variables are the same for different realizations of the process. This open-loop prediction is not correct because the feedback controller will respond differently for different realizations of the process because the measurements of y will be different. As a result, the manipulated variables will be different after the first controller execution (or for the dead time plus one) for every realization. Therefore, we conclude that a correct prediction of the future system behavior, and a correct controller formulation, has to include the effect of the control law. The dashed dotted lines show the predicted boundaries of the open-loop uncertainty, which are the maximum and minimum controlled or manipulated variables under this uncertainty with 99% confidence. The method for obtaining these boundaries is explained in the subsection 3.3.1.

Before developing the improved controller formulation, we will observe the same CSTR SISO system with feedback measurements and controller execution at a period of 0.30 minute. Several realizations from the parameter distribution have been selected, and the process with each realization of the process parameters is controlled by a nominal MPC. Figure 3.3 shows the transient results. We can see that both the controlled variable and the manipulated variable vary with the realization of parameter values, so that we say that they are uncertain in the prediction horizon. We use dashed lines to show the predicted boundaries of the closed-loop uncertainty, which are the maximum and minimum controlled or manipulated variables under this uncertainty with 99% confidence. The closed-loop uncertainty region is smaller than the open-loop uncertainty region for the controlled variable, and the controlled variable returns to its set point value in steady state with nearly zero uncertainty, while the manipulated variable final value is uncertain. In general, the open-loop prediction of uncertainty is bigger than the closed-loop prediction of uncertainty (with a properly tuned controller), and in this case the robust MPC with open-loop model gives a more (overly) conservative control.

Figure 3.4 shows the special case in which the manipulated variable saturates throughout the horizon. In this case, the open-loop uncertainty equals to closed-loop uncertainty because the controller action does not influence the manipulated variable for all realizations. In other words, the feedback information does not change the controller action at all, and the closed-loop system behaves as though no control existed. Thus, we conclude that the effect of the model uncertainty depends on the scenario occurring, which must be included in the robust MPC controller design.

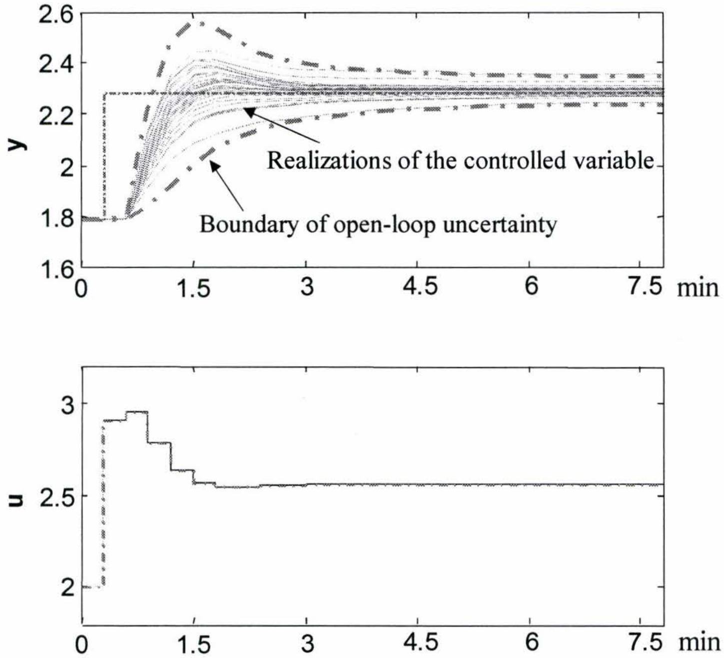


Figure 3.2 An open-loop prediction of the uncertain CSTR system in MPC control

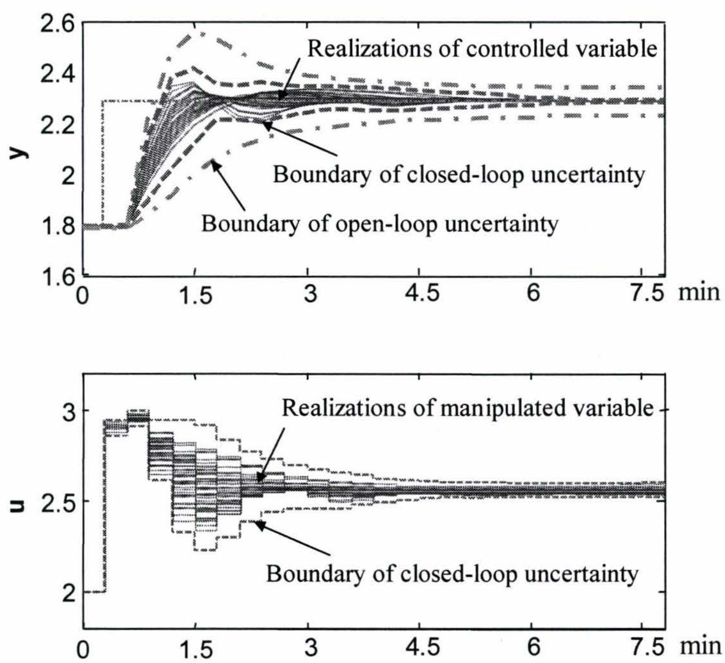


Figure 3.3 NMPC control of numerous realizations of the uncertain CSTR system (No saturation of the manipulated variable)

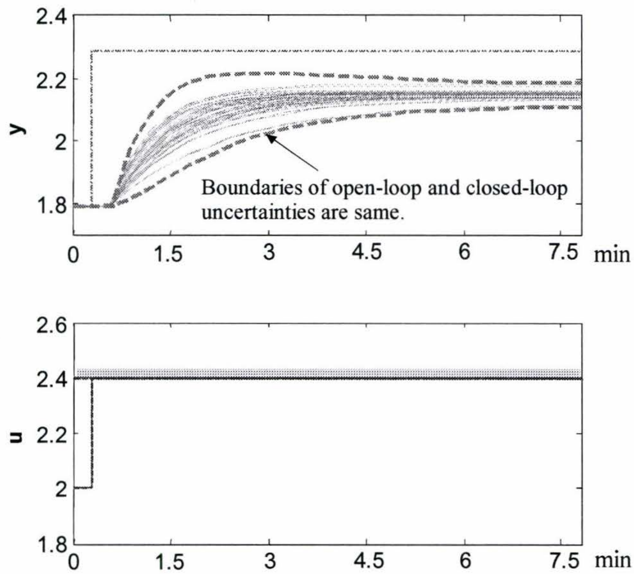


Figure 3.4 NMPC control of numerous realizations of the uncertain CSTR system (Saturation of the manipulated variable)

According to the above discussion, a good robust MPC method has to use the closed-loop model, i.e., a model of the (uncertain) process and the controller, to predict the future system behavior. This means that the effects of the future controller actions on the system behavior have to be modelled. In contrast to the nominal MPC, which has only a model for the process at each future time step, the robust MPC requires models for the process and the controller at every time step, and the future controllers would be the robust formulation. While this formulation would be correct, the resulting mathematical problem would be too complex for real-time computation. In this research, the nominal MPC controller, i.e., formulation (3.1), is adopted as the controller model that will be included at every future time step in the robust formulation. The performance with this approximation will be shown to provide good performance through numerous case studies in this thesis.

Therefore, a robust MPC formulation with closed-loop model, called RMPC-CL, can be built as shown in the following

RMPC-CL:

$$\min_{\hat{\mathbf{y}}_{sp,k+1},\hat{\mathbf{u}}_{sp,k},\mathbf{s}} \sum_{k=0}^{p-1}(y_{k+1}-y_{sp})^T Q(y_{k+1}-y_{sp})+\sum_{k=0}^{n-1}(u_k-u_{sp})^T R(u_k-u_{sp}) + \sum_{k=0}^{n-1}\Delta u_k{}^T W \Delta u_k + \sum_{k=0}^{p-1} s_{k+1}{}^T W_s s_{k+1} \tag{3.5a}$$

s.t.

$$x_{r,k+1} = A_{r,k+1}x_{r,k} + B_{r,k+1}u_{r,k} + B_{er,k+1}e_{r,k} + B_{dr,k+1}d_{m,k} + G_{wx}w_k \tag{3.5b}$$

$$e_{r,k+1} = e_{r,k} + G_{we}w_k \tag{3.5c}$$

$$y_{r,k+1} = C_{r,k+1}x_{r,k+1} \tag{3.5d}$$

$$y_{r,m,k+1} = C_{r,k+1}x_{r,k+1} + v_{k+1} \tag{3.5e}$$

$$(\hat{x}_{r,k+1},\hat{e}_{r,k+1}) = SE(\hat{x}_{r,k},\hat{e}_{r,k},d_{m.k},y_{m,k+1}) \tag{3.5f}$$

$$u_{\min,k} \le u_{r,k} \le u_{\max,k} \tag{3.5g}$$

$$y_{\min,k+1} - s_{k+1} \le y_{r,k+1} \le y_{\max,k+1} + s_{k+1} \tag{3.5h}$$

$$s_{k+1} \ge 0 \tag{3.5i}$$

(3.5b)–(3.5i): $k = 0,\cdots,p-1$

$$u_{r,k} = NMPC(\hat{x}_{r,k},\hat{e}_{r,k},\mathbf{d}_{m,k},\hat{\mathbf{y}}_{sp,k+1},\hat{\mathbf{u}}_{sp,k}) \tag{3.5j}$$

$$\Delta u_k = u_k - u_{k-1} \tag{3.5k}$$

(3.5j)–(3.5k): $k = 0,\cdots,n-1$

$$u_{r,k} = u_{r,k-1} \qquad k = n,\cdots,p-1 \tag{3.5l}$$

$$x_{r,0} = \hat{x}_0 \tag{3.5m}$$

$$e_{r,0} = \hat{e}_0 \tag{3.5n}$$

For all $A_{r,k+1}$, $B_{r,k+1}$, $B_{dr,k+1}$, $C_{r,k+1}$, w_k, v_{k+1} in uncertainty region, $k = 0,\ldots, n-1$

where

1) The objective of the optimization is still to minimize the nominal performance equation (3.5a), and the degrees of freedom are the values of controlled and manipulated variable set points in the future horizon $\hat{\mathbf{y}}_{sp,k+1} \in R^{n_y p}$, $\hat{\mathbf{u}}_{sp,k} \in R^{n_u n}$.
2) Equations (3.5b-3.5e) are the process models containing uncertain parameters. $A_{r,k+1}, B_{r,k+1}, B_{er,k+1}, B_{dr,k+1}, C_{r,k+1}$ denote all realizations ($r \in \Phi_r$) of the uncertain parameters of the model at time step k+1 within defined uncertainty region. These parameters could be either time-invariant or time-varying. $w_k \in R^{n_w}$ denotes the unmeasured disturbances and noise that affect the states of the system (through G_{wx}, G_{we}). $x_{r,k} \in R^{n_x}$, $e_{r,k} \in R^{n_e}$, $y_{r,k} \in R^{n_y}$ denote the uncertain values of state variables, feedback variables and controlled variables at the kth time step, which depend on the realizations of the process. $y_{rm,k+1} \in R^{n_y}$ denotes the measurements of the controlled variables at the $(k+1)^{\text{th}}$ time step, and $v_{k+1} \in R^{n_y}$ denotes the noise in the measurement.
3) Equation (3.5f) denotes the state estimation using linear steady-state Kalman filter (details in Appendix A).
4) Equation (3.5g) denotes the hard bounds on the uncertain manipulated variables in the future. Equations (3.5h-3.5i) denote the soft bounds on the uncertain controlled variables in the future.
5) Equations (3.5j-3.5l) denote the nominal MPC control that determines the manipulated variables at the future kth time step in the control horizon (with the estimates $\hat{x}_{r,k} \in R^{n_x}$, $\hat{e}_{r,k} \in R^{n_e}$, the predicted disturbances $\mathbf{d}_{m,k} \in R^{n_d p}$ and the set points $\hat{\mathbf{y}}_{sp,k+1}$, $\hat{\mathbf{u}}_{sp,k}$). The nominal MPC control law $NMPC(\hat{x}_{r,k}, \hat{e}_{r,k}, \mathbf{d}_{m,k}, \hat{\mathbf{y}}_{sp,k+1}, \hat{\mathbf{u}}_{sp,k})$ is from the solution of the QP formulation (3.1). Here, we do not consider the bounds on controlled variables (equations (3.1f-3.1g)) in this control law because they can be enforced by the soft bounds in the outer problem (equations (3.5h-3.5i)). Since the state estimates in the future depend on the realizations of the process, the manipulated variables determined in the future also depend on the realizations of the process. Therefore, we use the new symbol $u_{r,k} \in R^{n_u}$ to represent the uncertain

manipulated variables in the future *k*th time step so as to differentiate them from the nominal manipulated variables u_k.

6) Equations (3.5m-3.5n) approximate the current state and feedback variables with their nominally estimated values respectively.

There are basically two ways to characterize the parametric process model uncertainty. One way is to use the representative sampled values of the uncertain parameters. In this way the robust MPC formulation (3.5) would be a convex QP problem, and the problem is essentially a multi-stage stochastic program where the number of time steps in the input horizon is the number of decision stages. In this case, however, the problem size would increase exponentially with the increase of the number of realizations, so this approach would suffer the curse of dimensionality even for small-scale problems (Kleywegt and Shapiro, 2001). For example, a robust MPC problem with 2 manipulated variables, 2 controlled variables, 10 time steps control horizon and 10 plant realizations involves $n_{eq}=4\times 10^{10}$ variables. A linear equation system with the same number of variables and equations is easier than this robust MPC problem and it requires more than $2(n_{eq})^3/3$ floating-point operations (Golub and Van Loan, 1996). Solving this equation system with the IBM Roadrunner supercomputer (IBM, 2008) (the fastest computer of the world until 2008 with the computing power of about 10^{15} floating-point operations per second) would take over 1.3×10^9 years!

The other way to model the uncertainty is to use the continuous distribution or continuous uncertainty region of the uncertain parameters. We adopt this method to avoid the curse of dimensionality (Kleywegt and Shapiro, 2001). However, the bilevel stochastic optimization problem with continuous uncertainty is typically very difficult to solve in the real-time (Colson et al., 2007). Therefore, in the next several sections we will discuss our reformation and approximations of this difficult bilevel problem that yield tractable computation for real-time applications.

3.2.2 The reformulation to single level problem

Due to the challenges in solving a bilevel problem, we transform the bilevel problem (3.5) into a single-level problem by replacing the inner optimization problems (equation (3.5b) for nominal MPC calculations with their optimality conditions (Clark and Westerberg, 1983).

For convenience of the description, we use the matrix-vector form of nominal MPC (formulation (3.4) without soft bounds on the controlled variables) as the inner optimization problem. If the inner nominal MPC is properly tuned (typically it has the same weighting matrices as the outer problem), the QP problem will be strictly convex so that it's optimum can be determined through its first order Karush-Kuhn-Tucker (KKT) conditions (Nocedal and Wright, 1999). Therefore, the QP is equivalent to the following equations

$$\begin{aligned}&2\widetilde{B}^T\widetilde{C}^T\widetilde{Q}(\widetilde{C}\widetilde{A}_x\hat{x}_0+\widetilde{C}\widetilde{B}_d\mathbf{d}_m+\widetilde{C}\widetilde{A}_e\hat{e}_0-\mathbf{y}_{sp})+2(\widetilde{B}^T\widetilde{C}^T\widetilde{Q}\widetilde{C}\widetilde{B}+\widetilde{R}+I_{\Delta1}^T\widetilde{W}I_{\Delta1})\mathbf{u}\\&-2\widetilde{R}\mathbf{u}_{sp}-2I_{\Delta1}^T\widetilde{W}I_{\Delta2}u_{-1}+\boldsymbol{\lambda}^+-\boldsymbol{\lambda}^-=0\end{aligned}\tag{3.6a}$$

$$\boldsymbol{\lambda}^+\cdot(\mathbf{u}-\mathbf{u}_{\max})=0\,,\quad\boldsymbol{\lambda}^-\cdot(-\mathbf{u}+\mathbf{u}_{\min})=0\,,\quad\boldsymbol{\lambda}^+,\boldsymbol{\lambda}^{-1}\geq0\tag{3.6b}$$

$$\mathbf{u}_{\min}\leq\mathbf{u}\leq\mathbf{u}_{\max}\tag{3.6c}$$

Here equation (3.6a) is the stationary condition, and equation (3.6b) denotes the complementarity constraints that describe the active bounds on the manipulated variables. $\boldsymbol{\lambda}^+$ and $\boldsymbol{\lambda}^-$ denote the Lagrange multipliers for the constraints $\mathbf{u}\leq\mathbf{u}_{\max}$ and $\mathbf{u}_{\min}\leq\mathbf{u}$ respectively, where $\boldsymbol{\lambda}^+=\left((\lambda_0^+)^T,\cdots,(\lambda_{n-1}^+)^T\right)\in R^{n_u n}$ and $\lambda_k^+=\left(\lambda_{1,k}^+,\cdots,\lambda_{n_u,k}^+\right)^T\in R^{n_u}$, $\boldsymbol{\lambda}^-=\left((\lambda_0^-)^T,\cdots,(\lambda_{n-1}^-)^T\right)\in R^{n_u n}$ and $\lambda_k^-=\left(\lambda_{1,k}^-,\cdots,\lambda_{n_u,k}^-\right)^T\in R^{n_u}$, the dot "$\cdot$" denotes the element-wise multiplication.

From Equation (3.6a) we have

$$\begin{aligned}\mathbf{u}=&(\widetilde{B}^T\widetilde{C}^T\widetilde{Q}\widetilde{C}\widetilde{B}+\widetilde{R}+I_{\Delta1}^T\widetilde{W}I_{\Delta1})^{-1}(\widetilde{B}^T\widetilde{C}^T\widetilde{Q}(\mathbf{y}_{sp}-\widetilde{C}\widetilde{A}_x\hat{x}_0-\widetilde{C}\widetilde{B}_d\mathbf{d}_m-\widetilde{C}\widetilde{A}_e\hat{e}_0)\\&+\widetilde{R}\mathbf{u}_{sp}+I_{\Delta1}^T\widetilde{W}I_{\Delta2}u_{-1}-(\boldsymbol{\lambda}^+-\boldsymbol{\lambda}^-)/2)\end{aligned}\tag{3.7}$$

Since nominal MPC only implements the control actions at the solution time, we only require the part of Equation (3.7) that provides u_0,

$$\begin{aligned} u_0 = I_{pu}(\widetilde{B}^T\widetilde{C}^T\widetilde{Q}\widetilde{C}\widetilde{B} + \widetilde{R} + I_{\Delta 1}^T\widetilde{W}I_{\Delta 1})^{-1}(\widetilde{B}^T\widetilde{C}^T\widetilde{Q}(\mathbf{y}_{sp} - \widetilde{C}\widetilde{A}_x\hat{x}_0 - \widetilde{C}\widetilde{B}_d\mathbf{d}_m - \widetilde{C}\widetilde{A}_e\hat{e}_0) \\ + \widetilde{R}\mathbf{u}_{sp} + I_{\Delta 1}^T\widetilde{W}I_{\Delta 2}u_{-1} - (\boldsymbol{\lambda}^+ - \boldsymbol{\lambda}^-)/2) \end{aligned} \quad (3.8)$$

where I_{pu} is a "pick-up" matrix (containing only 1's and 0's) to extract the values of the manipulated variables at the first time step in the solution.

Since they are equivalent, we can replace the nominal MPC control law $NMPC(\hat{x}_k, \hat{e}_k, \mathbf{d}_{m,k}, \hat{\mathbf{y}}_{sp,k+1}, \hat{\mathbf{u}}_{sp,k})$ with equations (3.6b-3.6c) and (3.8), so that the bilevel stochastic problem (3.5) becomes single level Mathematical Program with Equilibrium Constraints (MPEC) (Luo et al., 1996). However, this single level stochastic problem is still difficult to solve in the real-time because:

1) The complementarity constraints (3.6b) are highly nonconvex. Also their Jacobian could be singular at the solution, which will cause numerical problems for convergence.
2) There is no systematic and efficient method to solve the complementarity constraints with uncertain parameters characterized by continuous uncertainty region.

3.2.3 The reformulation with known saturated manipulated variables

If we knew the "saturation pattern", i.e., which bounds are active at which time steps, before we solve the problem, we could avoid including the Lagrange multipliers and the complementarity constraints in the formulation. We will assume that the active inequality constraints are known and develop a simplified solution for the nominal controller; then, we will explain the method used to determine the active set.

Here we assume the following.

Assumption 3.1: *A manipulated variable at a time step either equals its bound for all the realizations of the process or is unconstrained for all the realizations of the process.*

In practice, this means that if the manipulated variable is active for the "most extreme" value in its uncertainty region, the manipulated variable at that time step of the solution to the nominal MPC is set active. The remaining part of this section gives the details of replacing the complementarity constraints and Lagrange multipliers in the nominal MPC control law (3.6b-3.6c) and (3.8) with selected equality constraints modeling a known saturation status of the manipulated variables.

When no bounds are active, all the Lagrange multipliers must be zero (because of the complementarity constraints (3.6b)), and equation (3.8) becomes

$$\begin{aligned} u_0 = I_{pu}(\tilde{B}^T\tilde{C}^T\tilde{Q}\tilde{C}\tilde{B} + \tilde{R} + I_{\Delta1}^T\tilde{W}I_{\Delta1})^{-1}(\tilde{B}^T\tilde{C}^T\tilde{Q}(\mathbf{y}_{sp} - \tilde{C}\tilde{A}_x\hat{x}_0 - \tilde{C}\tilde{B}_d\mathbf{d}_m - \tilde{C}\tilde{A}_e\hat{e}_0) \\ + \tilde{R}\mathbf{u}_{sp} + I_{\Delta1}^T\tilde{W}I_{\Delta2}u_{-1}) \end{aligned} \tag{3.9}$$

We note that equation (3.9) has $n_y \times p + n_u \times n$ degrees of freedom (set points for all controlled variables and manipulated variables) while the physical system has only $n_u \times n$ degrees of freedom (all manipulated variables). This situation will yield ill-conditioned problems with alternative solutions. We apply only the manipulated variable values, not the set points, in the solution of the robust MPC. Therefore, we define a new vector of variables, **t**, which is a linear combination of the set points of the controlled variables and manipulated variables targets and will be the variables adjusted to optimize the robust MPC:

$$t = I_{pu} \cdot (\tilde{B}^T\tilde{C}^T\tilde{Q}\tilde{C}\tilde{B} + \tilde{R} + I_{\Delta1}^T\tilde{W}I_{\Delta1})^{-1}(\tilde{B}^T\tilde{C}^T\tilde{Q}\mathbf{y}_{sp} + \tilde{R}\mathbf{u}_{sp}) \tag{3.10}$$

and write equation (3.9) as

$$u_0 = K_x\hat{x}_0 + K_e\hat{e}_0 + K_u u_{-1} + K_d\mathbf{d}_m + t \tag{3.11}$$

where

$K_x = -I_{pu} \cdot (\tilde{B}^T\tilde{C}^T\tilde{Q}\tilde{C}\tilde{B} + \tilde{R} + I_{\Delta1}^T\tilde{W}I_{\Delta1})^{-1}\tilde{B}^T\tilde{C}^T\tilde{Q}\tilde{C}\tilde{A}_x$,

$K_e = -I_{pu} \cdot (\tilde{B}^T\tilde{C}^T\tilde{Q}\tilde{C}\tilde{B} + \tilde{R} + I_{\Delta1}^T\tilde{W}I_{\Delta1})^{-1}\tilde{B}^T\tilde{C}^T\tilde{Q}\tilde{C}\tilde{A}_e$,

$K_u = I_{pu} \cdot (\widetilde{B}^T \widetilde{C}^T \widetilde{Q} \widetilde{C} \widetilde{B} + \widetilde{R} + I_{\Delta 1}^T \widetilde{W} I_{\Delta 1})^{-1} F_1^T \widetilde{W} F_2$,

$K_d = -I_{pu} \cdot (\widetilde{B}^T \widetilde{C}^T \widetilde{Q} \widetilde{C} \widetilde{B} + \widetilde{R} + I_{\Delta 1}^T \widetilde{W} I_{\Delta 1})^{-1} \widetilde{B}^T \widetilde{C}^T \widetilde{Q} \widetilde{C} \widetilde{B}_d$.

Note that the nominal MPC can be properly tuned so that the weighting matrix $\widetilde{R}$ has full rank. In this case, for any value of t there will be values of $\mathbf{y}_{sp}$, $\mathbf{u}_{sp}$ that give the same value of t through Equation (3.10); or for any value of $\mathbf{y}_{sp}$, $\mathbf{u}_{sp}$ there will be unique value of t corresponds to it through equation (3.10). Equation (3.11) denotes the unconstrained nominal MPC control law.

When a manipulated variable (an element in u_0) is active, the corresponding non-zero Lagrange multiplier forces the manipulated variable to its bound through the complementarity relationship and the multiplier value does not affect other manipulated variables. Therefore, the Lagrange multipliers can also been omitted in the formulation if we know the active set and enforce them via linear equations. We can address the saturation of the manipulated variables through extending equation (3.11) by adding the active constraints as equations.

$$u_0 = I_\delta (K_x \hat{x}_0 + K_e \hat{e}_0 + K_u u_{-1} + K_d \mathbf{d}_m) + t \tag{3.12}$$

$$(I - I_\delta) t = u_c \tag{3.13}$$

where $I_\delta \in R^{n_u \times n_u}$ is a diagonal matrix with the diagonal elements containing 0 or 1 to specify the saturation, $I \in R^{n_u \times n_u}$ is an identity matrix. The vector u_c contains the active upper bound or lower bound, which is known when we know the saturation pattern. Using this formulation, the controller can be modeled for any known active set. If an element in u_0 is active, the corresponding element in I_δ is 0, i.e., we make the element in u_0 equal to the corresponding element in t. The result is a solution that ensures the predefined active set is achieved, and the remaining manipulated variable values are calculated based on the MPC optimization.

The resulting MPC model can be substituted into the inner optimization problem *NMPC*($\hat{x}_k$, $\hat{e}_k$, $\mathbf{d}_{m,k}$, $\hat{y}_{sp,k+1}$, $\hat{u}_{sp,k}$) in the robust MPC formulation RMPC-CL (formulation (3.5)) with the control law (3.12) and (3.13). The problem contains a convex quadratic objective function, a series of linear equations and bounds, which can be summarized as the following problem RMPC-CLT:

RMPC-CLT:

$$\min_{\mathbf{t},\mathbf{s}} \ (\mathbf{y}-\mathbf{y}_{sp})^T \widetilde{Q}(\mathbf{y}-\mathbf{y}_{sp}) + (\mathbf{u}-\mathbf{u}_{sp})^T \widetilde{R}(\mathbf{u}-\mathbf{u}_{sp}) + (I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1})^T \widetilde{W}(I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1}) + \mathbf{s}^T \widetilde{W}_s \mathbf{s} \tag{3.14a}$$

$$\text{s.t.} \quad \mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \tag{3.14b}$$

$$\mathbf{y}_r = L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_y\boldsymbol{\omega} \tag{3.14c}$$

$$(\mathbf{I} - \mathbf{I}_{\boldsymbol{\delta}})\mathbf{t} = \mathbf{u}_c \tag{3.14d}$$

$$\mathbf{u}_{\min} \le \mathbf{u}_r \le \mathbf{u}_{\max} \tag{3.14e}$$

$$\mathbf{y}_{\min} - \mathbf{s} \le \mathbf{y}_r \le \mathbf{y}_{\max} + \mathbf{s} \tag{3.14f}$$

$$\mathbf{s} \ge 0 \tag{3.14g}$$

For all L_{ur}, L_{yr}, M_{ur}, M_{yr}, $\boldsymbol{\omega}$ in the uncertainty region and the pre-determined $\mathbf{I}_{\boldsymbol{\delta}}$ and $\mathbf{u}_c$

where

1) The objective function of the problem is still the nominal dynamic performance of the uncertain system, but the degrees of freedom change from $\hat{\mathbf{y}}_{sp,k+1}$, $\hat{\mathbf{u}}_{sp,k}$ to $\mathbf{t} = \left(t_0^T, \cdots, t_{n-1}^T\right)^T \in R^{n_u n}$.

2) Equations (3.14b-3.14c) are the closed-loop model of the system with uncertain parameters L_{ur}, M_{ur}, L_{yr}, M_{yr} and certain parameters N_u, N_y (see Appendix D for details on deriving this model). The bold symbol $\mathbf{u}_r = \left(u_{r,0}^T, \cdots, u_{r,n-1}^T\right)^T$ denotes the extended vector containing the uncertain manipulated variables in the control horizon and $\mathbf{y}_r = \left(y_{r,1}^T, \cdots, y_{r,p}^T\right)^T$ denotes the extended vector containing the uncertain controlled

variables in the prediction horizon. $\boldsymbol{\theta} = \left(u_{-1}^T, \tilde{\mathbf{d}}_m^T, \hat{x}_0^T, \hat{e}_0^T\right)^T$ contains data that define the scenario, which are (1) the current manipulated variables, (2) the disturbance forecast, (3) initial states, and (4) the current feedback error. $\boldsymbol{\omega}$ contains the noise in the future.

3) Equation (3.14d) is used to force the saturated manipulated variables to their corresponding bounds, where $\mathbf{I}_{\boldsymbol{\delta}} \in R^{(n_u n)\times(n_u n)}$ denotes the diagonal matrix with the diagonal vector $\boldsymbol{\delta} = (\delta_1^T, \cdots, \delta_n^T)^T$ specifying the saturation patterns for the future n time steps in the control horizon, $\mathbf{u}_c = \left(u_{c,0}^T, \cdots, u_{c,n-1}^T\right)^T$ contains the corresponding active bounds, $\mathbf{I} \in R^{(n_u n)\times(n_u n)}$ is an Identity matrix. As noted previously, $\mathbf{I}_{\boldsymbol{\delta}}$ and $\mathbf{u}_c$ are given before we solve problem RMPC-CLT.

4) The uncertainties in the parameters L_{ur}, M_{ur}, L_{yr}, M_{yr} depend on the saturation of the manipulated variables (defined by $\mathbf{I}_{\boldsymbol{\delta}}$).

It's not difficult to find that:

Remark 3.1: *If Assumption 3.1 holds and we know the correct saturation pattern of the manipulated variables, the control law (3.12) is equivalent to the inner optimization problem NMPC*($\hat{x}_{r,k}, \hat{e}_{r,k}, \mathbf{d}_{m,k}, \hat{y}_{sp,k+1}, \hat{u}_{sp,k}$) *and the formulation RMPC-CLT is equivalent to the formulation RMPC-CL.*

Now the question is: How do we get a "reasonable" saturation pattern for problem RMPC-CLT?

3.2.4 The active set heuristic to obtain the active bounds

We will use a heuristic to obtain the active bounds on the manipulated variables in an iterative manner. The heuristic is given in the following steps.

1) Assume no bounds are active in the future, set all diagonal elements in $\mathbf{I}_{\boldsymbol{\delta}}$ to 1, and solve problem RMPC-CLT.
2) The solution of problem RMPC-CLT gives the uncertain trajectory of the manipulated variables in the control horizon. If some manipulated variables,

which are assumed unsaturated, have a value(s) at its bound (i.e. the boundaries of their uncertainty regions reach the upper or lower bounds on these manipulated variables), go to step (3); otherwise, end the iterative procedure and the current solution is the final solution.

3) Set all manipulated variables, which are at their bounds at the earliest time step, to their bound values (by specifying $\mathbf{I}_\delta$, $\mathbf{u}_c$). Solve problem RMPC-CLT again and go to step (2). (Any manipulated variable that has been set to its bound value will be constrained for the remainder of the RMPC-CLT solution).

Figure 3.5 illustrates the heuristic. At the first iteration, the problem is solved with the assumption that no bounds are active (in the inner problem). However, the solution

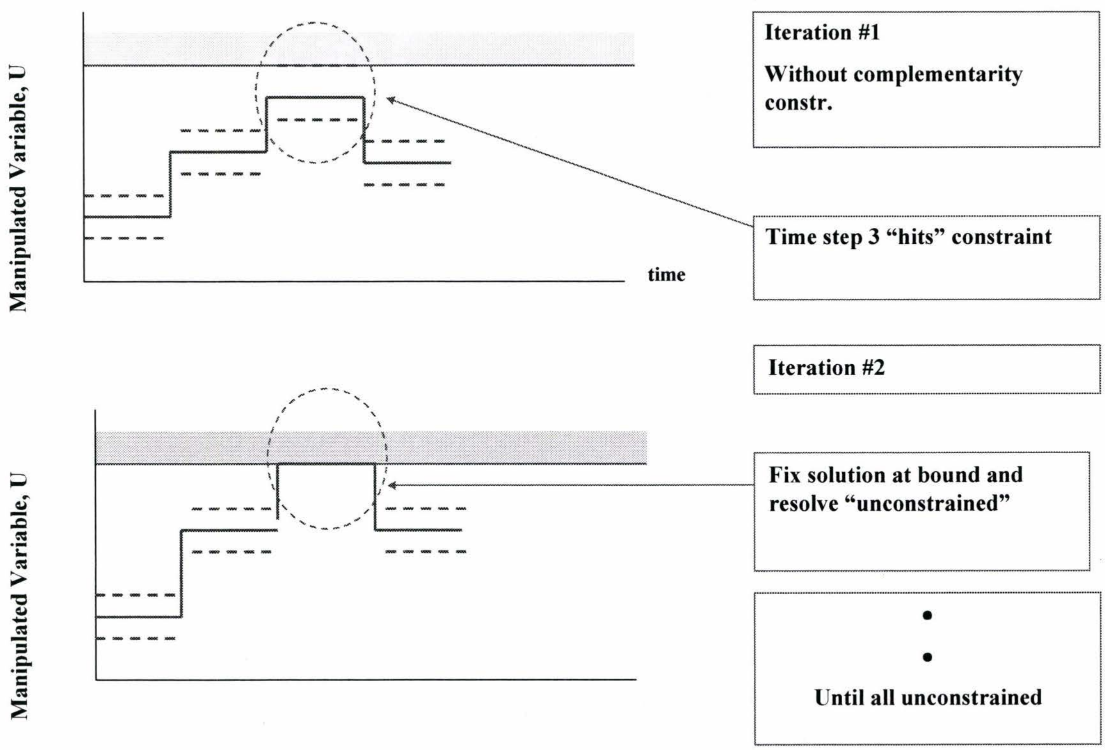


Figure 3.5 The active set heuristic to obtain the active bounds on manipulated variables

gives the uncertain trajectory of the manipulated variable whose boundary at the 3rd time step is active at its upper bound. Then, the manipulated variable at the 3rd time step is fixed to the upper bound. Then, the problem is solved again. The procedure is repeated until at each time step all manipulated variables are either fixed to a bound or are within their bounds.

Note that:

1) A similar heuristic has been successfully applied in industry for the constrained (nominal) MPC algorithm called Dynamic Matrix Control, DMC (Prett and Gillette, 1979), for more than 20 years. DMC is an industrial version of nominal MPC technique, and the heuristic addressed the hard bounds on the manipulated variables iteratively so that at each iteration only a linear least squares problem is to be solved in the MPC calculation. Due to its success in the deterministic MPC formulation, we believe the idea of the heuristic is also appropriate for the stochastic MPC formulation in this thesis.
2) Although the input bounds are assumed to be inactive (unless they have been fixed to their bounds already) during the heuristic, the uncertain values of all the inputs are still bounded with constraint (3.14e). Therefore, at the solution of the SOCP subproblem in the heuristic, the boundary of the uncertainty region of an input must be within its limits; if the limit is active, all the realizations of the uncertain input are forced to the limit according to Assumption 3.1. We recognize that this assumption is an approximation of real behavior, in which only a fraction of the realizations may be at the limit.
3) The reason for the success of heuristic lies in the special characteristics of the optimal control structure of the MPC formulation. The MPC controller typically wants to drive the controlled variables to their set points as quickly as possible to minimize its objective function, which is basically the sum of the difference between the controlled variables and their set points throughout the prediction horizon. It requires inputting the needed "energy" into the system at the beginning of the horizon. If the physical limits on the manipulated inputs (i.e. the opening of a valve can only range between 0%-100%) prevent inputting the required energy

immediately, the optimal manipulated inputs will tend to remain at their limits (i.e. a valve is fully open) to input the largest energy possible into the system.

4) The number of iterations in the heuristic is proportional to the length of the control horizon. Thus, the heuristic results in a small number of iterations.
5) The heuristic does not guarantee "global optimum" of solution, i.e., there may be another saturation pattern that is better than the one found by the heuristic. However, the heuristic converges to the optimum if the correct active set is selected.

This subsection and the last two subsections (Section 3.2.2-3.2.4) presents the method developed in this thesis to approximate the original bilevel stochastic optimization problem RMPC-CL with a limited number of single-level stochastic (convex) optimization problems RMPC-CLT. The next subsection will explain the developed method to enhance the process model (3.5b-3.5d) in the original formulation RMPC-CL for time-invariant uncertainty, with the goal of reducing the conservativeness in the uncertainty prediction.

3.2.5 The deviation model enhanced for time-invariant uncertainty

If the uncertain parameters of the plant do not change over time or they change slowly, we can assume they are invariant in the prediction horizon. In this situation, we will model the closed-loop dynamic system using deviation variables. There are several variable choices for the steady state about which the deviations are measured. Here, we will develop the method used consistently in this research for time-invariant systems. Further discussion of the importance of this choice is given in Appendix C.

To demonstrate the method using deviation variables, the uncertain process model (3.5b-3.5d) can be simplified by temporarily removing the noise variables as follows (because we will concentrate on the uncertainty in the process model):

$$x_{r,k+1} = A_r x_{r,k} + B_r u_{r,k} + B_{er} e_{r,k} + B_{dr} d_{m,k} \tag{3.15}$$

$$e_{r,k+1} = e_{r,k} \tag{3.16}$$

$$y_{r,k+1} = C_r x_{r,k+1} \tag{3.17}$$

$$k = 0, \cdots, p-1$$

where the parameters A_r, B_r, B_{er}, B_{dr}, C_r are uncertain but time-invariant. With the uncertain model (3.15-3.17), the predicted states $x_{r,k+1}$ and the controlled variables $y_{p,k+1}$ are different for different realizations of the uncertain parameters (i.e., their uncertainty is not zero). As discussed previously, the robust MPC without constraints should anticipate achieving its set points at steady state because (a) the values of u are known, and (b) the time-invariant model uncertainty in the predictions of x and y are compensated by the feedback e and the implicit integral mode in the MPC controller.

Model (3.15-3.17) includes uncertain parameters and predicts uncertainty in the process outputs *even if the manipulated variables do not change in the horizon*. However, the uncertainty only influences the outputs when the manipulated variables change. As a result, the robust MPC will perform conservatively at steady state, e.g., maintaining an excessive safety margin from controlled variable constraints. (See Appendix C for more extensive discussions on this issue.)

In this thesis, we modify the uncertain process model (3.15-3.17) for the prediction of uncertainty by calculating the deviation variables from a steady-state that would be determined by the most current manipulated variables u_{-1} and the measured disturbances $d_{m,-1}$. We will call this a "virtual" steady state because it does not occur in the process, although we can calculate it using the (nominal) model. A similar idea has been successfully applied to robust steady-state optimization by Kassmann et al. (2000).

For more details, we denote the variables at the virtual steady state as x_s, y_s, u_s, d_s, e_s, where u_s, d_s, e_s are known or are estimated at each controller execution as follows.

$$u_s = u_{-1} \tag{3.18}$$

$$d_s = d_{m,-1} \tag{3.19}$$

$$e_s = \hat{e}_0 \tag{3.20}$$

Then, x_s, y_s can be obtained using the nominal steady-state model.

$$x_s = Ax_s + Bu_s + B_e e_s + B_d d_s \tag{3.21}$$

$$y_s = Cx_s \tag{3.22}$$

We can express the model (3.15-3.17) as the deviation from the steady state in equations (3.21) and (3.22) as the following.

$$x_{r,k+1} - x_s = A_r(x_{r,k} - x_s) + B_r(u_{r,k} - u_s) + B_{dr}(d_{m,k} - d_s) \tag{3.23}$$

$$y_{r,k+1} - y_s = C_r(x_{r,k+1} - x_s) \tag{3.24}$$

$$k = 0, \cdots, p-1$$

As the system approaches a steady state x_{ss}, y_{ss}, u_{ss}, d_{ss}, the virtual steady state approaches the steady state too and uncertainty predicted using the deviation model (3.23-2.24) is appropriately small. If the system reaches the steady state, the virtual steady state will coincide with the actual steady-state, i.e. x_s=x_{ss}, y_s=y_{ss}, u_s=u_{ss}, d_s= d_{ss}, and $x_{r,k+1}$ and $y_{r,k+1}$ will be predicted to be x_{ss} and y_{ss} respectively, which means the their uncertainty is zero.

As a summary of the above discussion, the process model (3.15-3.17) overestimates the uncertainty caused by the changes in the manipulated variables, which may lead to conservative control. The deviation model (3.23-3.24), which recognizes that uncertainty results from changes in inputs, avoids this conservativeness. In this thesis, the robust MPC will use the deviation variable formulation for all the time-invariant systems for uncertainty prediction. Note that if we define the new deviation variables,

$$x'_{r,k+1} = x_{r,k+1} - x_s \tag{3.25}$$

$$x'_{r,k} = x_{r,k} - x_s \tag{3.26}$$

$$u'_{r,k+1} = u_{r,k} - u_s \tag{3.27}$$

$$d'_{m,k} = d_{m,k} - d_s \tag{3.28}$$

$$y'_{r,k+1} = y_{r,k+1} - y_s \tag{3.29}$$

$$e'_{r,k+1} = e_{r,k+1} - e_s \tag{3.30}$$

$$e'_{r,k} = e_{r,k} - e_s \tag{3.31}$$

$$k = 0,\cdots,p-1$$

then the deviation model (3.23-3.24) can be written into the form of model (3.15-3.17) (note that equations (3.20) and (3.30-3.31) implies $e'_{r,k}$=0 for all feedback variables over the prediction horizon). Therefore, the formulations derived previously are applicable to the situation using the deviation model, except that the variables should be replaced with the deviation variables as defined above. Accordingly, the set points and bounds of the deviation variables should be defined in deviation form as

$$y'_{sp} = y_{sp} - y_s \tag{3.32}$$

$$u'_{sp} = u_{sp} - u_s \tag{3.33}$$

$$y'_{\max,k+1} = y_{\max,k+1} - y_s \tag{3.34}$$

$$y'_{\min,k+1} = y_{\min,k+1} - y_s \tag{3.35}$$

$$u'_{\max,k} = u_{\max,k} - u_s \tag{3.36}$$

$$u'_{\min,k} = u_{\min,k} - u_s \tag{3.37}$$

$$k = 0, \cdots, p-1$$

For the convenience of discussion, we will not write out separate formulations for using deviation model in this chapter. When the deviation model is needed, we can modify the variables to deviation form and calculate the bounds as deviations as well before the optimization (using equations (3.25-3.37)) and if desired for plotting or plant implementation, restore the solution in the deviation-variable form back to the original form (using equations (3.25-3.37) again) after the optimization.

3.3 The Solution Techniques

In Section 3.2 we discussed how to transform the bilevel stochastic optimization problem RMPC-CL for robust MPC into a series of single-level stochastic optimization RMPC-CLT. In this section we will show how to solve the problem RMPC-CLT.

3.3.1 Solution with chance-constraints

The basic idea in solving problem RMPC-CLT is to approximate it by a deterministic optimization problem. The uncertainty in RMPC-CLT comes from the uncertain parameters in the plant behavior that appear in the closed-loop model (3.14b-3.14c). The closed-loop model (3.14b-3.14c) and the bounds (3.14e-3.14f)

$$\mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \tag{3.38a}$$

$$\mathbf{y}_r = L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_y\boldsymbol{\omega} \tag{3.38b}$$

$$\mathbf{u}_{\min} \le \mathbf{u}_r \le \mathbf{u}_{\max} \tag{3.38c}$$

$$\mathbf{y}_{\min} - \mathbf{s} \le \mathbf{y}_r \le \mathbf{y}_{\max} + \mathbf{s} \tag{3.38d}$$

can be combined into

$$L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_{u}\boldsymbol{\omega} \le \mathbf{u}_{\max} \tag{3.39a}$$

$$-L_{ur}\mathbf{t} - M_{ur}\boldsymbol{\theta} - N_{u}\boldsymbol{\omega} \le -\mathbf{u}_{\min} \tag{3.39b}$$

$$L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_{y}\boldsymbol{\omega} \le \mathbf{y}_{\max} \tag{3.39c}$$

$$-L_{yr}\mathbf{t} - M_{yr}\boldsymbol{\theta} - N_{y}\boldsymbol{\omega} \le -\mathbf{y}_{\min} \tag{3.39d}$$

Constraints (3.39a-3.39d) are linear inequalities with uncertain parameters. Ben-Tal and Nemirovski (1999) showed how to transform such inequalities to deterministic constraints when the uncertain parameters are within an ellipsoidal uncertainty region. Lobo et al. (1998) showed the transformation for the case where the uncertain parameters obey multivariate normal distribution using the chance-constrained program (Sen and Higle, 1999) framework. In this research, we adopt the latter formulation without requiring a normal distribution. To explain the approach, the lth constraint in (3.39a) will be considered:

$$L_{ur,l}\mathbf{t} + M_{ur,l}\boldsymbol{\theta} + N_{u,l}\boldsymbol{\omega} \le \mathbf{u}_{\max,l} \tag{3.40}$$

Here, $L_{ur,l}, M_{ur,l}, N_{u,l}$ denote the lth row of matrices L_{ur}, M_{ur}, N_{u} respectively and $\mathbf{u}_{\max,l}$ denotes the lth element in $\mathbf{u}_{\max}$. The idea of chance-constrained program is to guarantee the feasibility of constraint (3.40) at a confidence level α, i.e., to transform the constraint into the following form,

$$P_r(L_{ur,l}\mathbf{t} + M_{ur,l}\boldsymbol{\theta} + N_{u,l}\boldsymbol{\omega} \le \mathbf{u}_{\max,l}) \ge \alpha \tag{3.41}$$

If $\left(L_{ur,l}, M_{ur,l}, \boldsymbol{\omega}^T\right)$ follows a normal distribution, the above inequality is exactly equivalent to the following deterministic constraint (Lobo et al., 1998),

$$E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\theta} + N_{u,l}E(\boldsymbol{\omega}) + \Phi^{-1}(\alpha) \| V_{u,l}^{1/2}\left(\mathbf{t}^T, \boldsymbol{\theta}^T, N_{u,l}, 1\right)^T \|_2 \leq \mathbf{u}_{\max,l} \tag{3.42}$$

where $E(\cdot)$ denotes the expected value of the parameters in the brackets, $\Phi^{-1}(\alpha)$ denotes the inverse cumulative probability function of normal distribution, $V_{u,l}$ denotes the covariance matrix of $\left(L_{ur,l}, M_{ur,l}, \boldsymbol{\omega}^T, 1\right)$, $\| . \|_2$ means the L2-norm of the vector (Weisstein, 2009). The deterministic constraint (3.42) is called a second order conic constraint and is a convex inequality when the probability $\alpha > 1/2$. Note that the LHS of the constraint (3.42) is the maximum of the uncertain LHS of constraint (3.40) (in this expression, the maximum of the uncertain manipulated variable) with the confidence level α. Therefore, the constraint (3.42) requires the worst-case uncertain manipulated variable to be less than its upper bound (with confidence level α).

Note that the equivalence of constraints (3.41) and (3.42) is based on $L_{ur,l}, M_{ur,l}, \boldsymbol{\omega}$ obeying the normal distribution. Since $\boldsymbol{\omega}$ denotes unmeasured disturbances and noises in the system, it usually can be deemed to obey normal distribution. However, the parameters $L_{ur,l}, M_{ur,l}$ in closed-loop model may not be normally distributed, although such uncertain parameters may depend on normally distributed uncertainty. In this case, the reformulation from constraint (3.41) to constraint (3.42) is an approximation.

If we approximate all the uncertain linear inequalities (3.39a-3.39d) in the same way, the problem RMPC-CLT becomes a deterministic Second Order Cone Program (SOCP, Lobo et al., 1998) that can be solved efficiently and reliably with a state-of-the-art interior point optimizer, such as CPLEX. This SOCP problem, which is called RMPC-CLTSOCP, is as follows:

RMPC-CLTSOCP:

$$\min_{\mathbf{t},\mathbf{s}} \quad (\mathbf{y}-\mathbf{y}_{sp})^T\tilde{Q}(\mathbf{y}-\mathbf{y}_{sp})+(\mathbf{u}-\mathbf{u}_{sp})^T\tilde{R}(\mathbf{u}-\mathbf{u}_{sp}) + (I_{\Delta 1}\mathbf{u}-I_{\Delta 2}u_{-1})^T\tilde{W}(I_{\Delta 1}\mathbf{u}-I_{\Delta 2}u_{-1})+\mathbf{s}^T\tilde{W}_s\mathbf{s} \tag{3.43a}$$

s.t.

$$\left.\begin{aligned} & E(L_{ur,l})\mathbf{t}+E(M_{ur,l})\boldsymbol{\theta}+N_{u,l}E(\boldsymbol{\omega})-\mathbf{u}_{\max,l} + \Phi^{-1}(\alpha)\,\| V_{u,l}^{1/2}\left(\mathbf{t}^T,\boldsymbol{\theta}^T,N_{u,l},1\right)^T \|_2 \le 0, && (3.43\text{b}) \\ & -E(L_{ur,l})\mathbf{t}-E(M_{ur,l})\boldsymbol{\theta}-N_{u,l}E(\boldsymbol{\omega})+\mathbf{u}_{\min,l} + \Phi^{-1}(\alpha)\,\| V_{u,l}^{1/2}\left(\mathbf{t}^T,\boldsymbol{\theta}^T,N_{u,l},1\right)^T \|_2 \le 0, && (3.43\text{c}) \end{aligned}\right\} l=1,\cdots,n_u n$$

$$\left.\begin{aligned} & E(L_{yr,l})\mathbf{t}+E(M_{yr,l})\boldsymbol{\theta}+N_{y,l}E(\boldsymbol{\omega})-\mathbf{y}_{\max,l}-s_l + \Phi^{-1}(\alpha)\,\| V_{y,l}^{1/2}\left(\mathbf{t}^T,\boldsymbol{\theta}^T,N_{y,l},1\right)^T \|_2 \le 0, && (3.43\text{d}) \\ & -E(L_{yr,l})\mathbf{t}-E(M_{yr,l})\boldsymbol{\theta}-N_{y,l}E(\boldsymbol{\omega})+\mathbf{y}_{\min,l}-s_l + \Phi^{-1}(\alpha)\,\| V_{y,l}^{1/2}\left(\mathbf{t}^T,\boldsymbol{\theta}^T,N_{y,l},1\right)^T \|_2 \le 0 && (3.43\text{e}) \end{aligned}\right\} l=1,\cdots,n_y p$$

$$(\mathbf{I}_\delta-\boldsymbol{\delta})\mathbf{t}=\mathbf{u}_c \tag{3.43f}$$

$$\mathbf{s}\ge 0 \tag{3.43g}$$

where $L_{yr,l}, M_{yr,l}, N_{y,l}$ denote the lth row of matrices L_{yr}, M_{yr}, N_y respectively and $\mathbf{u}_{\min,l}, \mathbf{y}_{\max,l}, \mathbf{y}_{\min,l}$ denotes the lth element in $\mathbf{u}_{\min}, \mathbf{y}_{\max}, \mathbf{y}_{\min}$ respectively, $V_{y,l}$ denotes the covariance matrix of $\left(L_{yr,l}, M_{yr,l}, \boldsymbol{\omega}^T, 1\right)$.

Note that guaranteeing constraint-wise confidence level α does not ensure the satisfaction of all the constraints in the problem with such confidence criteria. Actually, a joint chance constrained program needs to be solved to achieve a specified confidence for the satisfaction of all the constraint, but this problem is typically very difficult to solve (Li et al., 2008). So this thesis, we propose to achieve the desired overall confidence level by select an appropriate constraint-wise confidence level and solving the resulting individual chance constraint program (such as problem RMPC-CLTSOCP) A trial and error procedure can be performed offline with numerical simulation for the selection of the appropriate constraint-wise confidence level.

The variance matrix $V_{u,l}$ is estimated through Monte Carlo sampling as follows (and a similar approach is used for $V_{y,l}$):

1) Randomly select a sample of the open-loop uncertain parameters ($A_{r,k+1}$, $B_{r,k+1}$, $B_{dr,k+1}$, $C_{r,k+1}$, w_k, v_{k+1}) shown in formulation (3.5) from their distribution within the α confidence level;
2) Calculate closed-loop uncertain parameters (L_{ur}, L_{yr}, M_{ur}, M_{yr}, N_u, N_y shown in formulation (3.32)) accordingly;
3) Repeat procedure (a-b) for a number of samples of the open-loop uncertain parameters and obtain different groups of closed-loop uncertain parameters, which are then be used to estimate $V_{u,l}$ according to the standard technique (Box et al., 2008), i.e., $V_{u,l}=X^TX/(n_s-1)$ where X denotes the matrix whose rows contain difference between different realizations of vector ($L_{ur,l}$, $M_{ur,l}$, $\boldsymbol{\omega}^T$, 1) and their average values, and n_s denotes the total number of realizations.

Note that the total number of samples n_s should be sufficiently large so that the variance calculation is accurate enough for the problem RMPC-CLTSOCP. We note that that the covariance matrix $V_{u,l}$ or $V_{y,l}$ reflects the range of the uncertainty (of a manipulated or controlled variable) through the norms in the constraints (3. 43b-3. 43e), so we choose the total number of samples n_s such that the a larger number of samples does not change the norm of the matrix $V_{u,l}$ or $V_{y,l}$ by 5% or more of its original value. We use the spectral norm of matrix that is induced from the L2-norm of vector (Weinstein, 2009). This procedure resulted in 100 samples being adequate for the calculation of the covariance matrices in the numerical experimentation for all the case studies in this thesis.

Substantial computing for the Monte Carlo sampling is performed off-line as part of the controller design and tuning, and therefore, it does not affect the tractability of the real-time solution. The computational complexity of the off-line calculation can be found in the next subsection. The covariance matrices $V_{u,l}$, $V_{y,l}$ depend on the saturation pattern of the manipulated variables. Thus, these covariance matrices must be updated for each controller execution and the method is given in the next subsection.

3.3.2 Efficient uncertainty characterization

To obtain the covariance matrices $V_{u,l}$, $V_{y,l}$ according to the saturation pattern of the manipulated variables in real-time, a natural way would be to obtain and store $V_{u,l}$, $V_{y,l}$

for each of the saturation patterns respectively before the real-time application (off-line). However, the number of saturation patterns is exponential with respect to the product of the number of manipulated variables times the number of time steps in the control horizon ($2^{n_u \cdot n}$), which could make the off-line sampling results unrealistically large even for small problems. For example, when n_u = 6, n = 9, the total number of saturation patterns is about 10^6. In this case, if the mean and variance matrices for one saturation pattern required 1 MB space, the total space required to store all the variance matrices would be about 10^3 TB!

Therefore, an on-line uncertainty characterization method has been developed in this research to reduce the complexity, which involves sampling and storing the uncertainties for the case of saturated manipulated inputs off-line and updating this result for specific saturation pattern on-line. The updating rule is key for applying this method. Assume the closed-loop model of manipulated variables $\mathbf{u}_r$ for a particular saturation pattern is:

$$\mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega}$$

where

$$L_{ur} = \begin{bmatrix} L_{ur}^{(0,0)} & & & & \\ L_{ur}^{(1,0)} & L_{ur}^{(1,1)} & & & \\ L_{ur}^{(2,0)} & L_{ur}^{(2,1)} & L_{ur}^{(2,2)} & & \\ \vdots & & & \ddots & \\ L_{ur}^{(m-1,0)} & L_{ur}^{(n-1,1)} & \cdots & L_{ur}^{(n-1,n-2)} & L_{ur}^{(n-1,n-1)} \end{bmatrix} \in R^{(n \cdot n_u)\times(n \cdot n_u)},$$

and the block in the matrix L_{ur} on the row k_1 ($k_1 = 0,\ldots,n-1$) and the column k_2 ($k_2 = 0,\ldots,k_1$),

$$L_{ur}^{(k_1,k_2)} = \begin{pmatrix} L_{ur,1,1}^{(k_1,k_2)} & \cdots & L_{ur,1,n_u}^{(k_1,k_2)} \\ \vdots & \ddots & \vdots \\ L_{ur,n_u,1}^{(k_1,k_2)} & \cdots & L_{ur,n_u,n_u}^{(k_1,k_2)} \end{pmatrix} \in R^{n_u \times n_u}.$$

Note that when $k_1 = k_2$, $L_{ur}^{(k_1,k_2)}$ is an identity matrix. Also,

$$M_{ur} = \begin{pmatrix} M_{ur}^{(0)} \\ \vdots \\ M_{ur}^{(n-1)} \end{pmatrix} \in R^{(n \cdot n_u) \times n_\theta} \text{ and } M_{ur}^{(k_1)} = \begin{pmatrix} M_{ur,1}^{(k_1)} \\ \vdots \\ M_{ur,n_u}^{(k_1)} \end{pmatrix} \in R^{n_u \times n_\theta},$$

$$N_{ur} = \begin{pmatrix} N_{ur}^{(0)} \\ \vdots \\ N_{ur}^{(n-1)} \end{pmatrix} \in R^{(n \cdot n_u) \times n_\omega} \text{ and } N_{ur}^{(k_1)} = \begin{pmatrix} N_{ur,1}^{(k_1)} \\ \vdots \\ N_{ur,n_u}^{(k_1)} \end{pmatrix} \in R^{n_u \times n_\omega},$$

where n_θ, n_ω denote the numbers of elements in $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ respectively, $k_1 = 0,\ldots,n-1$.

If manipulated inputs at a time step that were assumed to be unsaturated are found to encounter its constraint, e.g. the *i*th manipulated input at time step *j* (denoted by $u_{r,i,j}$), we need to update the closed-loop model of the manipulated variables as follows.

$$L_{ur,o_1,o_2}^{(k_1,k_2)} * = L_{ur,o_1,o_2}^{(k_1,k_2)} - L_{ur,o_1,i}^{(k_1,j)} L_{ur,i,o_1}^{(j,k_2)} \tag{3.44a}$$

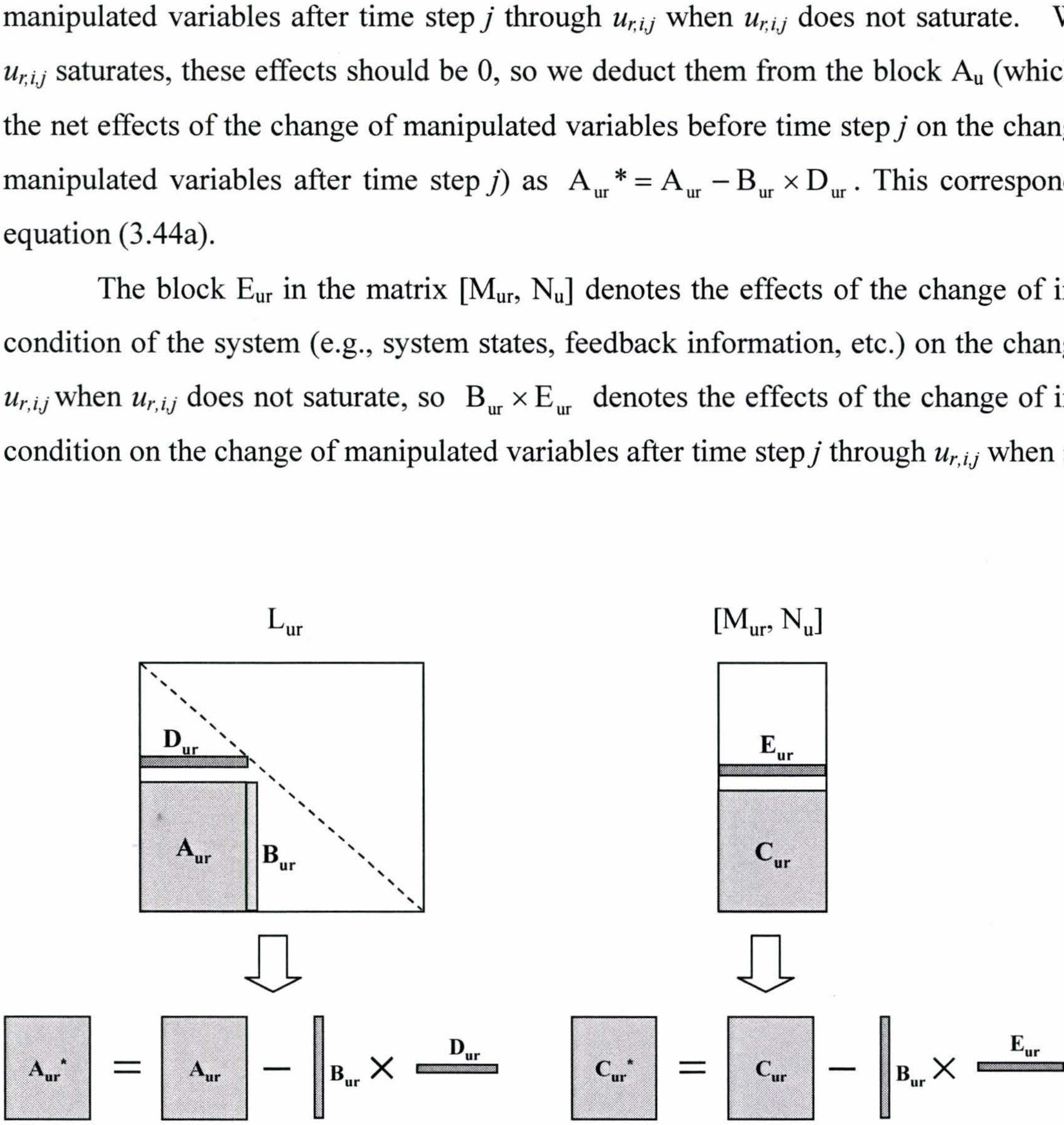
$$\left(M_{ur,o_1}^{(k_1)}, \quad N_{u,o_1}^{(k_1)}\right) * = \left(M_{ur,o_1}^{(k_1)}, \quad N_{u,o_1}^{(k_1)}\right) - L_{ur,o_1,i}^{(k_1,j)} \left(M_{ur,i}^{(j)}, \quad N_{u,i}^{(j)}\right) \tag{3.44b}$$

$$k_1 = j+1,\ldots,n-1, \quad k_2 = 0,\ldots,j-1, \quad o_1 = 1,\ldots,n_u, \quad o_2 = 1,\ldots,n_u$$

where $L_{ur,o_1,o_2}^{(k_1,k_2)} *$ and $\left(M_{ur,o_1}^{(k_1)}, \quad N_{u,o_1}^{(k_1)}\right) *$ denote the updated values of $L_{ur,o_1,o_2}^{(k_1,k_2)}$ and $\left(M_{ur,o_1}^{(k_1)}, \quad N_{u,o_1}^{(k_1)}\right)$, respectively. The proof of this updating rule is presented in Appendix E.

Figure 3.6 illustrates this updating rule. The block D_{ur} in the matrix L_{ur} denotes the effects of the change of manipulated variables before time step j on the change of $u_{r,i,j}$ when $u_{r,i,j}$ does not saturate, the block B_{ur} denotes the effects of the change of $u_{r,i,j}$ on the change of manipulated variables after time step j. Therefore, $B_{ur} \times D_{ur}$ denotes the effects of the change of manipulated variables before time step j on the change of the manipulated variables after time step j through $u_{r,i,j}$ when $u_{r,i,j}$ does not saturate. When $u_{r,i,j}$ saturates, these effects should be 0, so we deduct them from the block A_u (which are the net effects of the change of manipulated variables before time step j on the change of manipulated variables after time step j) as $A_{ur}{}^* = A_{ur} - B_{ur} \times D_{ur}$. This corresponds to equation (3.44a).

The block E_{ur} in the matrix $[M_{ur}, N_u]$ denotes the effects of the change of initial condition of the system (e.g., system states, feedback information, etc.) on the change of $u_{r,i,j}$ when $u_{r,i,j}$ does not saturate, so $B_{ur} \times E_{ur}$ denotes the effects of the change of initial condition on the change of manipulated variables after time step j through $u_{r,i,j}$ when $u_{r,i,j}$



Dimensions of matrices: $A_{ur}:[n_u(n-1-j)]\times[n_u(j-1)]$ $C_{ur}:[n_u(n-1-j)]\times[n_\theta+n_\omega]$

$B_{ur}:[n_u(n-1-j)]\times 1$ $D_{ur}:1\times[n_u(j-1)]$ $E_{ur}:1\times[n_\theta+n_\omega]$

Figure 3.6 Illustration of model-updating for manipulated variables with $u_{r,j,i}$ saturation does not saturate. When $u_{r,i,j}$ saturates, these effects should be 0, so we deduct them from the block C_u (which are the net effects of the change of initial condition on the change of manipulated variables after time step *j*) as $C_{ur}* = C_{ur} - B_{ur} \times E_{ur}$. This corresponds to equation (3.44b).

Similarly, the closed-loop model of the controlled variables can be updated as

$$L_{yr,o_1,o_2}^{(k_1,k_2)}* = L_{yr,o_1,o_2}^{(k_1,k_2)} - L_{yr,o_1,i}^{(k_1,j)} L_{yr,i,o_1}^{(j,k_2)} \tag{3.45a}$$

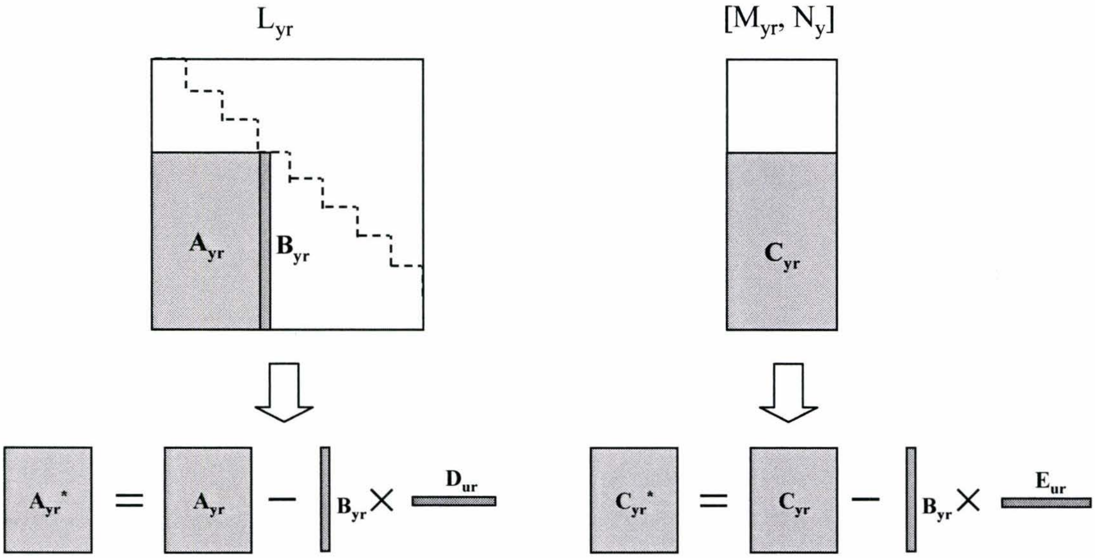
$$\left(M_{yr,o_1}^{(k_1)}, \quad N_{y,o_1}^{(k_1)}\right)* = \left(M_{yr,o_1}^{(k_1)}, \quad N_{y,o_1}^{(k_1)}\right) - L_{yr,o_1,i}^{(k_1,j)}\left(M_{yr,i}^{(j)}, \quad N_{y,i}^{(j)}\right) \tag{3.45b}$$

$$k_1 = j+1,\dots,p, \quad k_2 = 0,\dots,j, \quad o_1 = 1,\dots,n_y, \quad o_2 = 1,\dots,n_u$$

where $L_{yr,o_1,o_2}^{(k_1,k_2)}*$ and $\left(M_{yr,o_1}^{(k_1)}, \quad N_{y,o_1}^{(k_1)}\right)*$ denotes the updated value of $L_{yr,o_1,o_2}^{(k_1,k_2)}$ and $\left(M_{yr,o_1}^{(k_1)}, \quad N_{y,o_1}^{(k_1)}\right)$. Figure 3.7 illustrates this updating rule, where $A_{yr}* = A_{yr} - B_{yr} \times D_{ur}$ corresponds to (3.45a) and $C_{yr}* = C_{yr} - B_{yr} \times E_{ur}$ corresponds to (3.45b).

The online model-updating calculation can be performed for each of the saturated manipulated inputs sequentially. Then, we have the closed-loop model coefficients L_{ur}, M_{ur}, L_{yr}, M_{yr} of a particular saturation pattern for all the samples, which can be used to calculate the covariance matrices $V_{u,l}$, $V_{y,l}$.

The time complexity of the on-line calculation in the worst case (i.e., all the manipulated variables in the horizon saturate) is $O(n_s(n_u n)^3)$ and the storage complexity $O(n_s(n_u n)^2)$, where n_s denotes the total number of samples. Note that we only need to calculate and store the uncertainty information for the case with no manipulated variables saturation, so we avoid the exponential complexity of offline computation. The time complexity of the off-line calculation is $O(n_s(n_u n)^3)$ and storage complexity $O(n_s(n_u n)^2)$. Please refer to Appendix E for the discussion of the computational complexity for the on-line and off-line calculations.



Dimensions of matrices: $A_{yr}:[n_y(p-j)]\times[n_y j]$ $C_{yr}:[n_y(p-j)]\times[n_\theta+n_\omega]$
$B_{yr}:[n_y(p-j)]\times 1$ $D_{yr}:1\times[n_y j]$ $E_{yr}:1\times[n_\theta+n_\omega]$

Figure 3.7 Illustration of model-updating for controlled variables with $u_{rj,i}$ saturation

3.4 Uncertainties in State-Estimation

As mentioned in Section 3.1 and Appendix A, equations (3.1-3.2) denote the "nominal" steady-state Kalman filter that is based on assuming the nominal model of the system is perfect. In the context of robust MPC, however, we need to address the parametric uncertainty in the state estimation explicitly if the states cannot be measured directly. Here we introduce an approach developed in this research to incorporate parametric uncertainties in state estimation for time-invariant process.

First, we assume the real process model coefficients (A_r, B_r, B_{dr}, C_r) are known. Then, the state estimation can be performed as follows.

$$\hat{x}_{r,0} = \hat{x}_{r,0/-1} + L_{xr}(y_{m,0} - C_r\hat{x}_{r,0/-1}) \quad (3.46)$$

$$\hat{e}_{r,0} = \hat{e}_{r,0/-1} + L_{er}(y_{m,0} - C_r\hat{x}_{r,0/-1}) \quad (3.47)$$

where L_{xr}, L_{er} denote the steady-state Kalman gains calculated according to A_r, C_r using the well-known method (Kalman, 1960) whose details are shown in Appendix A. $\hat{x}_{r,0/-1}$, $\hat{e}_{r,0/-1}$ denote the estimate of the states and feedback errors at time step k given the output measurement at time step k-1 and $\hat{x}_{r,0}$ and $\hat{e}_{r,0}$ denote the update of the estimate given the output measurement at time step k. $\hat{x}_{r,0/-1}$, $\hat{e}_{r,0/-1}$ can be calculated by

$$\hat{x}_{r,0/-1} = A_r\hat{x}_{r,-1} + B_r u_{r,-1} + B_{er}\hat{e}_{r,-1} + B_{dr}d_{m,-1} \quad (3.48)$$

$$\hat{e}_{r,0/-1} = \hat{e}_{r,-1} \quad (3.49)$$

The equations (3.46-3.49) can be integrated into the following form

$$\zeta_0 = G_{\zeta\zeta}\zeta_{-1} + G_{\zeta\mu}\mu_{-1} \quad (3.50)$$

where we define

$$\zeta_{-k} = \begin{pmatrix} \hat{x}_{r,-k} \\ \hat{e}_{r,-k} \end{pmatrix}, \quad \mu_{-k} = \begin{pmatrix} y_{m,k} \\ u_{-k-1} \\ d_{m,-k-1} \end{pmatrix}, \quad G_{\zeta\zeta} = \begin{pmatrix} (I - L_{xr}C_r)A_r & (I - L_{xr}C_r)B_{er} \\ -L_{er}C_rA_r & I - L_{er}C_rB_{er} \end{pmatrix},$$

$$G_{\zeta\mu} = \begin{pmatrix} L_{xr} & (I - L_{xr}C_r)B_r & (I - L_{xr}C_r)B_{dr} \\ L_{er} & -L_{er}C_rB_r & -L_{er}C_rB_{dr} \end{pmatrix}.$$

Equation (3.50) means that the estimated states at the current time step ζ_0 depend linearly on the estimated states at the last time step ζ_{-1} (as well as the inputs, disturbances and the output measurements in μ_{-1}). ζ_{-1} depends on ζ_{-2} in the similar way, which is

$$\zeta_{-1} = G_{\zeta\zeta}\zeta_{-2} + G_{\zeta\mu}\mu_{-2} \tag{3.51}$$

and equations (3.50-3.51) can be combined to give the following equation

$$\zeta_0 = G_{\zeta\zeta}^2\zeta_{-2} + G_{\zeta\zeta}G_{\zeta\mu}\mu_{-2} + G_{\zeta\mu}\mu_{-1} \tag{3.52}$$

Repeat the above procedure for the previous $p\_$ time steps iteratively, and then we get the following equation,

$$\zeta_0 = G_{\zeta\zeta}^{p\_}\zeta_{-p\_} + \sum_{i=0}^{p\_-1}\left(G_{\zeta\zeta}^{i}G_{\zeta\mu}\mu_{-1-i}\right) \tag{3.53}$$

Note that even the state estimates $\zeta_{-p\_}$ may still be uncertain. However, we can set $p\_ \geq p_{obs}$, where p_{obs} denotes the least number of time steps for the nominal estimates of states to converge to the real states (no matter where the nominal estimate is correct or not); the symbol p_{obs} is called the (backward) horizon of the observer in the thesis. Refer to Appendix A for the way to obtain p_{obs}. Therefore, the current states and feedback ζ_0 can be estimated using the nominal estimate of the states the $p\_$ time steps ago ($\zeta_{-p\_}$) through equation (3.53). We call $p\_$ the estimation horizon in this thesis.

Note that the nominally estimated states at a steady state can be deemed as the real states. So if the system has been at steady state during the previous p_{obs} time steps, e.g. k time steps ago ($k<p_{obs}$), we could estimate ζ_0 using the nominal estimate of the states and feedback at that time step $\zeta_{-p\_} = \zeta_{-k}$ through equation (3.53).

The above discussion is based on assuming the real process model coefficients (A_p,

B_p, B_{dp}, C_p) are known. If they are unknown and uncertain, the coefficients $G_{\zeta\zeta}$ $G_{\zeta\mu}$ in equation (3.53) should be uncertain accordingly. So the estimate of the current states and feedback errors through equation (3.53) is uncertain, whose uncertainty depends on the estimated states and feedback $p\_$ time steps ago as well as the measurement of controlled and disturbance variables and the implemented manipulated variables during the past $p\_$ time steps. The robust MPC considering this uncertainty in the state estimate will have the same formulation as formulation (3.14) (so that the same SOCP formulation (3.43)), except that the nominal estimate of the current states and feedback $\hat{x}_0^T, \hat{e}_0^T$ is replaced by its uncertain expression in equation (3.14), thus in the uncertain closed-loop model (3.14b-3.14c), the vector

$$\boldsymbol{\theta} = \left(u_{-1}^T, \tilde{\mathbf{d}}_m^T, \hat{x}_0^T, \hat{e}_0^T\right)^T \tag{3.54}$$

is changed into

$$\boldsymbol{\pi} = \left(u_{-p\_-1}^T, \cdots, u_{-1}^T, y_{m,p\_}^T, \cdots, y_{m,0}^T, d_{m,-p\_-1}^T, \cdots, d_{m,-1}^T, \tilde{\mathbf{d}}_m^T, \hat{x}_{-p\_}^T, \hat{e}_{-p\_}^T\right)^T \tag{3.55}$$

Here the "current system state" cannot be simply expressed by its nominal estimate. It should be expressed using its nominal estimate $p\_$ time steps before as well as the measurement of controlled and disturbance variables and the implemented manipulated variables during the past $p\_$ time steps. The uncertain matrices M_{ur}, M_{yr} in the uncertain closed-loop model (3.14b-3.14c) are changed accordingly. So addressing uncertainty in state estimate using the new M_{ur}, M_{yr} and $\boldsymbol{\pi}$ will change the variance calculation of the manipulated and controlled variables. The benefits of integrating the uncertainty in state estimate will be shown in a case study in Section 3.6.

Another important issue to be clarified is: the Kalman filter is used here as an observation that provides a nominally stable observer instead of "filtering" the noises, because the all the uncertainties including the noises are addressed explicitly in the robust MPC framework. Therefore, other stable observers could be used to replace the Kalman filter in the formulation, e.g., a Luenberger observer (Luenberger, 1971).

3.5 Summary of the Robust MPC Algorithm

According to the discussions from Section 3.2 to Section 3.4, the new robust MPC algorithm can be summarized as follows.

Calculation performed off-line:

1) Calculate the nominal value of the coefficients L_{ur}, M_{ur}, N_{ur}, L_{yr}, M_{yr}, N_{yr} in the closed-loop model (3.14b-3.14c), for the situation where no input bounds are active, using the method derived in Appendix D;
2) Repeat the calculation in step 1 for samples of the open-loop uncertain system (100 samples used in this thesis). Calculate the covariance matrices for the closed-loop coefficients according to the results of the sample calculations.

Calculation performed on-line at each controller execution period:

1) Obtain the set points of the controlled and manipulated variables, y_{sp}, u_{sp} according to plant personnel or upper level controller/optimizer.
2) Read new measurements of controlled variables and the measured disturbances $y_{m,0}$, $d_{m,0}$ respectively. Set u_0 to be the implemented manipulated variables in the last controller execution.
3) Calculate nominally predicted controlled variable for the current time step, y_0, according to the previous implemented manipulated variables, measured disturbances and nominally estimated state and feedback variables, and estimate the nominal state and feedback variables for the current time step, $\hat{x}_0$, $\hat{e}_0$.
4) If the uncertainty in the system is time-invariant and the deviation model is needed, calculate the virtual steady state according to equations (3.18-3.22) and express the system variables, set points and bounds as deviations from the virtual steady state according to equations (3.25-3.37); otherwise skip this step.

5) Assume no bounds are active in the future, and set all diagonal elements in $\mathbf{I}_\delta$ to 1. Then solve problem RMPC-CLTSOCP equations (3.43a-3.43g) if all the system states are measured, or estimate and store the current state and feedback variables and solve the following problem RMPC-SOCP2 (with all the variables and parameters in the formulation same as defined before) to include the uncertainty in state estimation if not all the system states are measurable:

RMPC-CLTSOCP2:

$$\min_{\mathbf{t},\mathbf{s}} \quad (\mathbf{y}-\mathbf{y}_{sp})^T \widetilde{Q}(\mathbf{y}-\mathbf{y}_{sp}) + (\mathbf{u}-\mathbf{u}_{sp})^T \widetilde{R}(\mathbf{u}-\mathbf{u}_{sp}) + (I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1})^T \widetilde{W}(I_{\Delta 1}\mathbf{u} - I_{\Delta 2}u_{-1}) + \mathbf{s}^T \widetilde{W}_s \mathbf{s} \tag{3.56a}$$

s.t.

$$\left.\begin{aligned} &E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\pi} + N_{u,l}E(\boldsymbol{\omega}) - \mathbf{u}_{\max,l} + \Phi^{-1}(\alpha)\,\| V_{u,l}^{1/2}\left(\mathbf{t}^T, \boldsymbol{\pi}^T, N_{u,l}, 1\right)^T \|_2 \le 0, \quad (3.56\text{b}) \\ &-E(L_{ur,l})\mathbf{t} - E(M_{ur,l})\boldsymbol{\pi} - N_{u,l}E(\boldsymbol{\omega}) + \mathbf{u}_{\min,l} + \Phi^{-1}(\alpha)\,\| V_{u,l}^{1/2}\left(\mathbf{t}^T, \boldsymbol{\pi}^T, N_{u,l}, 1\right)^T \|_2 \le 0, \quad (3.56\text{c}) \end{aligned}\right\} l = 1, \cdots, n_u n$$

$$\left.\begin{aligned} &E(L_{yr,l})\mathbf{t} + E(M_{yr,l})\boldsymbol{\pi} + N_{y,l}E(\boldsymbol{\omega}) - \mathbf{y}_{\max,l} - s_l + \Phi^{-1}(\alpha)\,\| V_{y,l}^{1/2}\left(\mathbf{t}^T, \boldsymbol{\pi}^T, N_{y,l}, 1\right)^T \|_2 \le 0, \quad (3.\ 56\text{d}) \\ &-E(L_{yr,l})\mathbf{t} - E(M_{yr,l})\boldsymbol{\pi} - N_{y,l}E(\boldsymbol{\omega}) + \mathbf{y}_{\min,l} - s_l + \Phi^{-1}(\alpha)\,\| V_{y,l}^{1/2}\left(\mathbf{t}^T, \boldsymbol{\pi}^T, N_{y,l}, 1\right)^T \|_2 \le 0 \quad (3.\ 56\text{e}) \end{aligned}\right\} l = 1, \cdots, n_y p$$

$$(\mathbf{I}_\delta - \boldsymbol{\delta})\mathbf{t} = \mathbf{u}_c \tag{3. 56f}$$

$$\mathbf{s} \ge 0 \tag{3. 56g}$$

6) The solution gives the uncertain trajectory of the manipulated variables in the control horizon. If some manipulated variables, which are assumed unsaturated, have a value(s) at its bound (i.e. the boundaries of their uncertainty regions reach the upper or lower bounds on these manipulated variables), go to step 7; otherwise,

end the iterative procedure of the heuristic and go to step 8 (the current solution is the final solution obtained by the heuristic).

7) Set all manipulated variables, which are at their bounds at the earliest time step, to their bound values (by specifying $\mathbf{I}_{\delta}$, $\mathbf{u}_c$). Update the closed-loop model coefficients L_{ur}, M_{ur}, N_{ur}, L_{yr}, M_{yr}, N_{yr} according to the current saturation pattern using equations (3.44a-3.44b) and (3.45a-3.45b), and solve problem RMPC-CLTSOCP (if all the system states are measurable) or RMPC-CLTSOCP2 (if some system states are not measurable). Then go to step 6. Note that any manipulated variable that has been set to its bound value will be constrained for the remainder of the iterative procedure.
8) If the uncertainty in the system is time-invariant and the deviation model has been used in the previous calculation, restore the solution from the deviation variable mode using equations (3.25-3.37); otherwise skip this step.
9) Implement the values of the manipulated variable in the first future controller execution period in the solution.

3.6 Case Study Results and Discussion

The simulation case studies were performed on a PC with Intel Core 2 Duo 3.0 GHz, 4GB memory and Windows Vista. The solution for the plant simulation is programmed in MATLAB 7.5, and the QP and SOCP problems are solved in GAMS with the interior point (barrier) solver of CPLEX 11. The data in MATLAB and CPLEX are exchanged using the interface software MATGAMS developed by Ferris (2005). All the system models are initially expressed with continuous input-output model in S-domain, and they are all discretized and transformed into state-space model using the Control System Toolbox in MATLAB 7.5.

3.6.1 The control methods evaluated in the case studies

We will evaluate several control methods in the case studies, through which the advantage of the new robust MPC method will be demonstrated. We basically compare the following three methods:

1) The nominal MPC

This method solves the QP problem (3.1) at each controller execution period, where the initial states and feedback are estimated using equations (3.2-3.3).

2) The robust MPC (developed in this thesis)

The detailed steps to implement this method are shown in Section 3.6.

3) The open-loop robust MPC

Here the open-loop robust MPC means the robust MPC method using open-loop uncertainty prediction, as discussed in Section 3.2.1. In this (incomplete) formulation the future controller actions are assumed to be unchanged for different realizations of the plant. This is equivalent to assuming that the control laws of the closed-loop system in the future horizon are $\mathbf{u}_r=\mathbf{t}$ and the dynamic performance of the system is optimized is by adjusting $\mathbf{t}$. Therefore, this method can be implemented in the same way as detailed in Section 3.6 with the uncertainty calculated assuming that all the manipulated variables saturated, but with the values of the manipulated variables determined by $\mathbf{t}$ instead of being fixed to the bounds. The active set heuristic is not needed here because the saturation pattern has been defined.

3.6.2 CSTR control system 1

The first case study applies control to the Continuous Stirred-Tank Reactor (CSTR) process in the Appendix C of Marlin (2000), page 897-908, which is shown in Figure 3.8. In this case study, the inlet feed concentration of A (C_{A0}) into the reactant is used to control the outlet concentration of A (C_A). The temperature of the reactor is maintained constant by a temperature controller manipulating the cooling flow rate. So, we consider the CSTR system to be isothermal. The non-linear plant model is given by the following equation,

$$Vol\frac{dC_A}{dt} = F(C_{A0} - C_A) - VolK_A C_A \tag{3.57}$$

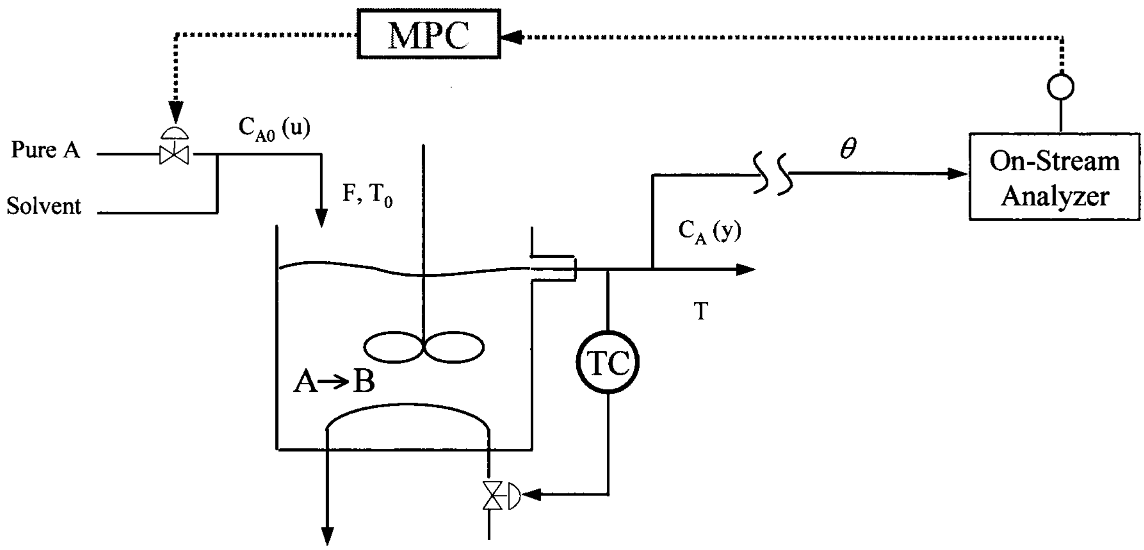


Figure 3.8 CSTR control system 1

where K_A is the constant first-order reaction rate, *Vol* is the reactor volume, *F* is the volumetric feed in flow rate.

This non-linear model can be linearized at a particular steady-state operating point and the linear differential equation expressed as a transfer function as the following.

$$y(s) = \frac{K_p}{\tau_p s + 1} e^{-\theta s} u(s) \tag{3.58}$$

where y is the controlled variable C_A, u is the manipulated variable C_{A0}, $\theta = 0.9$ minutes denotes the time for the output flow to reach the remote component analyzer, which introduces the delay between u and y. See Appendix F for the details of the linearization procedure and the parameters and operating points used in the thesis.

The uncertainty of the system comes from the slowly varying inlet flow rate F, whose uncertain value is assumed to obey normal distribution with mean 1 m^3/min and standard deviation 0.3 m^3/min. The nominal plant model is derived at F=1 m^3/min as, Nominal Model CSTR-1:

$$y(s) = \frac{0.8953e^{-0.9s}}{0.8953s+1}u(s) \tag{3.59}$$

In the case studies, we evaluate the performance of the controllers at two specific plant mismatch realizations in the case study. One is with F=1.87 m^3/min and its model is:

Plant CSTR-1.1:

$$y(s) = \frac{0.9417e^{-0.9s}}{0.5025s+1}u(s) \tag{3.60}$$

The other is with F=0.41 m^3/min and its model is:

Plant CSTR-1.2:

$$y(s) = \frac{0.7790e^{-0.9s}}{1.8957s+1}u(s) \tag{3.61}$$

These plants represent a "faster" plant (Plant CSTR-1.1 with a higher gain and smaller time constant) and a "slower" plant (Plant CSTR-1.2 with a lower gain and larger time constant). Both are within the uncertainty considered in the robust controller design.

The controller execution period for this system is selected to be 0.3 minutes, so the models of the system are discretized with sampling time of 0.3 minutes. The state-space form of the reactor model *without time delays and feedback variables* has the state vector x with 1 element and the system is controllable and observable. The feedback scheme assumes the unmeasured disturbance enters the system through the input channel (See Appendix A for more discussion on the selection of unmeasured disturbance model), which introduces the unmeasured disturbance vector e with 1 element. So the augmented system with x and e has 2 states, and it is detectable. Furthermore, the time delay between y and u is described by 3 additional states using the method introduced in Appendix C.

Since these 3 states denote the u in the last 3 time steps, they are known, and no observer gain is need for them.

The tuning of all nominal MPC controllers in this thesis follows these guidelines: a) The control and prediction horizons n, p are tuned according to the guidelines introduced in Camacho and Bordons (1999). b) The observer horizon p_{obs} and observer gains are tuned according to the discussion in Appendix A. c) The weighting matrices Q and R, W are used to keep an acceptable trade-off between variability of the controlled variables y and the variability of the manipulated variables u in the closed-loop dynamics. In this thesis, the variability of variable is evaluated through set point step change test by calculating the sum of its squared difference from its set point; and we tune the weighing matrices such that the variability of u is within the $\pm 50\%$ of the variability of y (when the system model is so scaled that the gain is 1). Also, when comparing robust MPC with nominal MPC in a case study, the robust MPC controller has the same tuning as the nominal MPC controller, except for the additional tuning parameter of confidence level.

We can tune the MPC controllers for CSTR control system 1 with the above tuning guidelines, and tuning parameters are shown in Table 3-1.

Table 3-1 Tuning parameters for the MPC controllers for CSTR control system 1

Tuning Parameter	Value
Control horizon, n	8
Prediction horizon, p	20
Estimation horizon, $p\_$	20
Observer gain for $[x^T, e^T]$, L	$[1, 0.556]^T$
Weight for controlled variables in Q	10
Weight for controlled variables in R	0.1
Weight for controlled variables in W	1
Penalty on controlled variable violation in W_s	10^5
Confidence of each stochastic bound, α	99.9%

The deterministic SOCP subproblem solved by the proposed robust MPC method for this system has 8 decision variables, 92 linear constraints and 56 second order cones. This subproblem is typically solved in 0.02 CPU seconds. When applying the active set heuristic, the maximum number of SOCP subproblems solved for this system is 8 (the number of time steps in the control horizon), so the robust MPC costs at most $0.02\times8=0.16$ CPU seconds.

3.6.2.1 Set point tracking while observing bounds on controlled

In this study on control system 1, there is a set point change from 1.79 kmole/m^3 to 2.79 kmole/m^3 (toward the upper bound of the controlled variable) and then from 2.79 kmole/m^3 to 2.09 kmole/m^3 (away from the bound). Figure 3.9 shows the system dynamic behaviours with the nominal MPC, the robust MPC and the open-loop robust MPC in two situations: (a) Plant = Nominal Model CSTR-1, and (b) Plant=Plant CSTR-1.1.

We can see from Figure 3.9 (a) that the nominal MPC gives the best performance when there is no plant/model mismatch. The two robust MPC methods are more conservative when the set point is moved toward the bound, because they take actions that prevent the potential bound violation due to uncertainty. However, all the three methods give identical performance when the set point is moved away from the bound.

The conservativeness of the two robust MPC methods is advantageous when plant/model mismatch is present, as shown in Figure 3.9 (b). In this study when the set point is moved toward the bound, the y-bound is observed with the two robust MPC methods; however, it is violated with the nominal MPC. Again, all the three methods give identical performance when the set point is moved away from the bound.

Note that for both realizations, the open-loop robust MPC is unnecessarily more conservative than the robust MPC, because it does not consider the effect of the feedback in the prediction; therefore, it overestimates the uncertainty in the future.

One hundred simulations of the closed-loop system during time the first part of the transient (0-9 minutes) with the three MPC methods have been run with Monte Carlo sampling of the plant realizations. The results are summarized in Table 3-2. We observe that the nominal MPC gives the most aggressive control, which results in bound violations in many scenarios. The robust MPC gives more conservative control, and it

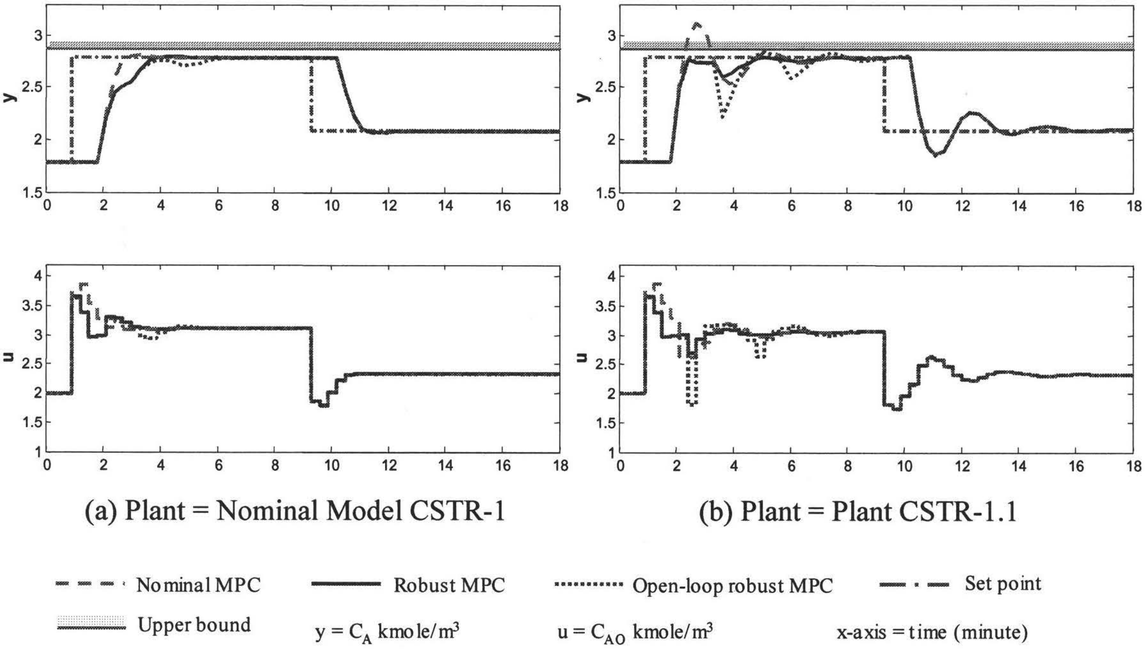


Figure 3.9 Set point tracking while observing bounds on manipulated variables – CSTR control system 1

Table 3-2 Monte-Carlo Simulation Results of case study in Figure 3.9 during 0-9 min

	Average IAE (1)	Average IAV (2)	Maximum violation	Samples with violation/total samples
Nominal MPC	5.1727	0.1940	0.1940	44/100
Robust MPC	5.3475	0.0006	0.0115	2/100
Open-loop robust MPC	5.4834	0	0	0/100

Note: (1) IAE denotes Integrated Absolute Error. (2) IAV denotes Integrated Absolute Violation.

observes the bound for 98% scenarios. The open-loop robust MPC is even more conservative, so it prevents violation for all the 100 plant realizations but provides poorer control as measured by the IAE.

3.6.2.2 Driving controlled variables back to feasible region

In this study using control system 1, we assume that the controller is initially off and the controlled variable is outside its feasible region (smaller than its lower bound). Then, the controller is switched on after 1 controller execution time step (i.e., at 0.3 minute), and it regulates the controlled variable back to the set point in the feasible region. Figure 3.3 shows the system dynamics with the nominal MPC, the robust MPC and the open-loop robust MPC in two situations: (a) Plant = Nominal Model CSTR-1, and (b) Plant = Plant CSTR-1.2.

We can see from Figure 3.10 (a) for the case with no model mismatch that the nominal MPC drives the controlled variable back to its feasible region fast while giving the best dynamic performance. The two robust MPC methods are more aggressive; they return the controlled variables to feasible region quickly for not only the nominal plant realization, but also all other probable plant realizations.

The advantage of the aggressiveness of the two robust MPC methods is apparent when plant/model mismatch is considered in Figure 3.10 (b). In this situation, the nominal MPC takes about four minutes to drive the controlled variable back to feasible region while the two robust MPC methods spend only 1 minute to do the same thing. For both situations, the open-loop robust MPC is more conservative than the robust MPC.

One hundred simulations of the closed-loop system with the three MPC methods have been run with Monte Carlo sampling of the plant realizations. The results are summarized in Table 3-3. The two robust MPC methods drive the controlled variable quickly back to the feasible region, and to achieve this result, they are more aggressive than required for most realizations. Therefore, their IAE is relatively high, although the robust MPC is better than the open-loop robust MPC. In contrast, the nominal MPC is not aggressive enough and could take as long as 3.9 minutes to return the controlled variable to the feasible region.

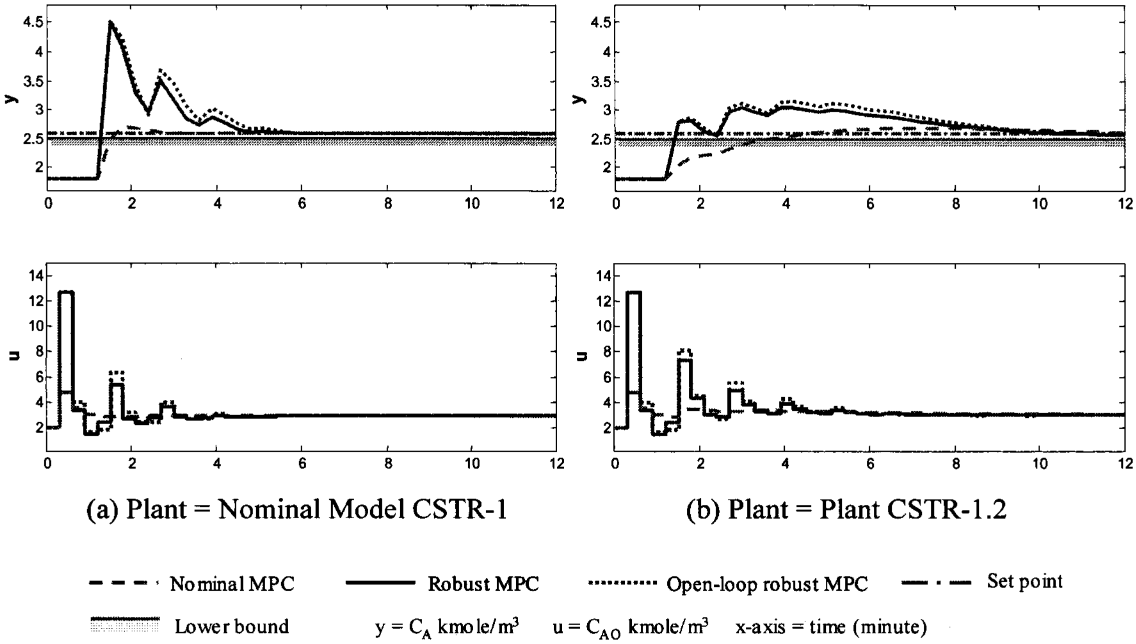


Figure 3.10 Return to feasible region – CSTR control system 1

Table 3-3 Monte-Carlo Simulation Results of case study in Figure 3.10

	Average IAE [1]	Average time back to feasible region (min)	Maximum time back to feasible region (min)
Nominal MPC	4.4180	1.6	3.9
Robust MPC	11.1919	1.2	1.5
Open-loop robust MPC	11.9372	1.2	1.5

Note: (1) IAE denotes Integrated Absolute Error.

3.6.2.3 Set point tracking while observing the hard input bounds

This case study shows the importance of addressing the input saturation in the prediction in robust MPC. We compare the performance of the robust MPC and the robust MPC whose algorithm does not consider manipulated variable (input) saturation in the closed-loop prediction model. The latter method assumes the inner optimization problem in the bilevel problem (3.5) is unconstrained, so the active set heuristic is not implemented and only one SOCP problem needs to be solved at each controller execution period. Thus, the latter method has unconstrained inner problems (but the outer problems are still constrained) and we call it unconstrained robust MPC (UCRMPC) here.

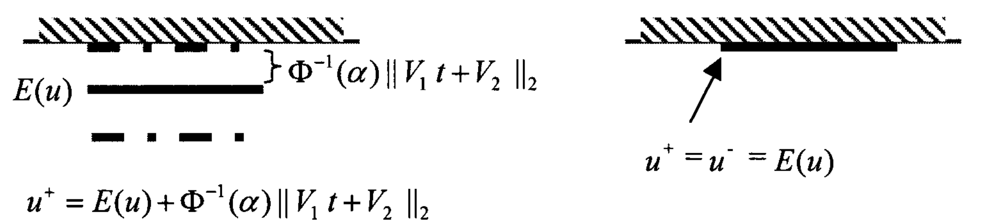
Naturally, these two methods are identical when saturation does not occur; but they could have significant difference when saturation occurs. To understand this conceptually, let's rewrite the constraints on a manipulated variable (equation (3.56b)) in the following simplified form,

$$u^{+} = E(u) + \Phi^{-1}(\alpha) \| V_1 \, t + V_2 \|_2 \leq u_{\max} \tag{3.62}$$

where the left-hand-side of the constraint denotes the maximum value of the uncertain manipulated variable u (u^{+}), which is the sum of the expected value of u ($E(u)$) and the effects of the uncertainties $\Phi^{-1}(\alpha) \| V_1 \, t + V_2 \|_2$). If the robust MPC does not address input saturation in the closed-loop prediction, the predicted u will be different for different plant realizations and the effects of the uncertainties $\Phi^{-1}(\alpha) \| V_1 \, t + V_2 \|_2$ will not be zero. So the constraint (3.62) only enforces the maximum value of the uncertain u to its upper bounds, and it keeps other realizations of u away from the bound. Figure 3.11 (a) illustrates this situation. The proper robust MPC (proposed in this work) addresses input saturation explicitly in the closed-loop prediction. When the active set heuristic decides a variable is at its bound, u will be constant for all plant realizations and the effects of the uncertainties $\Phi^{-1}(\alpha) \| V_1 \, t + V_2 \|_2$ will be zero. So the constraint (3.62) enforces the u for all the plant realizations (including the maximum u value, u^{+}, and the minimum u value, u^{-}) to its upper bound. Figure 3.11 (b) illustrates this situation.

The advantage of addressing input saturation in the closed-loop prediction is demonstrated in the case study shown in Figure 3.12. In this case study, the feed flow rate is constant at F=1 m^3/min, i.e., the plant is the Nominal Model CSTR-1. Two situations are simulated, and their results are shown in Figure 3.12 (a) and (b), respectively. In the first situation shown in Figure 3.12 (a), there is a set point step decrease and then a set point step increase. When the set point decreases, u moves away from its bound and the robust MPC and the unconstrained robust MPC give the same performance. When the set point increases, y cannot be driven to the set point because of the upper bound on u. So the robust MPC forces u to remain at its upper bound, and y is driven as close as possible to its set point quickly. However, the unconstrained robust MPC moves u slowly to its upper bound, and thus, y is driven slowly towards its set point. This is because the algorithm requires that all realizations of the manipulated variable remain feasible, which moderates the aggressiveness of the manipulated variable when it approaches a constraint.

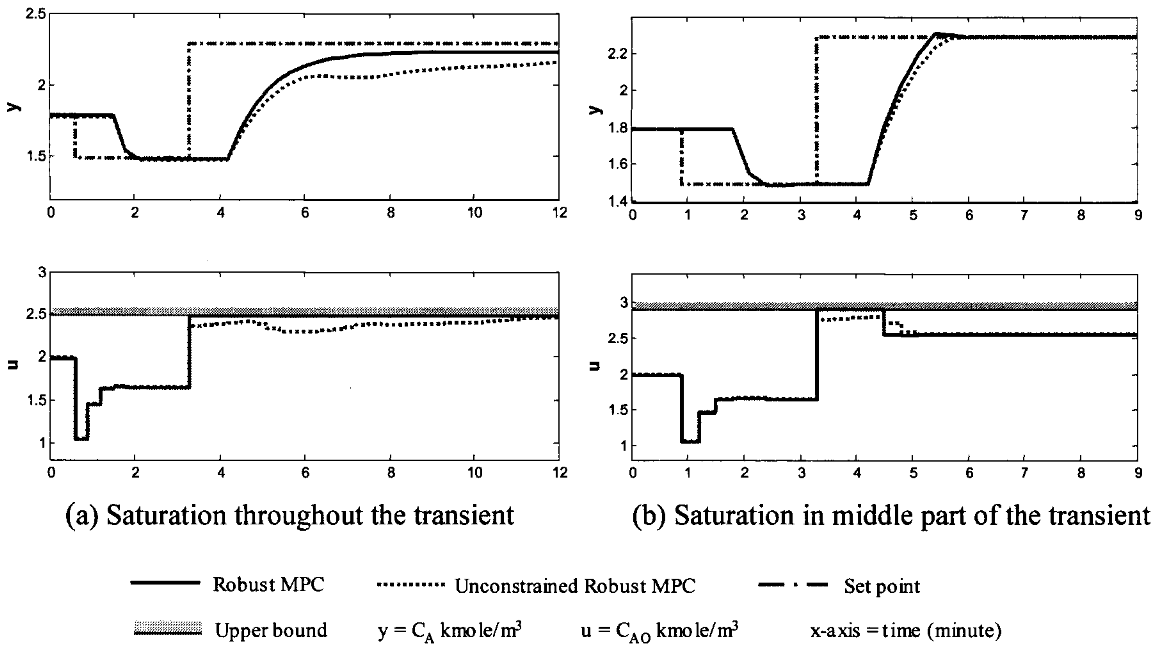
In the second situation shown in Figure 3.12 (b), the upper bound on u is not so tight, and the set point can be reached at the steady state. The robust MPC allows the input saturation during the transient to drive y quickly to the set point. However, the unconstrained robust MPC keeps u away from its upper bound during the transient, so y is driven to the set point slower.



(a) Prediction without addressing saturation (b) Prediction addressing saturation

Maximum or minimum value — Expected value — Upper bound

Figure 3.11 Conceptual comparison of robust MPC predictions for input saturation



(a) Saturation throughout the transient (b) Saturation in middle part of the transient

Figure 3.12 Observing hard input bounds – CSTR control system 1

3.6.3 CSTR control system 2

The second control system involves the CSTR process from the Appendix C of Marlin (2000), page 897-908. The controlled variable is still the outlet concentration of A (C_A) and the manipulated variable the inlet feed concentration of A (C_{A0}). However, there is no temperature controller to maintain the temperature of the reactor, so the system is non-isothermal. The cooling flow rate F_c is measured at the beginning of each controlled execution period, and its measurement is sent to the MPC controller as a measured disturbance. Figure 3.13 shows the diagram of this control system.

The non-linear plant model is given by the following equations,

Mass balance:

$$Vol\frac{dC_A}{dt} = F(C_{A0} - C_A) - Vol \cdot k_{A0}e^{-E/RT}C_A \tag{3.63}$$

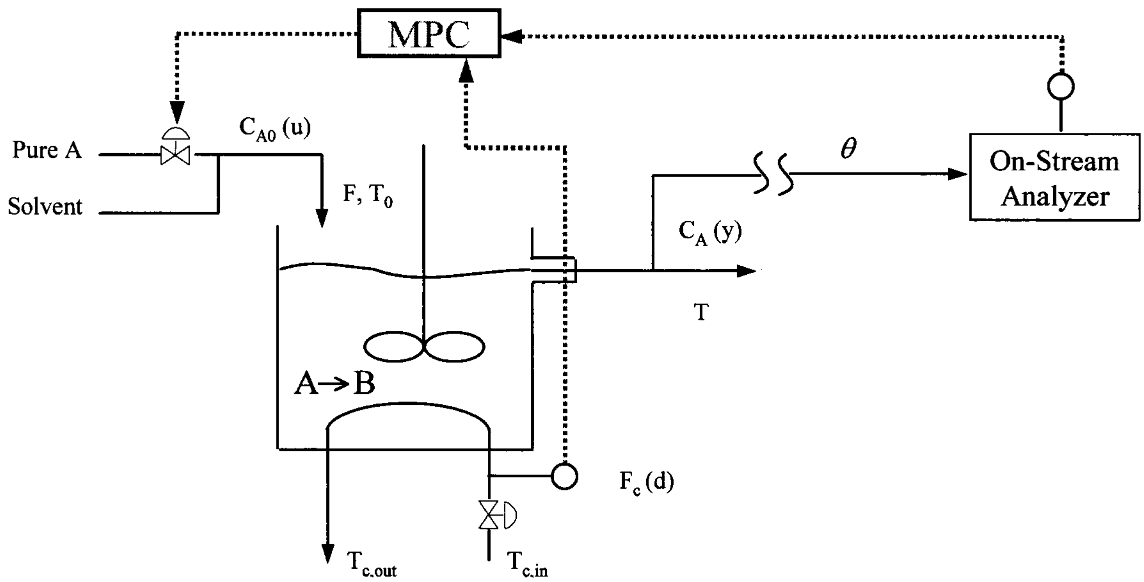


Figure 3.13 CSTR control system 2

where $k_{A0}e^{-E/RT}$ gives the reaction rate K_A that obeys first order Arrhenius equation.

<u>Energy balance</u>:

$$Vol \cdot \rho C_p \frac{dT}{dt} = \rho C_p F(T_0 - T) - \frac{aF_c^{b+1}}{F_c + \frac{aF_c^b}{2\rho_c C_{pc}}}(T - T_{c,in}) - \Delta H_{rxn} Vol \cdot k_{A0} e^{-E/RT} C_A \tag{3.64}$$

where ρ and C_p are the density and specific heat capacity of the mixture in reactor, ρ_c and C_{pc} are the density and specific heat capacity of the coolant, a, b denote the coefficients of the heat transfer during the cooling procedure, T_0 denotes the temperature of the inlet flow, $T_{c,in}$ denotes the temperature of the inlet cooling flow, $\Delta H_{rxn}<0$ denotes the enthalpy change due to the (exothermic) reaction.

This non-linear model can be linearized at a particular operating point and expressed as a transfer function. See Appendix F for the details of the linearization procedure and the parameters and operating points used in the thesis. The uncertainty of the system comes from the slowly varying inlet cooling flow temperature $T_{c,in}$, whose uncertain value is assumed to obey normal distribution with mean 310 K and standard deviation 5 K. The nominal plant model is derived at $T_{c,in}$ =310 K as,

Nominal Model CSTR-2:

$$y(s) = \frac{(s+0.8078)e^{-0.9s}}{s^2+1.925s+1.143}u(s) + \frac{0.02204e^{-0.9s}}{s^2+1.925s+1.143}d(s) \tag{3.65}$$

where y is the controlled variable C_A, u is the manipulated variable C_{A0}, d is the measured disturbance F_c. The time delay $\theta = 0.9$ minutes denotes the time for the output flow to reach the remote component analyzer.

In this case study, we will evaluate performance at two mismatch plant realizations. One is with $T_{c,in}$=318.95 K, and its model is:

Plant CSTR-2.1:

$$y(s) = \frac{(s+0.8078)e^{-0.9s}}{s^2+1.925s+1.143}u(s) + \frac{0.0126e^{-0.9s}}{s^2+1.925s+1.143}d(s) \tag{3.66}$$

The other is with $T_{c,in}$= 300.36, and its model is:

Plant CSTR-2.2:

$$y(s) = \frac{(s+0.8078)e^{-0.9s}}{s^2+1.925s+1.143}u(s) + \frac{0.0322e^{-0.9s}}{s^2+1.925s+1.143}d(s) \tag{3.67}$$

The controller execution period for this system is selected to be 0.3 minutes, so the models of the system are discretized with sampling time of 0.3 minutes. The state-space form of the models *without time delays and feedback variables* has the state vector x with 2 elements, and the system is controllable and observable. The feedback scheme assumes the unmeasured disturbance enters the system through the disturbance channel, which introduces the unmeasured disturbance vector e with 1 element. So the augmented system with x and e has 3 states, and it is detectable. The time delays between

y and u, d are described by 3 additional states using the method introduced in Appendix C. Since these 3 states denote the u and d in the last 3 time steps, they are naturally known and no observer gains are need for them.

The MPC controllers for this system have the same tuning parameters (except that NMPC does not have a confidence value for stochastic bounds), which are obtained using the method for the CSTR control system 1. Table 3-4 shows these parameters.

The deterministic SOCP subproblem solved by the proposed robust MPC method for this system has 10 decision variables, 115 linear constraints and 70 second order cones. This problem is typically solved in 0.02 CPU seconds. When applying the active set heuristic, the maximum number of SOCP subproblems solved for this system is 10 (the number of time steps in the control horizon), so the robust MPC costs at most 0.02×10=0.2 CPU seconds.

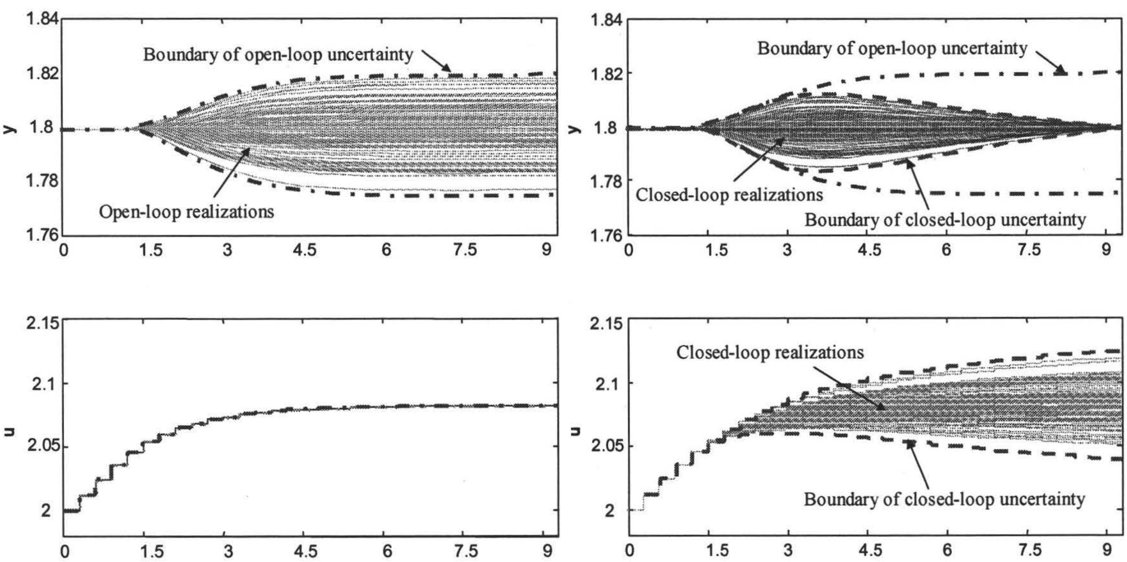
Since the unmeasured disturbance variable $T_{c,in}$ affects the dynamics between the measured disturbance variable and the controlled variable, we need to address the

Table 3-4 Tuning parameters of the MPC controllers for CSTR control system 2

Tuning Parameter	Value
Control horizon, n	10
Prediction horizon, p	25
Estimation horizon, $p\_$	25
Observer gains for $[x^T, e^T]^T$, L	$[1, -12.902, 18.983]^T$
Weight for controlled variables in Q	10
Weight for controlled variables in R	0.1
Weight for controlled variables in W	1
Penalty on controlled variable violation in W_s	10^5
Confidence of each stochastic bound, α	99.9%

uncertainty in the disturbance model used explicitly in the robust MPC. As we discussed previously in this chapter, closed-loop uncertainty should be used in the uncertainty prediction. Figure 3.14 illustrates the open-loop uncertainty and closed-loop uncertainty after a disturbance step change of 1.5 m^3/min. The open-loop uncertainty shown in Figure 3.14 (a) is the uncertainty based on assuming no feedback correction occurs for disturbance plant/model mismatch in the future. So the future u series is deterministic and the future y series is different for different plant realizations. The closed-loop uncertainty shown in Figure 3.14 (b) is the uncertainty including the effect of the future feedback corrections on the closed-loop system. So, the both the future u series and y series are different for different plant realizations.

In Figure 3.14, the dashed dotted lines and the dashed lines show the boundaries of the open-loop and closed-loop uncertainty regions respectively with 99% confidence, which are obtained with the method shown in the Subsection 3.3.1. We observe that the closed-loop uncertainty region is smaller than the open-loop uncertainty region.



(a) Open-loop uncertainty (b) Closed-loop uncertainty

Setpoint $y = C_A$ kmole/m^3 $u = C_{AO}$ kmole/m^3 x-axis = time (minute)

Figure 3.14 Open-loop and closed-loop uncertainty – CSTR control system 2

Therefore, using closed-loop uncertainty in robust MPC prediction for disturbance plant/model mismatch is not only conceptually more accurate, but also less conservative.

3.6.3.1 Constant disturbance rejection

In this study with the CSTR control system 2, there is a step change in the measured disturbance F_c from 15 m^3/min to of 7.5 m^3/min. Figure 3.15 shows the system dynamics with the nominal MPC and the robust MPC in two situations: (a) Plant = Nominal Model CSTR-2, and (b) Plant = Plant CSTR-2.1. Figure 3.15 (a) shows that the nominal MPC compensates the disturbance change perfectly while the robust MPC drives the controlled variable away from its upper bound to prevent potential violation for some realizations. When the plant/model mismatch is present in the second situation shown in Figure 3.15 (b), the controlled variable violates its upper bound with the nominal MPC, while the bound is observed with the robust MPC.

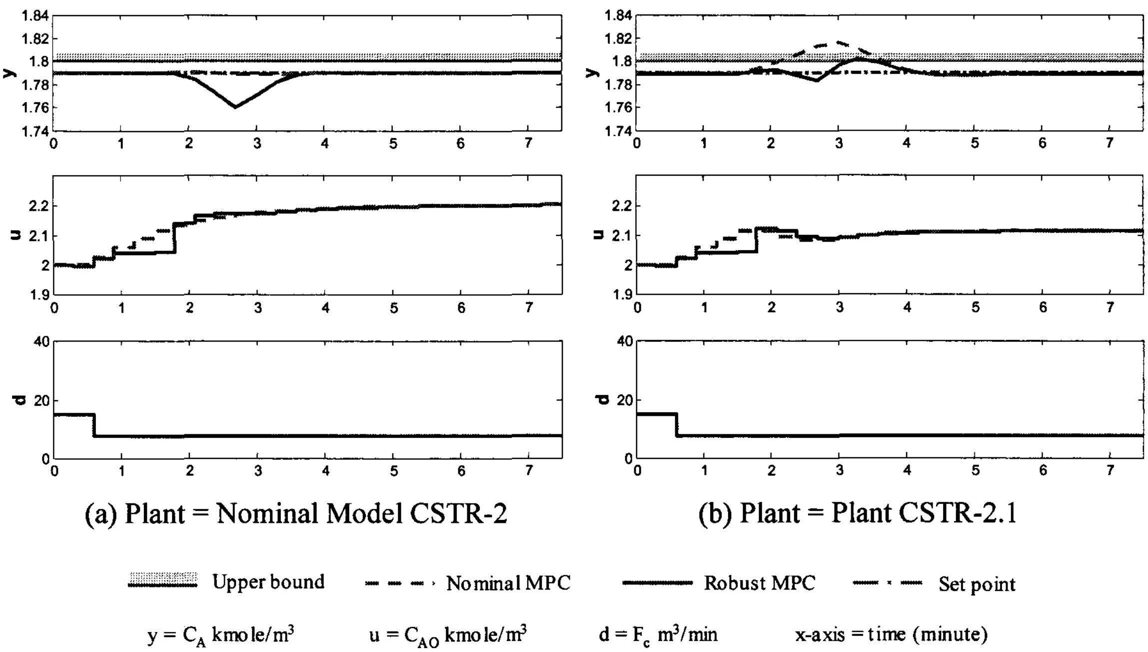


Figure 3.15 Constant disturbance rejection – CSTR control system 2

Table 3-5 Monte-Carlo Simulation Results of case study in Figure 3.15

	Average IAE [(1)]	Average IAV [(2)]	Maximum violation	Samples with violation/total samples
Nominal MPC	0.0590	0.0062	0.0296	18/100
Robust MPC	0.0999	0.0007	0.0131	3/100

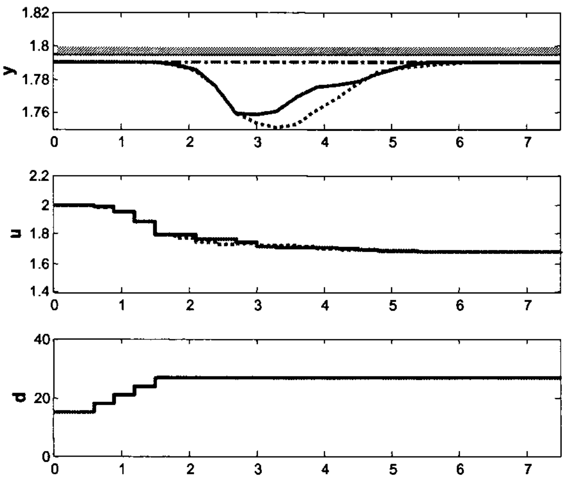
Note: (1) IAE denotes Integrated Absolute Error. (2) IAV denotes Integrated Absolute Violation.

Table 3-5 summarizes 100 simulation results with Monte Carlo sampling of plant realizations. The robust MPC experiences far fewer constraint violations, and to achieve this good feasibility performance, it incurs a larger IAE. If the set point were far from a constraint, both controllers would provide the same performance, as measured by IAE.

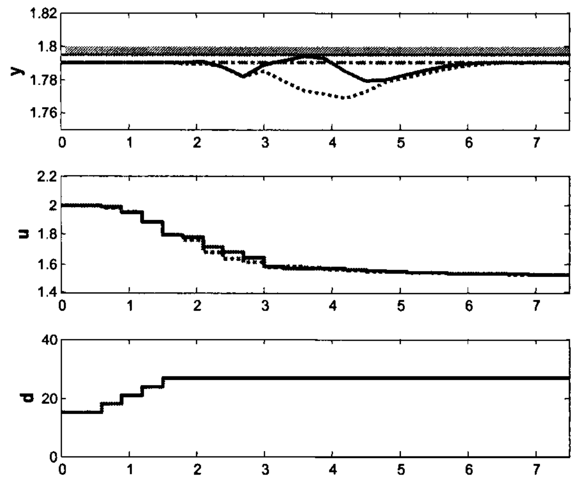
3.6.3.2 Stairs disturbance rejection

In this study with the CSTR control system 2, the measured disturbance increases by 3 m^3/min during three successive controller execution periods. Figure 3.16 shows the system dynamics with the robust MPC and the open-loop robust MPC in two situations: (a) Plant = Nominal Model CSTR-2, and (b) Plant = Plant CSTR-2.2. In both cases, the open-loop robust MPC is more conservative.

Table 3-6 summarizes the simulation results of the closed-loop system with Monte Carlo sampling of 100 plant realizations. We can see that the robust MPC ensures the bound satisfaction at a high confidence level (97%) while it achieves much better dynamic performance than the open-loop robust MPC does. This demonstrates that the open-loop prediction of the disturbance plant uncertainty is overly conservative for a robust MPC formulation.



(a) Plant = Nominal Model CSTR-2



(b) Plant = Plant CSTR-2.2

Upper bound — Robust MPC ········ Open-loop robust MPC — · — Set point

$y = C_A$ kmole/m^3 $u = C_{AO}$ kmole/m^3 $d = F_c$ m^3/min x-axis = time (minute)

Figure 3.16 Stairs disturbance rejection – CSTR control system 2

Table 3-6 Monte-Carlo Simulation Results of case study in Figure 3.16

	Average IAE $^{(1)}$	Average IAV $^{(2)}$	Maximum violation	Samples with violation/total samples
Robust MPC	0.1929	0.0005	0.0122	3/100
Open-loop robust MPC	0.2496	0.0001	0.0053	1/100

Note: (1) IAE denotes Integrated Absolute Error. (2) IAV denotes Integrated Absolute Violation.

3.6.3.3 Periodic disturbance rejection

In this study with the CSTR control system 2, the measured disturbance changes periodically from 3 m^3/min to -3 m^3/min. Figure 3.17 shows the system dynamics with the nominal MPC, the robust MPC and the open-loop robust MPC for Plant = Plant

CSTR-2.2. Again, we can see the nominal MPC is the most aggressive control method, and the controlled variable violates the constraint periodically during the transient. The robust MPC gives more conservative control according to its explicit consideration of uncertainty, so it prevents the constraint violation. The open-loop robust MPC prevents the constraint violation but it is more conservative than the robust MPC.

Table 3-7 summarizes 100 simulation results of the closed-loop system with Monte Carlo sampling of the plant realizations. Again, the robust MPC observes the constraints for almost all the plant realizations, and to achieve this behavior, it achieves a somewhat higher IAE. The nominal MPC achieves lower IAE, but it results in constraint violation for many plant realizations. The open-loop robust MPC observes the constraints for all the plant realizations, but it achieves the largest IAE.

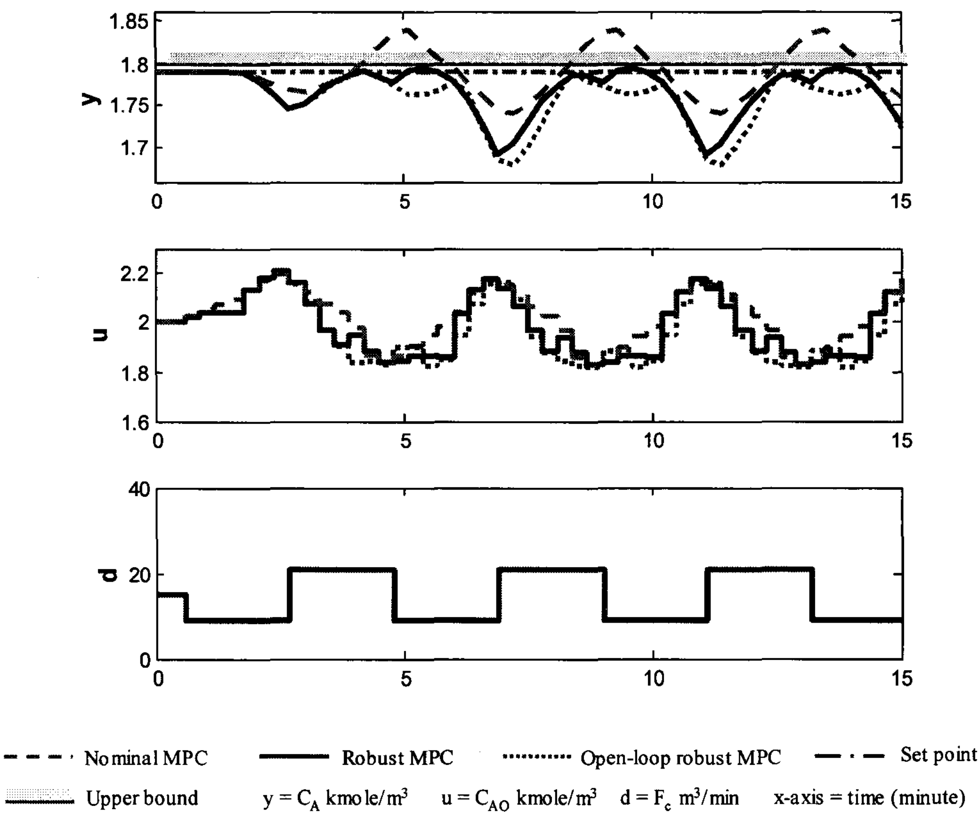


Figure 3.17 Seasonal disturbance rejection - CSTR control system 2

Table 3-7 Monte-Carlo Simulation Results of case study in Figure 3.17

	Average IAE (1)	Average IAV (2)	Maximum violation	Samples with violation/total samples
Nominal MPC	0.3362	0.0397	0.0296	47/100
Robust MPC	1.1394	0.0002	0.0021	1/100
Open-loop robust MPC	1.4684	0	0	0/100

Note: (1) IAE denotes Integrated Absolute Error. (2) IAV denotes Integrated Absolute Violation.

3.6.3.4 Disturbance rejection with different state estimation methods

This case study with CSTR control system 2 shows the importance of including the uncertainty in state estimation in the robust MPC formulation. We compare the performance of the robust MPC with traditional state estimation not addressing uncertainty explicitly (which solves the SOCP problem (3.32)) and the robust MPC with state estimation addressing uncertainty explicitly (which solves the SOCP problem (3.45). Figure 3.18 compares the system dynamics under the robust MPC using the two different state estimation methods respectively for the situation in which Plant = Plant CSTR-2.2 and the measured disturbance decreases from 15 m^3/min to 9 m^3/min and then increases to 21 m^3/min. We can see that the controlled variable violates its upper bound if uncertainty in state estimation is not addressed and the bound is not observed when uncertainty in state estimation is addressed.

The advantage of addressing uncertainty in state estimation lies in the incorporation of greater (and more accurate) uncertainty in the closed-loop prediction of the controlled variables. To explain this conceptually, let's rewrite the constraints on a controlled variable y (equation (3.43d) without addressing uncertainty in state estimation

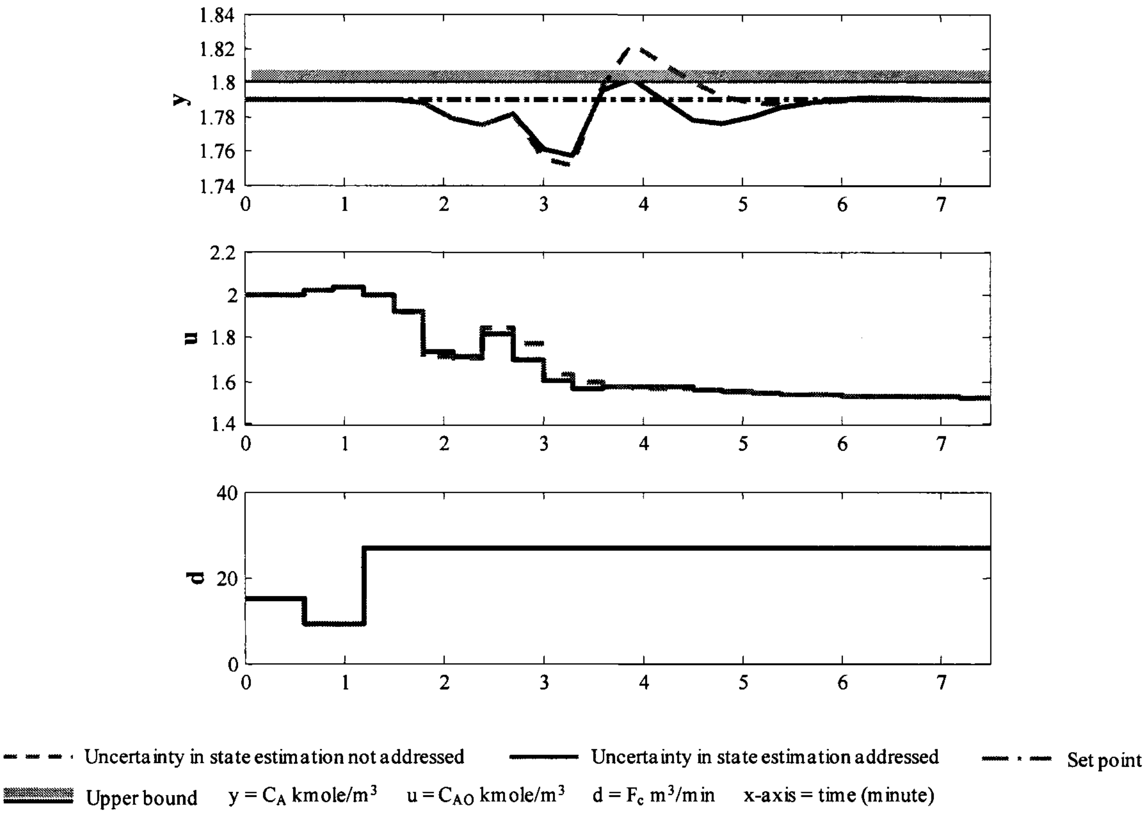


Figure 3.18 RMPC using different state estimation methods – CSTR control system 2

or equation (3.56d) addressing uncertainty in state estimation) in the following simplified form,

$$y^{+} = E(y) + \Phi^{-1}(\alpha) \| V_1 t + V_2 \|_2 \leq y_{\max} \tag{3.68}$$

where the left-hand-side of the constraint denotes the maximum value of the uncertain controlled variable y (y^{+}), which is the sum of the expected value of y ($E(y)$) and the effects of the uncertainties ($\Phi^{-1}(\alpha)\|V_1t+V_2\|_2$). If the robust MPC includes the uncertainties in the state estimation, the norm term $\Phi^{-1}(\alpha)\|V_1t+V_2\|_2$ will be bigger than that without the uncertainties in the state estimation. Thus, the decisions will be more conservative if uncertainty in state estimation is also included, so that the constraint violation will be less probable to occur.

The above conceptual discussion is validated by the simulation results shown in Figure 3.19. This figure compares the maximum values of the norm term of the output over the prediction horizon with and without including uncertainty in state estimation, at each controller execution period. We can find that this norm term is bigger when uncertainty in state estimation is addressed than that when uncertainty in state estimation is not addressed, which explains the more appropriate control when including uncertainty in state estimation in Figure 3.18.

One hundred simulations have been run for the same disturbance scenario and with Monte Carlo sampling of different plant realizations. The controlled variable-bound is observed for 83% of the plant realizations when uncertainty in state estimation is not addressed and 97% when it is addressed, with the same robust MPC controller. This again demonstrates the importance of including the uncertainty in state estimation in the robust MPC formulation explicitly.

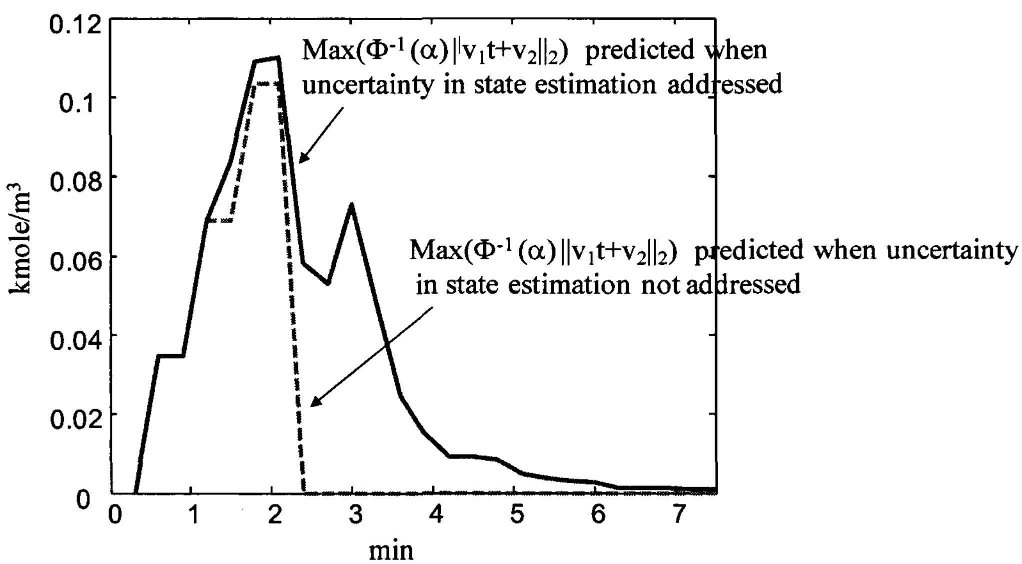


Figure 3.19 Robust MPC predictions with different state estimation methods – CSTR control system 2

3.7 Conclusions

In this chapter, a new robust MPC method has been developed for feedforward and feedback control of uncertain dynamic systems. The presentation has followed the organization in the "roadmap" in Figure 3.1. This method primarily addresses robust feasibility for MPC with model uncertainty, and it possesses the following characteristics:

1) A framework that addresses:

    a. Bilevel optimization – The original bilevel optimization problem was reformulated as a single-level problem by replacing the inner optimization with its optimality conditions;
    b. Correlated parametric uncertainty – The uncertainty from the plant, the measured disturbance(s) and the stochastic unmeasured disturbances/noises. A novel deviation model formulation, which is obtained by the deviation of the variables from a virtual steady state of them (determined by the latest implemented manipulated variables), is used to reduce the conservativeness in the prediction of time-invariant uncertainty by limiting the effects of plant uncertainty to *changes* in the input variables;
    c. Hard bounds on manipulated variables in closed-loop prediction – The saturation pattern of the manipulated variables is obtained through an active set heuristic in an iterative way;
    d. Tractable solution for real-time implementation – The method solves a limited number of (convex) SOCPs, which can be solved by the stat-of-the-art interior point optimizers.

    We note that other researchers have addressed some of these issues, but no published method provides a controller that addresses all of these issues.

2) A novel deviation model formulation obtained by the deviation of the variables from a virtual steady state of them (determined by the latest implemented manipulated

variables). This formulation is used for time-invariant uncertainty to reduce the conservativeness in uncertainty prediction by limiting the effects of plant uncertainty to *changes* in the input variables.

3) Efficient real-time calculation for uncertainty characterization, where the extensive calculation is performed off-line.

4) Explicit handling of uncertainties in state-estimation.

The new robust MPC method can be used to optimize uncertain systems with feedback. Its advantages are demonstrated in simulation case studies. We can conclude from the simulation results that:

1) The robust MPC outperforms the nominal MPC on handling the constraints on controlled variables;
2) The robust MPC, which uses a closed-loop uncertainty estimate, is better than the robust MPC with open-loop prediction of uncertainty, which could be unnecessarily conservative due to its overestimation of uncertainty;
3) The robust MPC handles the saturation of the manipulated variables well (but without a global optimality guarantee);
4) The feasibility of the plant can be achieved with a high probability though chance-constrained programming for the robust MPC formulation (provided a feasible plant trajectory exists). The probability of constraint violations can be reduced by increasing the confidence level for each constraint as well as increasing the uncertainty of the parameters.

The new robust MPC method can be applied to process control and supply chain optimization problems. The following Chapters 4 and 5 will discuss special issues in these two types of applications and introduce special extensions in modeling and formulation for the unique needs of each application.

Chapter 4

Robust MPC for Process Control

In this chapter, we extend the general framework of the robust MPC developed in Chapter 3 to include features required for the process control problems. Two extensions are developed: i) integrating the robust steady-state optimization, and ii) including the robust dynamic performance in the objective function.

Industrial MPC control systems usually include a steady-state optimization unit that is executed immediately before each controller execution (Qin and Badgwell, 2003). Since a nominal steady-state optimization may give infeasible set points with the presence of uncertainty, a new robust steady-state optimization method that is developed to address the closed-loop uncertainty explicitly is introduced in Section 4.1. The original formulation is a bilevel stochastic optimization problem, which is then approximately transformed into a limited number of convex optimization problems for efficient real-time calculation. This reformulation is similar to the one introduced in Chapter 3 for dynamic feedback systems.

In some process control problems, we would like to optimize the dynamic performance based on the behaviors of all realizations of the uncertain system instead of just the nominal dynamic performance. Section 4.2 discusses including the expected performance and the *variances* of the controlled variables in the objective function, and it also shows that the new objective function is still convex and quadratic, yielding tractable real-time calculations.

Section 4.3 reports the case study results that show the advantages of the robust MPC method and the two extensions in process control, and the conclusions are summarized in Section 4.4.

4.1 Robust Steady-State Optimization

4.1.1 Steady-state optimization in industrial MPC control system

As introduced in Chapter 1, a typical industrial MPC control system involves a steady-state optimization unit that obtains optimal set points and a trajectory optimization unit (i.e., MPC controller) that determines the controller action to regulate the dynamics of the system. Figure 4.1 shows the simplified diagram of the closed-loop system that includes both the steady-state and trajectory optimizations.

The steady-state optimization is executed immediately before every trajectory optimization at the beginning of each controller execution period. It is formulated to find a feasible "settling point" or steady state of the system that is close to the reference values

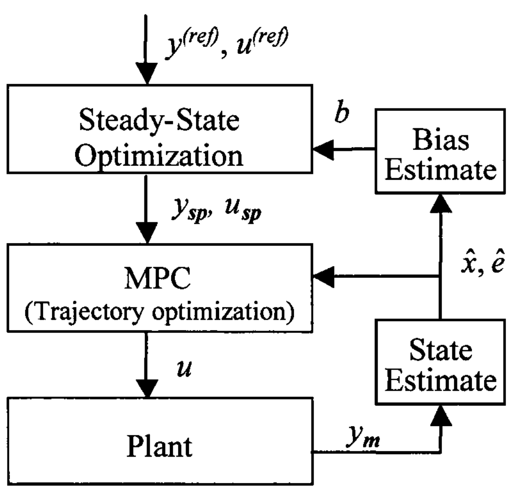


Figure 4.1 Closed-loop system with steady-state and trajectory optimization

of the controlled and manipulated variables $y^{(ref)}$, $u^{(ref)}$, which are reference values determined by an upper-level optimizer or by plant personnel. The results of the steady-state optimization are the set point of the system (y_{sp}, u_{sp}), which are used by the MPC trajectory optimization as described in Chapter 3.

The steady-state optimization is important because disturbances entering the system or new input information from the operator may change the location of the optimal steady state. It is performed based on the steady-state plant model, which can be obtained from the state-space dynamic model (equations (3.5b-3.5d) in Chapter 3) as

$$x_{ssr} = A_r x_{ssr} + B_r u_{ssr} + B_{dr} d_{mss} + B_{er} e_{ss} \tag{4.1}$$

$$y_{ssr} = C_r x_{ssr} \tag{4.2}$$

where $y_{ssr} \in R^{n_y}$, $x_{ssr} \in R^{n_x}$, $u_{ssr} \in R^{n_u}$, $d_{mss} \in R^{n_d}$, $e_{ss} \in R^{n_e}$ denote the vectors containing the controlled variables, state variables, manipulated variables, measured or forecast disturbances and feedback variables at the steady state respectively, A_r, B_r, B_{dr}, B_{er} and C_r are the uncertain parameters of the plant. Here, we do not consider the noises for steady-state optimization.

Since the steady-state measured disturbances d_{mss} and the steady-state feedback e_{ss} can not be obtained directly during the transient, we have to use estimated values for them in equation (4.1). In this thesis, we assume the disturbances measured at the current time step to be constant in the future and estimate d_{mss} by the current measured disturbance $d_{m,0}$. We also assume the feedback estimated at the current time step to be constant in the future and estimate e_{ss} by $\hat{e}_0$. Thus, the steady state model equations (4.1-4.2) can be combined into the following form.

$$y_{ssr} = K_r u_{ssr} + K_{dr} d_{m,0} + b_0 \tag{4.3}$$

where $K_r = C_r(I - A_r)^{-1} B_r$ and $K_{dr} = C_r(I - A_r)^{-1} B_{dr}$ are the uncertain gains and $b_0 = C_r(I - A_r)^{-1} B_{er}\hat{e}_0 \approx C(I - A)^{-1} B_e \hat{e}_0$, which means we approximate b_0 by its nominally estimated value.

The nominal steady-state optimization formulation can be written in the following form.

NSSO:

$$\min_{u_{ss},s} \quad c_y^T s_y + c_u^T s_u + f^T s \tag{4.5a}$$

$$\text{s.t.} \quad y_{ss} = K u_{ss} + K_d d_{m,0} + b_0 \tag{4.5b}$$

$$y_{sp} = y_{ss} \tag{4.5c}$$

$$u_{sp} = u_{ss} \tag{4.5d}$$

$$u_{\min} \le u_{ss} \le u_{\max} \tag{4.5e}$$

$$y_{\min} - s \le y_{ss} \le y_{\max} + s \tag{4.5f}$$

$$-s_y \le y_{ss} - y^{(ref)} \le s_y \tag{4.5g}$$

$$-s_u \le u_{ss} - u^{(ref)} \le s_u \tag{4.5h}$$

$$s_y, s_u, s \ge 0 \tag{4.5i}$$

The mathematical program (4.5a-4.5i) is a Linear Program (LP, Boyd and Vandenberghe, 2004). Equation (4.5b) denotes the nominal steady-state plant model (4.3) using the nominal value of all parameters. $y_{ss} \in R^{n_y}$ and $y_{ss} \in R^{n_u}$ are the nominal steady-state controlled and manipulated variables, and the nominal gains K and K_d can be calculated using the nominal values of the plant parameters. Equations (4.5c-4.5d) define results of the steady-state optimization as the set points $y_{sp} \in R^{n_y}$, $u_{sp} \in R^{n_u}$ to MPC controller.

Equation (4.5e) denotes the hard lower bounds $u_{min} \in R^{n_u}$ and upper bounds $u_{max} \in R^{n_u}$ on the manipulated variables. Equation (4.5f) denotes the soft lower bounds $y_{min} \in R^{n_y}$ and upper bounds $y_{max} \in R^{n_y}$ on the controlled variables with the slack variables s measuring any violation. Equations (4.5g-4.5h) define the slack variables $s_y \in R^{n_y}$ and $s_u \in R^{n_u}$ for the deviation of the steady-state controlled and manipulated variables from their reference values, respectively. Equation (4.5i) denotes all the slack variables are nonnegative. The objective function for the optimization (4.5a) minimizes the weighted deviation of the steady-state controlled and manipulated variables from their references plus the weighted violation of the soft bounds on the controlled variables. The weighing and penalty coefficients c_y, c_u and f can be determined according to economics or other preferences for the operation.

4.1.2 The robust steady-state optimization with closed-loop uncertainty

The purpose of the steady-sate optimization is to find the "best, feasible" settling condition for the control system. However, due to the uncertainty in the plant, the steady state of the uncertain plant is different for different plant realizations. Then, the set points y_{sp}, u_{sp} obtained by solving problem NSSO will likely not be the best values and the variables in the plant may not even be feasible. Therefore, a robust steady-state optimization method is developed in this research to address the uncertainty explicitly. This method is designed to achieve the following goals: the set points y_{sp} and u_{sp} give the maximum profit for the nominal model and feasible y and u for all model parameters within their uncertainty definition.

A straight-forward approach for formulating the robust steady-state optimization for the above two goals is to a) Replace the nominal model (4.5b) by the uncertain plant model (4.3) (so that for deterministic steady state manipulated variables, the steady state controlled variables are different for different plant realizations); b) Pose the constraint (4.5f) on the uncertain steady-state controlled variables instead of their nominal values.

This idea has been used in the robust steady-state optimization method developed by Kassmann et al. (2000). However, this approach does not address the effect of the controller on the steady state in the closed-loop system. As we know, the controller will compensate for model errors by adjusting the manipulated variables to minimize the objective function. These adjustments would tend to maintain feasibility and bring the controlled variables, y_{sp}, to their reference values, $y^{(ref)}$, if possible. So, the method of Kassmann et. al. is not correct for the uncertainty in the closed-loop system because it overlooks the effect of the controller. We call this method robust steady-state optimization with open-loop uncertainty.

To address the uncertainty of the steady state of closed-loop system correctly, we have to consider both the uncertain steady-state plant model (4.3) and a steady-state controller model. We call the method using this idea robust steady-state optimization with closed-loop uncertainty. Similar to our discussion in Chapter 3 for trajectory optimization, closed-loop uncertainty is more accurate and less conservative than open-loop uncertainty for steady-state optimization. This will be demonstrated by some case study results in Section 4.3.

Here, we develop the robust steady-state optimization method using closed-loop uncertainty. The steady-state MPC controller model is approximated by the steady-state version of nominal MPC (similar to equation (3.1) in Chapter 3) as

NMPCSS:

$$\min_{u_{ss}} \quad (y_{ss} - y_{sp})^T Q (y_{ss} - y_{sp}) + (u_{ss} - u_{sp})^T R (u_{ss} - u_{sp}) \tag{4.6a}$$

$$\text{s.t.} \quad y_{ss} = K u_{ss} + K_d d_{m,0} + b_{ssr} \tag{4.6b}$$

$$u_{\min} \le u_{ss} \le u_{\max} \tag{4.6c}$$

The objective function (4.6a) is simplified to contain only the deviations of the controlled and manipulated variables from their set points at the steady state because the system is invariant at the steady state throughout the horizon. Q and R contain the weighting coefficients from $\tilde{Q}$ and $\tilde{R}$ in NMPC equation (3.1) for only one time step. The nominal steady-state model (4.6b) is the steady-state version of the nominal dynamic model (3.1b-3.1d), which can be derived from the model (3.1b-3.1d) as we discussed in the previous Section 4.1.1. Note that the feedback term used by the controller in its model at steady state will be different for different plant realizations, so the bias $b_{ssr} \in R^{n_y}$ in the model (4.6b) is not exactly b_0, but an uncertain value depending on plant realization. Here, we only include the hard bounds on the manipulated variables (4.6c) because the soft bounds on the controlled variables will be enforced by the robust steady-state optimization in the outer layer of the bilevel problem.

Then, we can write the new robust steady-state optimization formulation in the following form

RSSO-CL:

$$\min_{y_{sp},u_{sp},s} \quad c_y^T s_y + c_u^T s_u + f^T s \tag{4.7a}$$

$$\text{s.t.} \quad u_{ssr} = NMPCSS(b_{ssr}) \tag{4.7b}$$

$$y_{ssr} = K_r u_{ssr} + K_{dr} d_{m,0} + b_0 \tag{4.7c}$$

$$b_{ssr} = K_r u_{ssr} + K_{dr} d_{m,0} + b_0 - K u_{ssr} - K_d d_{m,0} \tag{4.7d}$$

$$u_{sp} = u_{ss} \tag{4.7e}$$

$$y_{sp} = y_{ss} \tag{4.7f}$$

$$u_{\min} \le u_{ssr} \le u_{\max} \tag{4.7g}$$

$$y_{\min} - s \le y_{ssr} \le y_{\max} + s \tag{4.7h}$$

$$-s_y \le y_{ss} - y^{(ref)} \le s_y \tag{4.7i}$$

$$-s_u \le u_{ss} - u^{(ref)} \le s_u \tag{4.7j}$$

$$s_y, s_u, s \ge 0 \tag{4.7k}$$

For all K_r, K_{dr} in the uncertainty region

Equation (4.7b) denotes the steady-state nominal MPC control law (4.6), with the optimal manipulated variables u_{ssr} different for different estimated steady-state bias b_{ssr} due to different realizations of the plant. Equation (4.7c) relates the uncertain y_{ssr} to the uncertain u_{ssr} using the uncertain steady-state plant model, and equation (4.7c) relates the uncertain estimated steady-state bias to the uncertain plant realizations. Equations (4.7e-4.7f) define that nominal steady state values y_{ss}, u_{ss} that are sent to the MPC controller as set points y_{sp}, u_{sp}. Equations (4.7g-4.7h) enforce hard bounds on the uncertain manipulated variables u_{ssr} and soft bounds on the uncertain controlled variables y_{ssr}. Equations (4.7i-4.7j) define the slack variables s_y and s_u for the deviation of the nominal steady-state controlled and manipulated variables from their reference values, respectively. Equation (4.7k) denotes all the slack variables are nonnegative. The objective of the optimization (4.7a) minimizes the weighted deviation of the nominal steady-state controlled and manipulated variables from their references plus the weighted violation of the soft bounds on the controlled variables.

Obviously, RSSO-CL is a bilevel stochastic optimization problem. As we discussed in Chapter 3, we do not use the scenario-based uncertainty because of tractability; rather, we use a continuous parametric uncertainty description. However, the bilevel problem is very difficult to solve in the real-time. Again, we can transform this problem approximately into a limited number of single-level deterministic optimization problems using the similar idea we use for dynamic optimization in Chapter 3. We will discuss the details in the following sections.

4.1.3 The reformulation to convex optimization

The first step of the reformulation is to replace the inner optimization problem (4.7b) by its KKT conditions. If NMPCSS is properly tuned, the QP Problem will be strictly convex so that it's optimum can be uniquely determined through its first order KKT conditions as

$$2K^T Q(Ku_{ssr} + K_d d_{m,0} + b_{ssr} - y_{sp}) + 2R(u_{ssr} - u_{sp}) + \lambda^+ - \lambda^- = 0 \tag{4.8a}$$

$$\lambda^+ \cdot (u_{ssr} - u_{\max}) = 0 \,, \quad \lambda^- \cdot (-u_{ssr} + u_{\min}) = 0 \,, \quad \lambda^+, \lambda^{-1} \geq 0 \tag{4.8b}$$

$$u_{\min} \leq u_{ssr} \leq u_{\max} \tag{4.8c}$$

where the Lagrange multiplier vectors $\lambda^+, \lambda^- \in R^{n_u}$ relate to the upper and lower bounds on the manipulated variables respectively, the dot "·" denotes the element-wise multiplication. We can replace the nominal MPC control law $u_{ssr} = NMPCSS(b_{ssr})$ with the KKT conditions (4.8a-4.8c) so that the bilevel stochastic problem (4.7) becomes single level stochastic MPEC problem. However, this problem is still difficult to solve in the real-time due to the complementarity constraints (4.8b).

So the second step of reformulation is to remove the complementarity constraints (4.8b). Similar to what we have done for trajectory optimization, we assume the following assumption holds

Assumption 4.1: *A manipulated variable at steady state either equals its bound for all the realizations of the process or is unconstrained for all the realizations of the process.*

We also assume the all the active bounds on manipulated variables at the steady state are known (using a heuristic to be explained). When no bounds are active, all the Lagrange multipliers must be zero due to (4.8b), and equation (4.8a) becomes

$$2K^T Q(Ku_{ssr} + K_d d_{m,0} + b_{ssr} - y_{sp}) + 2R(u_{ssr} - u_{sp}) = 0 \tag{4.9}$$

According to equation (4.7d), equation (4.9) can be written as

$$u_{ssr} = (K^T Q K_r + R)^{-1}[K^T Q(y_{sp} - b_0 - K_{dp} d_{m,0}) + Ru_{sp}] \tag{4.10}$$

Now we define a new, artificial vector

$$t_{ss} = K^T Q(y_{sp} - b_0) + Ru_{sp} \tag{4.11}$$

Note that the nominal MPC can be properly tuned so that the weighting matrix R has full rank, so that for any value of t_{ss} there will be (perhaps several sets of) values of y_{sp}, u_{sp} which gives the same value of t_{ss} through equation (4.11); or for any value of y_{sp}, u_{sp} there will be unique value of t_{ss} corresponds to it through equation (4.11). Then, equation (4.10) can be equivalently transformed into the following unconstrained steady-state nominal MPC control law.

$$u_{ssr} = (K^T Q K_r + R)^{-1} t_{ss} - (K^T Q K_r + R)^{-1} K^T Q K_{dr} d_{m,0} \tag{4.12}$$

When a steady-state manipulated variable (an element in u_{ssr}) saturates, the corresponding Lagrange multiplier forces it to its bound and the multiplier value does not affect other manipulated variables. Therefore, the Lagrange multipliers can also been omitted in the formulation when we know the active set. We can address the *known* saturation of the manipulated variables by modifying equation (4.12) into give the following.

$$u_{ssr} = I_{\delta,ss}\left((K^T Q K_r + R)^{-1} t_{ss} - (K^T Q K_r + R)^{-1} K^T Q K_{dr} d_{m,0}) - t_{ss}\right) + t_{ss} \tag{4.13}$$

$$(I - I_{\delta,ss}) t_{ss} = u_c \tag{4.14}$$

where $I_{\delta,ss} \in R^{n_u \times n_u}$ is a diagonal matrix with the diagonal vector containing 0 or 1 to specify the saturation, $I \in R^{n_u \times n_u}$ is an identity matrix. The vector u_c contains the active upper bound or lower bound, which is known when we know the saturation pattern.

According to equations (4.7c) and (4.13),

$$\begin{aligned} y_{ssr} &= K_r u_{ssr} + K_{dr} d_{m,0} + b_0 \\ &= K_r \left[I_{\delta,ss}\left((K^T Q K_r + R)^{-1} t_{ss} - (K^T Q K_r + R)^{-1} K^T Q K_{dr} d_{m,0}) - t_{ss}\right) + t_{ss} \right] \\ &\quad + K_{dr} d_{m,0} + b_0 \end{aligned} \tag{4.15}$$

The equations (4.13-4.15) construct the closed-loop steady-state model that can be rewritten into the following form for convenience

$$u_{ssr} = G_{ur} t_{ss} + G_{udr} d_{m,0} \quad (4.16)$$

$$y_{ssr} = G_{yr} t_{ss} + G_{ydr} d_{m,0} + b_0 \quad (4.17)$$

$$(I - I_{\delta,ss}) t_{ss} = u_c \quad (4.18)$$

Substitute the equations (4.7b-4.7d) in RSSO-CL with the above closed-loop model, then the robust steady-state optimization formulation is changed into the following form

RSSO-CLT:

$$\min_{t_{ss}, s} \quad c_y^T s_y + c_u^T s_u + f^T s \quad (4.19a)$$

$$\text{s.t} \quad u_{ssr} = G_{ur} t_{ss} + G_{udr} d_{m,0} \quad (4.19b)$$

$$y_{ssr} = G_{yr} t_{ss} + G_{ydr} d_{m,0} + b_0 \quad (4.19c)$$

$$(I - I_{\delta,ss}) t_{ss} = u_c \quad (4.19d)$$

$$u_{sp} = u_{ss} \quad (4.19e)$$

$$y_{sp} = y_{ss} \quad (4.19f)$$

$$u_{\min} \le u_{ssr} \le u_{\max} \quad (4.19g)$$

$$y_{\min} - s \le y_{ssr} \le y_{\max} + s \quad (4.19h)$$

$$-s_y \le y_{ss} - y^{(ref)} \le s_y \quad (4.19i)$$

$$-s_u \le u_{ss} - u^{(ref)} \le s_u \quad (4.19j)$$

$$s_y, s_u, s \ge 0 \quad (4.19k)$$

For all G_{ur}, G_{udr}, G_{yr}, G_{ydr} in the uncertainty region where the of freedom change from y_{sp}, u_{sp} to t_{ss}, the different realizations of the parameters G_{ur}, G_{udr}, G_{yr}, G_{ydr} can be calculated from different realizations of K_r, K_{dr} and the saturation of the manipulated variables (defined by $I_{\delta,ss}$).

Based on the previous development:

Remark 4.1: *If Assumption 4.1 holds and we know the correct saturation pattern of the manipulated variables at the steady state, the control law (4.13) is equivalent to the inner optimization problem u_{ssr}= NMPCSS(b_{ssr}) and the formulation RSSO-CLT is equivalent to the formulation RSSO-CL.*

The third step of the reformulation is to introduce an active set heuristic, which is similar to the one discussed in Chapter 3, to obtain the active bounds on the manipulated variables in an iterative way. The heuristic is given in the following steps:

1) Assume no bounds are active at steady state, set all elements in δ_{ss} to 1, and solve problem RSSO-CLT.
2) The solution of problem RSSO-CLT gives the uncertain steady-state manipulated variables. If some manipulated variables, which are not assumed to saturate, have a value(s) at its bound (i.e., the boundaries of their uncertainty regions reach the upper or lower bounds on these manipulated variables), go to step (3); otherwise, end the iterative procedure and the current solution is the final solution.
3) Review the saturation status of all manipulated variables not already fixed at their bounds. Fix the manipulated variables that have encountered their bounds to their bound values (by specifying δ_{ss}, u_c). Solve problem RSSO-CLT again and go to step (2).

The heuristic does not guarantee the "global optimum" of solution, i.e., there may be another saturation pattern that is better than the one found by the heuristic. However, the heuristic converges to the optimum if the correct active set is selected. The maximum number of iterations in the heuristic is the number of manipulated variables.

The problem RSSO-CLT is an LP with uncertain linear constraints. Using the method explained in Section 3.3.1 of Chapter 3, this problem can be transformed approximately into a SOCP by chance-constrained program technique with a given confidence level α_{ss} as

RSSO-CLTSOCP:

$$\min_{t_{ss}, s} \; c_y^T s_y + c_u^T s_u + f^T s \tag{4.20a}$$

s.t.

$$\left.\begin{aligned} & E(G_{ur,l})t_{ss} + E(G_{udr,l})d_{m,0} \\ & + \Phi^{-1}(\alpha_{ss}) \| V_{uss,l}^{1/2}\left(t_{ss}^T, d_{m,0}^T, 1\right)^T \|_2 \le u_{\max,l} \quad (4.20b) \\ & E(G_{ur,l})t_{ss} + E(G_{udr,l})d_{m,0} \\ & - \Phi^{-1}(\alpha_{ss}) \| V_{uss,l}^{1/2}\left(t_{ss}^T, d_{m,0}^T, 1\right)^T \|_2 \ge u_{\min,l} \quad (4.20c) \end{aligned}\right\} \; l = 1,\dots,n_u$$

$$\left.\begin{aligned} & E(G_{yr,l})t_{ss} + E(G_{ydr,l})d_{m,0} \\ & + \Phi^{-1}(\alpha_{ss}) \| V_{yss,l}^{1/2}\left(t_{ss}^T, d_{m,0}^T, 1\right)^T \|_2 \le y_{\max,l} + s_l \quad (4.20d) \\ & E(G_{yr,l})t_{ss} + E(G_{ydr,l})d_{m,0} \\ & - \Phi^{-1}(\alpha_{ss}) \| V_{yss,l}^{1/2}\left(t_{ss}^T, d_{m,0}^T, 1\right)^T \|_2 \ge y_{\min,l} - s_l \quad (4.20e) \end{aligned}\right\} \; l = 1,\dots,n_y$$

$$(I - I_{\delta,ss})t_{ss} = u_c \tag{4.20f}$$

$$u_{sp} = u_{ss} = G_u t_{ss} + G_{ud} d_{m,0} \tag{4.20g}$$

$$y_{sp} = y_{ss} = G_y t_{ss} + G_{yd} d_{m,0} + b_0 \tag{4.20h}$$

$$-s_y \le y_{ss} - y^{(ref)} \le s_y \tag{4.20i}$$

$$-s_u \le u_{ss} - u^{(ref)} \le s_u \tag{4.20j}$$

$$s_y, s_u, s \ge 0 \tag{4.20k}$$

For all G_{ur}, G_{udr}, G_{yr}, G_{ydr} in the uncertainty region

where α_{ss} is constraint-wise confidence level, $G_{ur,l}$, $G_{udr,l}$ denote the lth row of matrices G_{ur}, G_{udr}, and $u_{\min,l}$, $u_{\max,l}$ denote the lth element in $u_{\min}$, $u_{\max}$ respectively, $G_{yr,l}$, $G_{ydr,l}$, denote the lth row of matrices G_{yp}, G_{ydp}, and $y_{\min,l}$, $y_{\max,l}$ denote the lth element in $y_{\min}$, $y_{\max}$ respectively. s_l denotes the lth row of the slack variable vector s.

$V_{uss,l}$ and $V_{yss,l}$ denote the covariance matrices of vector ($G_{up,l}$, $G_{udp,l}$, 1) and vector ($G_{yp,l}$, G_{ydplj}, 1) respectively, which are different for different saturation patterns of the manipulated variables (defined by $I_{\delta,ss}$). $V_{uss,l}$ is obtained through Monte Carlo sampling as follows (and a similar approach is used for $V_{yss,l}$):

1) Randomly select a sample of the open-loop uncertain parameters (K_r, K_{dr}) shown in model (4.3);
2) Calculate closed-loop steady-state uncertain parameters (G_{ur}, G_{udr}) accordingly;
3) Repeat procedure (1-2) for 100 samples of the open-loop uncertain parameters and obtain different groups of closed-loop uncertain parameters, which are then be used to calculate $V_{uss,l}$ according to the standard technique (Box et al., 2008).

The characterization of $V_{uss,l}$ and $V_{yss,l}$ can be performed using the same method introduced in Section 3.3.2., where the extensive computation is performed off-line and the real-time computation requires little time.

Equations (4.19g-4.19h) mean nominal steady state values y_{ss}, u_{ss} are sent to the MPC controller as the set points y_{sp}, u_{sp}.

Therefore, the new robust steady-state optimization method is implemented by solving the SOCP problem RSSO-CLTSOCP iteratively using the heuristic.

4.1.4 The deviation model to exploit the feedback information

In many cases, the uncertain parameters in an industrial MPC control system change slowly with respect to the closed-loop dynamics of the control system. In these cases, we can assume they are time-invariant, and we can exploit the feedback information to reduce the predicted uncertainty using the same idea we discussed in Section 3.2.5, i.e., to enhance the uncertain steady-state plant model (4.3) by expressing the variables as deviations from a virtual steady-state that is determined by the "most current" manipulated variables and the measured disturbances.

For more details, we define the variables at the virtual steady state are y_s, u_s, d_s, b_s, where u_s, d_s, e_s are known or are estimated at each controller execution as follows.

$$u_s = u_{-1} \tag{4.20}$$

$$d_s = d_{m,-1} \tag{4.21}$$

$$b_s = b_0 \tag{4.22}$$

and then y_s can be obtained through the nominal steady-state model as

$$y_s = Ku_s + K_d d_s + b_s \tag{4.23}$$

As discussed in Section 3.2.5, if there are time delays between the controlled and manipulated variables, we choose the value of a manipulated variable $\theta_{\max}$ time steps before for its value at the virtual steady state, where $\theta_{\max}$ denotes the maximum time delay between this manipulated variable and different controlled variables. We can also choose the value of a measured disturbance in the similar way if there are time delays between the controlled variables and the measured disturbances.

With the virtual steady state determined by equations (4.20-4.23), we can enhance the uncertain steady-state plant model (4.3) into the following deviation model.

$$y_{ssr} - y_s = K_r(u_{ssr} - u_s) + K_{dr}(d_{m,0} - d_s) \tag{4.24}$$

We can see that when the system is at a steady state, y_{ss} , u_{ss}, $d_{ss}(=d_{m,0})$, the virtual steady state coincides with the actual steady-state, i.e. $y_s=y_{ss}$, $u_s=u_{ss}$, $d_s= d_{ss}$. Then the deviation model (4.24) will correctly predict y_{ssr} to be y_{ss} and its uncertainty zero. However, the steady-state model (4.3), which is not formulated in deviation variables, would predict that y_{ssr} is different for different plant realizations (i.e. its uncertainty is not zero) at the steady state, which is not correct for time-invariant system. Therefore, in this thesis we use the deviation model for the prediction of time-invariant uncertainty in problem RSSO-CLTSOCP. The same idea has been successfully applied by Kassmann et al. (2000) in their robust steady-state optimization formulation with open-loop uncertainty.

As discussed in Section 3.2.5 in Chapter 3, when a deviation model is needed to handle time-invariant uncertainty, we do not need to change the structure of the general formulation developed; we only need to express the variables, their references, set points and bounds as deviation variables before the optimization as follows,

$$u'_{ssr} = u_{ssr} - u_s \tag{4.25}$$

$$d'_{mss} = d_{mss} - d_s \tag{4.26}$$

$$y'_{ssr} = y_{ssr} - y_s \tag{4.27}$$

$$b'_{ssr} = b_{ssr} - b_s \tag{4.28}$$

$$y'^{(reft)} = y^{(reft)} - y_s \tag{4.29}$$

$$u'^{(reft)} = u^{(reft)} - u_s \tag{4.30}$$

$$y'_{sp} = y_{sp} - y_s \tag{4.31}$$

$$u'_{sp} = u_{sp} - u_s \tag{4.32}$$

$$y'_{\max} = y_{\max} - y_s \tag{4.33}$$

$$y'_{\min} = y_{\min} - y_s \tag{4.34}$$

$$u'_{\max} = u_{\max} - u_s \tag{4.35}$$

$$u'_{\min} = u_{\min} - u_s \tag{4.36}$$

$$k = 0, \cdots, p-1$$

And after the optimization, we can restore the solution in the deviated variable form back to the original form (using equations (4.25-4.36) again).

Note that if the steady-state optimization is required for a system, this system usually can be deemed as a time-invariant system (otherwise the calculation of the steady state is meaningless). Therefore, for the case studies that run robust steady-state optimization in this research, all can be deemed as time-invariant. We will consistently use the deviation model (i.e., express the variables as deviation variables for optimization) for the robust steady-state optimization in the thesis.

4.1.5 Summary of the Robust Steady-State Optimization Method

According to the previous discussions in Section 4.1, the new robust steady-state optimization algorithm can be summarized as follows.

Calculation performed off-line:

1) Calculate the nominal value of the coefficients G_{ur}, G_{udr}, G_{yr}, G_{ydr} in the closed-loop model (4.16-4.17), for the situation no input bounds are active, according to equations (4.13-4.17);
2) Repeat the calculation in step 1 for other samples of the open-loop uncertain system (100 samples used in this thesis). Calculate the covariance matrices for the closed-loop coefficients according to the results of the sample calculations.

Calculation performed on-line at each controller execution period:

1) Obtain the reference values of the controlled and manipulated variables, $y^{(ref)}$, $u^{(ref)}$ according to plant personnel or upper level optimizer.
2) Read new measurements of controlled variables and the measured disturbances $y_{m,0}$, $d_{m,0}$ respectively. Set u_0 to be the implemented manipulated variables in the last controller execution.
3) Calculate nominally predicted controlled variable for the current time step, y_0, according to the previous implemented manipulated variables, measured disturbances and nominally estimated state and feedback variables. Then get the bias variables $b_0 = y_{m,0} - y_0$.
4) Calculate the virtual steady state according to equations (4.20-4.23) and deviate the variables, references, set points and bounds from the virtual steady state according to equations (4.25-4.36).

5) Assume no bounds are active at steady state, set all elements in δ_{ss} to 1, and solve the following problem RSSO-CLTSOCP2 with deviation variables:

RSSO-CLTSOCP2:

$$\min_{t_{ss},s} \quad c_y^T s_y + c_u^T s_u + f^T s \tag{4.37a}$$

s.t.

$$\left.\begin{aligned}
&E(G_{ur,l})t_{ss} + E(G_{udr,l})d'_{m,0} \\
&+\Phi^{-1}(\alpha_{ss}) \| V_{uss,l}^{1/2}\left(t_{ss}^T, d'^T_{m,0}, 1\right)^T \|_2 \le u'_{\max,l} && (4.37b)\\
&E(G_{ur,l})t_{ss} + E(G_{udr,l})d'_{m,0} \\
&-\Phi^{-1}(\alpha_{ss}) \| V_{uss,l}^{1/2}\left(t_{ss}^T, d'^T_{m,0}, 1\right)^T \|_2 \ge u'_{\min,l} && (4.37c)
\end{aligned}\right\} \quad l = 1,\ldots,n_u$$

$$\left.\begin{aligned}
&E(G_{yr,l})t_{ss} + E(G_{ydr,l})d'_{m,0} \\
&+\Phi^{-1}(\alpha_{ss}) \| V_{yss,l}^{1/2}\left(t_{ss}^T, d'^T_{m,0}, 1\right)^T \|_2 \le y'_{\max,l} + s_l && (4.37d)\\
&E(G_{yr,l})t_{ss} + E(G_{ydr,l})d'_{m,0} \\
&-\Phi^{-1}(\alpha_{ss}) \| V_{yss,l}^{1/2}\left(t_{ss}^T, d'^T_{m,0}, 1\right)^T \|_2 \ge y'_{\min,l} - s_l && (4.37e)
\end{aligned}\right\} \quad l = 1,\ldots,n_y$$

$$(I - I_{\delta,ss})t_{ss} = u'_c \tag{4.37f}$$

$$u'_{sp} = u'_{ss} = G_u t_{ss} + G_{ud} d'_{m,0} \tag{4.37g}$$

$$y'_{sp} = y'_{ss} = G_y t_{ss} + G_{yd} d'_{m,0} + b'_0 \tag{4.37h}$$

$$-s_y \le y'_{ss} - y'^{(ref)} \le s_y \tag{4.37i}$$

$$-s_u \le u'_{ss} - u'^{(ref)} \le s_u \tag{4.37j}$$

$$s_y, s_u, s \ge 0 \tag{4.37k}$$

For all G_{ur}, G_{udr}, G_{yr}, G_{ydr} in the uncertainty region

Note that the problem RSSO-CLTSOCP2 is different from the problem RSSO-CLTSOCP (equations (4.20a-4.20k) only in using the system variables and parameters that has been expressed as deviations from the virtual steady state. All the other variables and parameters in the problem RSSO-CLTSOCP2 are the same as defined before.

6) The solution gives the uncertain steady-state manipulated variables. If some manipulated variables, which are not assumed to saturate, have a value(s) at their bounds (i.e., the boundaries of their uncertainty regions reach the upper or lower bounds on these manipulated variables), go to step 7; otherwise, end the iterative procedure of the heuristic and go to step 8 (the current solution is the final solution obtained by the heuristic).
7) Review the saturation status of all manipulated variables not already fixed at their bounds. Fix the manipulated variables that have encountered their bounds to their bound values (by specifying δ_{ss}, u_c). Solve problem RSSO-CLTSOCP2 again and go to step 6.
8) Restore the solution from the deviation variable mode using equations (4.25-4.36) and send the solved set points y_{sp}, u_{sp} to the lower level controller.

4.2 Optimization of Robust Dynamic Performance

The general framework of robust dynamic MPC developed in Chapter 3 addresses the uncertainty explicitly in constraint handling only, where the objective of the optimization is still to minimize the *nominal* dynamic performance,

$$J(\mathbf{y},\mathbf{u},\Delta\mathbf{u}) = (\mathbf{y}-\mathbf{y}_{sp})^T\widetilde{Q}(\mathbf{y}-\mathbf{y}_{sp}) + (\mathbf{u}-\mathbf{u}_{sp})^T\widetilde{R}(\mathbf{u}-\mathbf{u}_{sp}) + \Delta\mathbf{u}^T\widetilde{W}\Delta\mathbf{u} \tag{4.38}$$

However, in some circumstances we would like to include in the objective function not only the nominal dynamic performance, but also some measure of uncertainty in the dynamic performance. (We will show several such circumstances in the cases studies in Section 4.3.) Our approach here is to express the robust objective function as a combination of the expected objective function and the appropriately weighted variances of the variables **y**, **u**, and **Δu** that provide a measure of the effects of the uncertainty on the performance. We will demonstrate that this form is the natural result of the uncertainty description and the application of the expectation operator. Then, we will prove that the robust objective function can be reformulated as a convex quadratic function of the optimization decision variables, **t**, so that the resulting robust MPC formulation can also be transformed (approximately) into SOCP problems.

As a first step in the development, we will prove the following relationship.

$$E(J(\mathbf{y}_r, \mathbf{u}_r, \Delta\mathbf{u}_r)) = J(E(\mathbf{y}_r), E(\mathbf{u}_r), E(\Delta\mathbf{u}_r)) + \sum_{l=1}^{n_y p} q_l Var(y_{r,l}) + \sum_{l=1}^{n_u n} r_l Var(u_{r,l}) + \sum_{m=1}^{n_y n} w_l Var(\Delta u_{r,l}) \tag{4.39}$$

where $\mathbf{y}_r$, $\mathbf{u}_r$, $\Delta\mathbf{u}_r$ are the uncertain controlled variables, manipulated variables and change of manipulated variables in the future horizon, $y_{r,l}$ denotes the lth element of vector $\mathbf{y}_r$, $u_{r,l}$ denotes the lth element of vector $\mathbf{u}_r$, q_l, r_l and w_l denote the lth element of the diagonal of weighting matrices $\widetilde{Q}$ $\widetilde{R}$ and $\widetilde{W}$, respectively, $V(\cdot)$ denotes the variance of the variables in the parentheses. For the convenience of notation, we define the following

$$J_1(\mathbf{y}_r) = (\mathbf{y}_r - \mathbf{y}_{sp})^T \widetilde{Q}(\mathbf{y}_r - \mathbf{y}_{sp}) \tag{4.40}$$

$$J_2(\mathbf{u}_r) = (\mathbf{u}_r - \mathbf{u}_{sp})^T \widetilde{R}(\mathbf{u}_r - \mathbf{u}_{sp}) \tag{4.41}$$

$$J_3(\Delta\mathbf{u}_r) = \Delta\mathbf{u}_r^T \widetilde{W} \Delta\mathbf{u}_r \tag{4.42}$$

then

$$E(J(\mathbf{y}_r, \mathbf{u}_r, \Delta\mathbf{u}_r)) = E(J_1(\mathbf{y}_r)) + E(J_2(\mathbf{u}_r)) + E(J_3(\Delta\mathbf{u}_r)) \tag{4.43}$$

We can reformulate equation (4.40) as

$$\begin{aligned} J_1(\mathbf{y}_r) &= (\mathbf{y}_r - \mathbf{y}_{sp})^T \tilde{Q}(\mathbf{y}_r - \mathbf{y}_{sp}) \\ &= \left[(\mathbf{y}_r - E(\mathbf{y}_r)) + (E(\mathbf{y}_r) - \mathbf{y}_{sp})\right]^T \tilde{Q}\left[(\mathbf{y}_r - E(\mathbf{y}_r)) + (E(\mathbf{y}_r) - \mathbf{y}_{sp})\right] \\ &= (\mathbf{y}_r - E(\mathbf{y}_r))^T \tilde{Q}(\mathbf{y}_r - E(\mathbf{y}_r)) + (E(\mathbf{y}_r) - \mathbf{y}_{sp})^T \tilde{Q}(E(\mathbf{y}_r) - \mathbf{y}_{sp}) \\ &\quad + 2(E(\mathbf{y}_r) - \mathbf{y}_{sp})^T \tilde{Q}(\mathbf{y}_r - E(\mathbf{y}_r)) \end{aligned} \tag{4.44}$$

So

$$\begin{aligned} E[J_1(\mathbf{y}_r)] &= E\left[(\mathbf{y}_r - E(\mathbf{y}_r))^T \tilde{Q}(\mathbf{y}_r - E(\mathbf{y}_r))\right] + E\left[(E(\mathbf{y}_r) - \mathbf{y}_{sp})^T \tilde{Q}(E(\mathbf{y}_r) - \mathbf{y}_{sp})\right] \\ &\quad + E\left[2(E(\mathbf{y}_r) - \mathbf{y}_{sp})^T \tilde{Q}(\mathbf{y}_r - E(\mathbf{y}_r))\right] \\ &= E\left[(\tilde{Q}^{1/2}\mathbf{y}_r - E(\tilde{Q}^{1/2}\mathbf{y}_r))^T (\tilde{Q}^{1/2}\mathbf{y}_r - E(\tilde{Q}^{1/2}\mathbf{y}_r))\right] + J_1(E(\mathbf{y}_r)) + 0 \end{aligned} \tag{4.45}$$

Note that the first term of the above expression is the sum of the variances of each element of vector $\tilde{Q}^{1/2}\mathbf{y}_r$, so it equals to $tr(Var(\tilde{Q}^{1/2}\mathbf{y}_r)) = tr(\tilde{Q}Var(\mathbf{y}_r))$. Therefore,

$$\begin{aligned} E[J_1(\mathbf{y}_r)] &= tr(\tilde{Q}Var(\mathbf{y}_r)) + J_1(E(\mathbf{y}_r)) \\ &= \sum_{l=1}^{n_y p} q_l Var(y_{r,l}) + J_1(E(\mathbf{y}_r)) \end{aligned} \tag{4.46}$$

Similarly, we can obtain

$$E[J_2(\mathbf{u}_r)] = \sum_{m=1}^{n_u n} r_m Var(u_{r,m}) + J_2(E(\mathbf{u}_r)) \tag{4.47}$$

$$E[J_3(\Delta \mathbf{u}_r)] = \sum_{m=1}^{n_u n} w_m Var(\Delta u_{r,m}) + J_3(E(\Delta \mathbf{u}_r)) \tag{4.48}$$

The equation (4.39) can be obtained by summing equations (4.46-4.48).

We can also include the variances of the variables $\mathbf{y}_r$, $\mathbf{u}_r$, and $\Delta\mathbf{u}_r$ in the objective to get a more general objective function as,

$$\begin{aligned}
&E(J(\mathbf{y}_r, \mathbf{u}_r, \Delta \mathbf{u}_r)) + \sum_{l=1}^{n_y p} \gamma_{y,l} Var(y_{r,l}) + \sum_{l=1}^{n_u n} \gamma_{u,l} Var(u_{r,l}) + \sum_{m=1}^{n_y n} \gamma_{\Delta u,l} Var(\Delta u_{r,l}) \\
&= J(E(\mathbf{y}_r), E(\mathbf{u}_r), E(\Delta \mathbf{u}_r)) \\
&\quad + \sum_{l=1}^{n_y p} (q_l + \gamma_{y,l}) Var(y_{r,l}) + \sum_{l=1}^{n_u n} (r_l + \gamma_{u,l}) Var(u_{r,l}) + \sum_{m=1}^{n_y n} (w_l + \gamma_{\Delta u,l}) Var(\Delta u_{r,l})
\end{aligned} \tag{4.49}$$

where $\gamma_{y,l}$, $\gamma_{u,l}$ and $\gamma_{\Delta u,l}$ are the weighting coefficients of the variances, whose non-negative values can be tuned to express extra emphasis for reduced variances.

Next, we will express each term in equation (4.49) as function of $\mathbf{t}$. We rewrite the closed-loop model developed in Chapter 3 again as follows (variable $\boldsymbol{\theta}$ will be replaced by $\boldsymbol{\pi}$ if some system states are not measurable and state estimation is required, as discussed in Section 3.4 of Chapter 3):

$$\mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \tag{4.50}$$

$$\mathbf{y}_r = L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_y\boldsymbol{\omega} \tag{4.51}$$

and from equation (4.50) we can obtain

$$\Delta\mathbf{u}_r = L_{\Delta ur}\mathbf{t} + M_{\Delta ur}\boldsymbol{\theta} + N_{\Delta u}\boldsymbol{\omega} \tag{4.52}$$

According to the closed-loop model (4.50-4.52), we can obtain

$$E(\mathbf{u}_r) = E(L_{ur})\mathbf{t} + E(M_{ur})\boldsymbol{\theta} + E(N_u)\boldsymbol{\omega} \tag{4.53}$$

$$E(\mathbf{y}_r) = E(L_{yr})\mathbf{t} + E(M_{yr})\boldsymbol{\theta} + E(N_y)\boldsymbol{\omega} \tag{4.54}$$

$$E(\Delta\mathbf{u}_r) = E(L_{\Delta ur})\mathbf{t} + E(M_{\Delta ur})\boldsymbol{\theta} + E(N_{\Delta u})\boldsymbol{\omega} \tag{4.55}$$

$$Var(y_{r,l}) = Var(L_{yr,l}\mathbf{t} + M_{yr,l}\boldsymbol{\theta} + N_{y,l}\boldsymbol{\omega}) = \| V_{y,l}^{1/2}\left(\mathbf{t}^T, \quad \boldsymbol{\theta}^T, \quad N_{y,l}^T\right)^T \|_2^2 \tag{4.56}$$

$$Var(u_{r,l}) = Var(L_{ur,l}\mathbf{t} + M_{ur,l}\boldsymbol{\theta} + N_{u,l}\boldsymbol{\omega}) = \| V_{u,l}^{1/2}\left(\mathbf{t}^T, \quad \boldsymbol{\theta}^T, \quad N_{u,l}^T\right)^T \|_2^2 \tag{4.57}$$

$$Var(\Delta u_{r,l}) = Var(L_{\Delta ur,l}\mathbf{t} + M_{\Delta ur,l}\boldsymbol{\theta} + N_{\Delta u,l}\boldsymbol{\omega}) = \| V_{\Delta u,l}^{1/2}\left(\mathbf{t}^T, \quad \boldsymbol{\theta}^T, \quad N_{\Delta u,l}^T\right)^T \|_2^2 \tag{4.58}$$

where $L_{yr,l}$, $M_{yr,l}$ denote the lth rows of the uncertain matrices L_{yr}, M_{yr} respectively, $V_{y,l}$ denotes the covariance matrix of vector ($L_{yr,l}$, $M_{yr,l}$, $\boldsymbol{\omega}$); $L_{ur,l}$, $M_{ur,l}$ denote the lth rows of the uncertain matrices L_{ur}, M_{ur} respectively, $V_{u,m}$ denotes the covariance matrix of vector ($L_{ur,l}$, $M_{yr,l}$, $\boldsymbol{\omega}$); $L_{\Delta ur,l}$, $M_{\Delta ur,l}$ denote the lth rows of the uncertain matrices $L_{\Delta ur}$, $M_{\Delta ur}$ respectively, $V_{\Delta u,l}$ denotes the covariance matrix of vector ($L_{\Delta ur,l}$, $M_{\Delta yr,l}$, $\boldsymbol{\omega}$).

According to equations (4.38) and (4.53-4.58), the objective function (4.49) that includes both the expected performance and the variances or the system variables can be transformed into the following function of **t**,

$$\begin{aligned}
&\| \widetilde{Q}^{1/2}(E(L_{yr})\mathbf{t} + E(M_{yr})\boldsymbol{\theta} + E(N_y)\boldsymbol{\omega} - \mathbf{y}_{sp}) \|_2^2 \\
&+ \| \widetilde{R}^{1/2}(E(L_{ur})\mathbf{t} + E(M_{ur})\boldsymbol{\theta} + E(N_u)\boldsymbol{\omega} - \mathbf{u}_{sp}) \|_2^2 \\
&+ \| \widetilde{W}^{1/2}(E(L_{\Delta ur})\mathbf{t} + E(M_{\Delta ur})\boldsymbol{\theta} + E(N_{\Delta u})\boldsymbol{\omega}) \|_2^2 + \sum_{l=1}^{n_u n}(q_l + \gamma_{y,l}) \| V_{u,l}^{1/2}\left(\mathbf{t}^T, \quad \boldsymbol{\theta}^T, \quad N_{u,l}^T\right)^T \| \\
&+ \sum_{l=1}^{n_y n}(r_l + \gamma_{u,l}) \| V_{\Delta u,l}^{1/2}\left(\mathbf{t}^T, \quad \boldsymbol{\theta}^T, \quad N_{\Delta u,l}^T\right)^T \|_2^2 + \sum_{l=1}^{n_y p}(w_l + \gamma_{\Delta u,l}) \| V_{y,l}^{1/2}\left(\mathbf{t}^T, \quad \boldsymbol{\theta}^T, \quad N_{y,l}^T\right)^T \|_2^2
\end{aligned} \tag{4.59}$$

It is easy to confirm that the objective function (4.59) is convex and quadratic with respect to the degrees of freedom **t**. Therefore, with this "robust" objective function, the robust MPC can still be solved by solving a series of SOCP problems using the method developed in Chapter 3.

Note that as stated in Section 3.2.5, if the uncertainty in the system can be deemed as time-invariant and the deviation model is needed for better performance, we can still optimize an objective function with the same structure of equation (4.59) for the expected performance and weighted variances of the system variables; and the only change is replacing the variables in equation (4.59) with the corresponding deviation variables that have been deviated from the virtual steady state. For the simplicity of the discussion, we will not show a separate objective function for time-invariant uncertainty only.

4.3 Case Study Results and Discussion

This section contains the case studies of the several distillation control and CSTR control systems, which are investigated to show:

1) The advantage of using closed-loop uncertainty over using open-loop uncertainty in the robust steady-state optimization;
2) The advantage of the robust steady-state and dynamic optimization over the nominal steady-state and dynamic optimization in Multiple Input Multiple Output (MIMO) system;
3) The advantage of minimizing robust performance instead of nominal performance in the robust MPC formulation for particular situations.

The simulation case studies were performed on a PC with Intel Core 2 Duo 3.0 GHz, 4GB memory and Windows Vista. The solution for the plant simulation is programmed in MATLAB 7.5 and the QP and SOCP problems are solved in GAMS with the interior point (barrier) solver of CPLEX 11. The data in MATLAB and CPLEX are exchanged using the interface software MATGAMS developed by Ferris (2005). All the system models are initially expressed with continuous input-output transfer functions, and they are all discretized and transformed into state-space model using the Control System Toolbox in MATLAB 7.5.

4.3.1 The control and optimization methods used in the case studies

We will evaluate several dynamic control and steady-state optimization methods in the case studies, through which the advantage of the new methods will be demonstrated. These methods include:

1) The nominal steady-state optimization

This method solves the LP problem (4.5) at each controller execution period.

2) The robust steady-state optimization

The detailed steps to implement this method are shown in Section 4.1.5.

3) The open-loop robust steady-state optimization

Here the open-loop robust steady-state optimization means the robust MPC method that uses open-loop uncertainty prediction (e.g. the method developed by Kassmann et al., 2000), where the controller action at the steady state are assumed to be unchanged for different realizations of the plant. This is equivalent to assuming that the control law of the closed-loop system at the steady state are $u_{rss}=t_{ss}$ and the steady-state system is optimized is by adjusting t_{ss}.

Therefore, this method can be implemented in the same way as detailed in Section 4.1.5 with the uncertainty calculated assuming that all the manipulated variables saturated, but with the values of the manipulated variables determined by $\mathbf{t}$ instead of being fixed to the bounds. The active set heuristic is not need here because the saturation pattern has been defined.

4) The different dynamic control methods

In the case studies in this section, we apply the nominal MPC, the robust MPC that is developed in Chapter 3, the unconstrained robust MPC that does not address input saturation in the closed-loop prediction, and the robust MPC minimizing robust performance that is developed in this chapter. The first three methods are the same as those we described in Section 3.5.1 in Chapter 3. The last method is similar to the robust MPC developed in Chapter 3, but the SOCP problems it solves have the robust objective function in the form of equation (4.59) instead of a nominal objective.

4.3.2 Binary distillation control system 1

Figure 4.2 shows the diagram of the binary distillation control system. The controlled variables are the distillate composition of light key XD (y_1) and the bottoms composition of light key XB (y_2). The manipulated variables are the Reflux rate R (u_1) and the reboiler rate V (u_2). The nonlinear model of the binary distillation process is described by a simulator, which is developed using the formulation from Marlin (1995) and the parameters from Luyben (1989). This nonlinear process can be linearized around

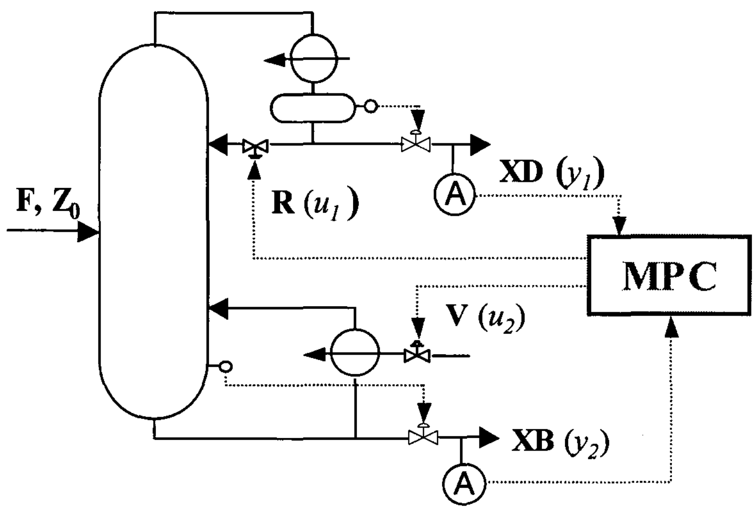


Figure 4.2　Binary distillation control system 1

the initial steady-state and expressed as input-output transfer functions. See Appendix G for the details the parameters and linearization procedure used in the thesis. We assume it takes $\theta = 10$ minutes for the distillate and bottoms outlet flows to reach the component analyzer and get analyzed, which introduces the time delay of 10 minutes between the controlled and manipulated variables.

The uncertainty of the system comes from the slowly varying feed in flow rate F, whose uncertain value is assumed to obey normal distribution with mean 8.7713 kmole/min and standard deviation 1.4619 kmole/min. The nominal plant model is derived at $F = 8.7713$ kmole/min as,

Nominal Model:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{0.1491e^{-10s}}{45.58s+1} & \dfrac{-0.1386e^{-10s}}{53.28s+1} \\ \dfrac{0.0649e^{-10s}}{34.13s+1} & \dfrac{-0.0775e^{-10s}}{31.33s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \tag{4.60}$$

The controller execution period for this system is selected to be 10 minutes. The linearized model of the system is discretized with sampling time of 10 minutes and transformed into state-space model using the MATLAB control system toolbox. The state-space form of the distillation model *without time delays and feedback variables* has the state vector x with 4 element and the system is controllable and observable. The feedback scheme assumes the unmeasured disturbance enters the system through the input channel (See Appendix A for discussion on selection of unmeasured disturbance model), which introduces the unmeasured disturbance vector e with 2 element. So the augmented system with x and e has 6 states, and it is detectable. Furthermore, the time delay between y and u is described by 2 additional states using the method introduced in Appendix C. Since these 2 states denote the u in the last time step, they are known, and no observer gain is need for them. The MPC controllers are tuned according to the methods described in Chapter 3, and Table 4-1 shows the tuning parameters.

The deterministic SOCP subproblem solved by the proposed robust MPC method for this system has 20 decision variables, 230 linear constraints and 140 second order cones. This problem is typically solved in 0.04 CPU seconds. When applying the active set heuristic, the maximum number of SOCP subproblems solved for this system is 10 (the number of time steps in the control horizon), so the robust MPC costs at most 0.04×10=0.4 CPU seconds.

Table 4-1 Tuning parameters of the MPC controllers for distillation control system 1

Tuning Parameter	Value
Control horizon, n	10
Prediction horizon, p	25
Estimation horizon, $p\_$	25
Observer gain for $[x^T, e^T]^T$, L	$\begin{bmatrix} 9.5 & 28 & 8.1 & 14 & 2.9 & 1.3 \\ -10 & -29 & -17 & -30 & -2.2 & -4.4 \end{bmatrix}^T$
Weights for controlled variables, $[q_1, q_2]$	[10, 100]
Weights for manipulated variables, $[r_1, r_2]$	[0.001, 0.001]
Move suppression weights $[w_1, w_2]$	[0.1, 0.1]
Penalty on slack variables $[w_{s,1}, w_{s,2}]$	$[10^5, 10^6]$
Cost of controlled variables c_y	$[10, 1]^T$
Cost of manipulated variables c_u	$[0, 0]^T$
Confidence of each stochastic bound in robust steady-state optimization, α_{ss}	99.7%
Confidence of each stochastic bound in robust MPC, α	99.9%

4.3.2.1 Closed-loop steady-state uncertainty

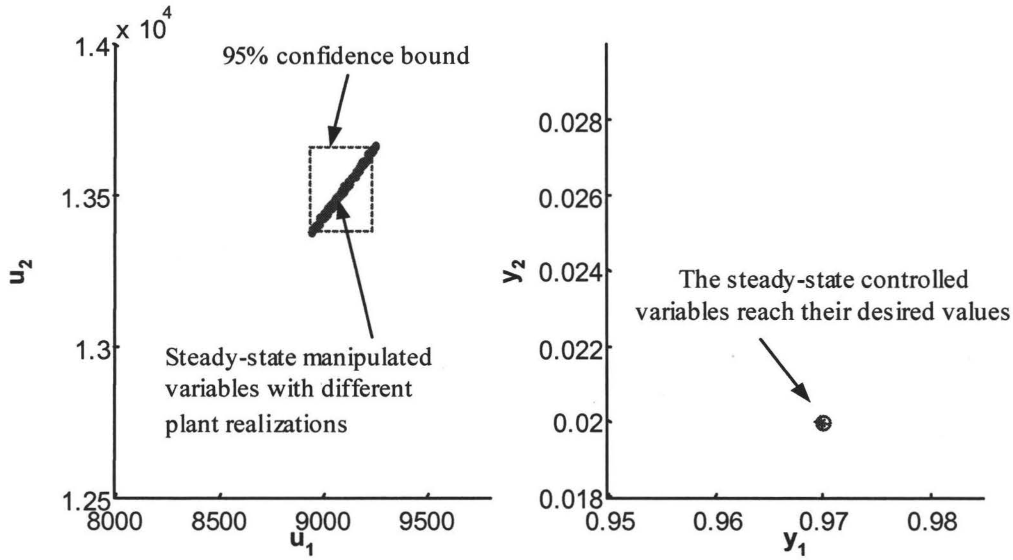
As discussed in Section 4.1, the correct prediction of the uncertain steady state must include the effect of the controller on the closed-loop system, which could make either the manipulated variable or the controlled variable or both uncertain at the steady state depending on the active set. An "open-loop" prediction of uncertainty is incorrect.

Figure 4.3 shows three situations of the steady state with the presence of uncertainty for the distillation control system 1. In the first situation, shown in Figure 4.3 (a), the system is unconstrained (or none of the constraints are active), so the steady-state values of the controlled variable will be equal to their reference values for all plant realizations because of the "implicit integral mode" in the MPC structure. Accordingly, the manipulated variables are different for different plant realizations.

Figure 4.3 (b) shows the second situation, where the upper bound on u_1 is active at the steady state. So here u_1 is the same for all realizations and u_2 is different for different plant realizations. Accordingly, the controlled variables cannot be kept at their desired values at the steady state, and their values are different for different plant realizations.

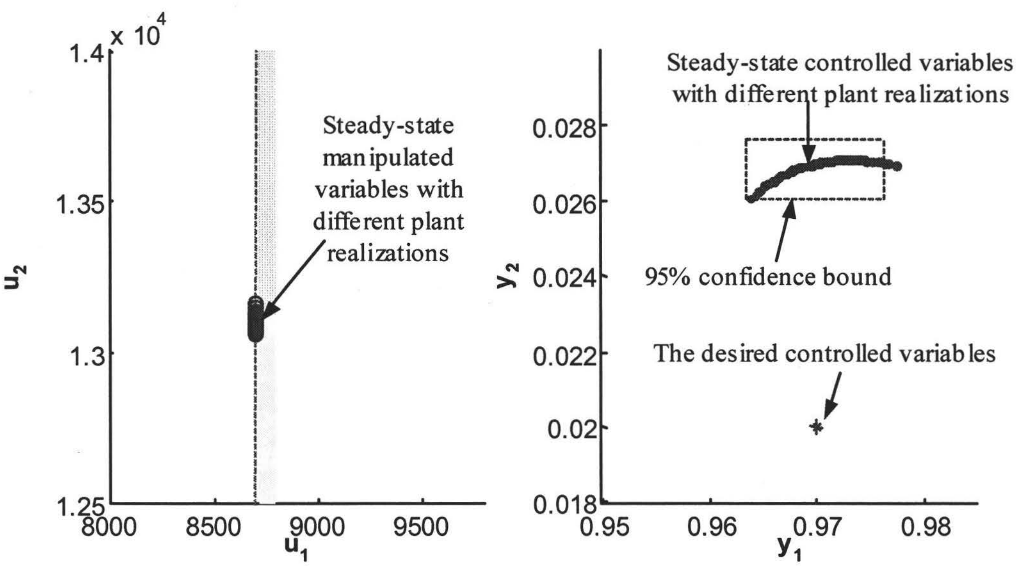
Figure 4.3 (c) shows the third situation, where the upper bounds of both manipulated variables are active at the steady state. Therefore, both of the manipulated variables are the same for all realizations, and both of the controlled variables are uncertain at the steady state. The controlled variables are far from their desired values, and part of their uncertainty region is outside the feasible region.

The results demonstrate the need of a closed-loop prediction, which includes the effect of the controller on the system, to accurately model the uncertainty at the steady state for the robust steady-state optimization. Next, we compare the robust steady-state optimization with open-loop and closed-loop uncertainty in the control of dynamic system.



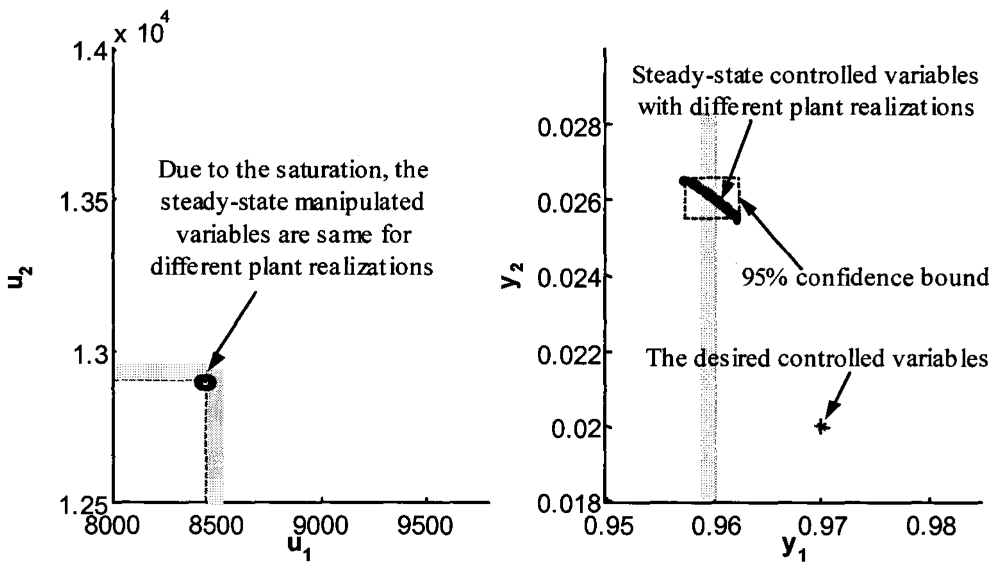
y_1 = XD (mole frac of light key) y_2 = XB (mole frac of light key) u_1=R (kmole/min) u_2 = V (kmole/min)

(a) No bounds are active



y_1 = XD (mole frac of light key) y_2 = XB (mole frac of light key) u_1=R (kmole/min) u_2 = V (kmole/min)

(b) One of the input bound is active



y_1 = XD (mole frac of light key) y_2 = XB (mole frac of light key) u_1=R (kmole/min) u_2 = V (kmole/min)

(c) Both input bounds are active

Figure 4.3 Closed-loop steady-state uncertainty - Binary distillation control system 1

4.3.2.2 Compare two robust steady-state optimization approaches: open-loop uncertainty vs. closed-loop uncertainty

Here, the closed-loop simulation is performed with either of the steady-state optimization methods (using either open-loop or closed-loop uncertainty description) and the robust MPC method developed in Chapter 3 (naturally, using the better, closed-loop uncertainty description in all cases here). We run the simulation for two situations. In both situations, the feed flow rate F equals to its nominal value 8.7713 kmole/min, i.e., plant = nominal model.

In the first situation, there is an initial step change of the reference of y_1 of +0.02 mole fraction and then a step change of –0.049 mole fraction. Figure 4.4 shows the closed-loop dynamics of the system under the two robust steady-state optimization methods integrated with the robust MPC for dynamic control. We can find that both methods lead to the same dynamics when the reference moves away from the y_1 bound, but the open-loop steady-state optimization results in more conservative control than appropriate when the reference moves toward the y_1 bound.

The results shown in Figure 4.4 can be understood by observing the variables in Figure 4.5, which compares the set points calculated by the two steady-state optimization methods. We observe that the set points calculated by both methods are same for the first

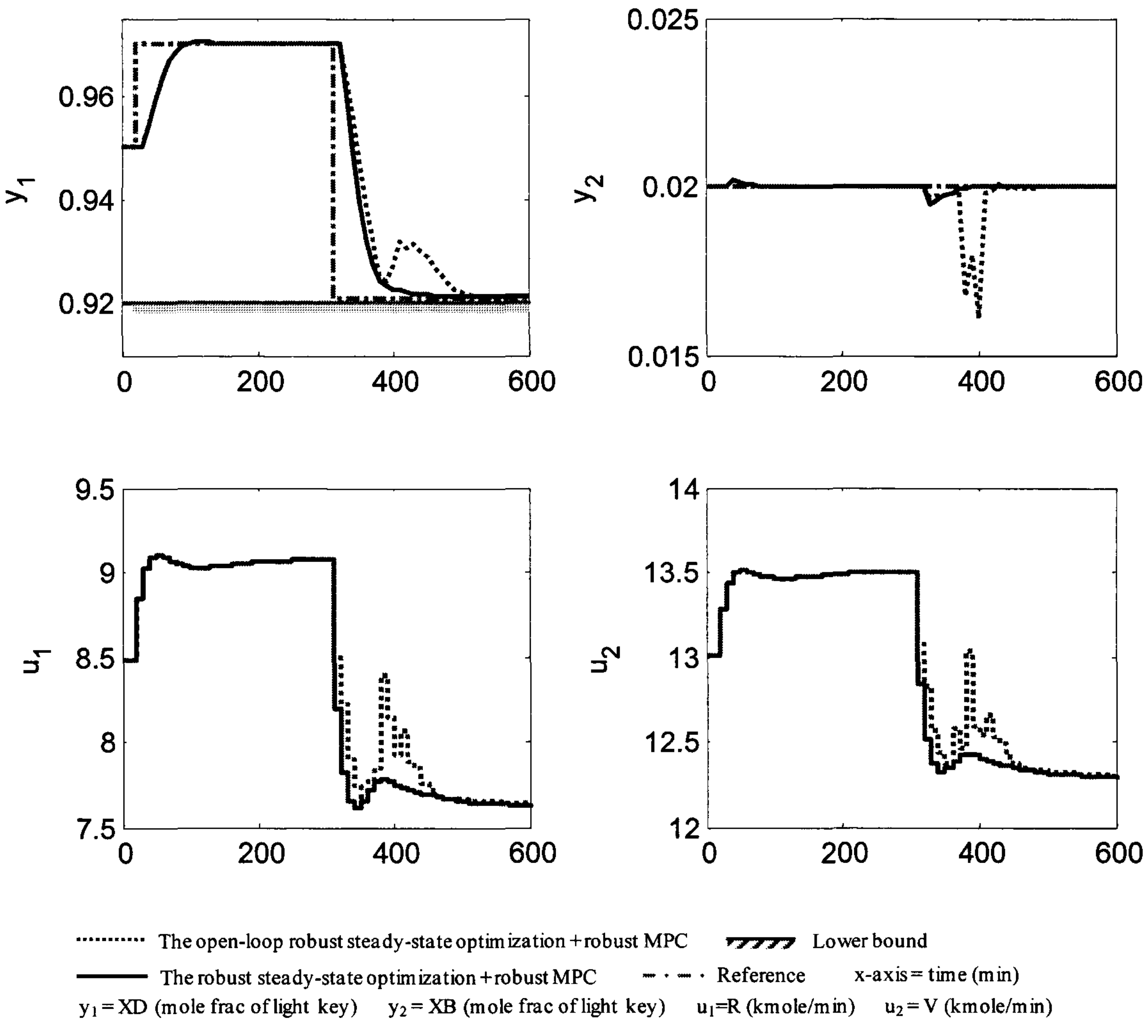


Figure 4.4 Closed-loop dynamics under the robust MPC and the two robust steady-state optimization methods (first situation) - binary distillation control system 1

part of the transient, because no bounds are active during the steady-state calculation and the reference values of the controlled variables can be achieved at the steady state. When the reference for y_1 approaches its bound, the y_1 set points ($y_{sp,1}$) calculated by the open-loop robust steady-state optimization differs greatly from its reference, and it is away from y_1 bound at many time steps. This is because the open-loop robust steady-state optimization method overestimates the steady-state uncertainty of y_1. In contrast, the

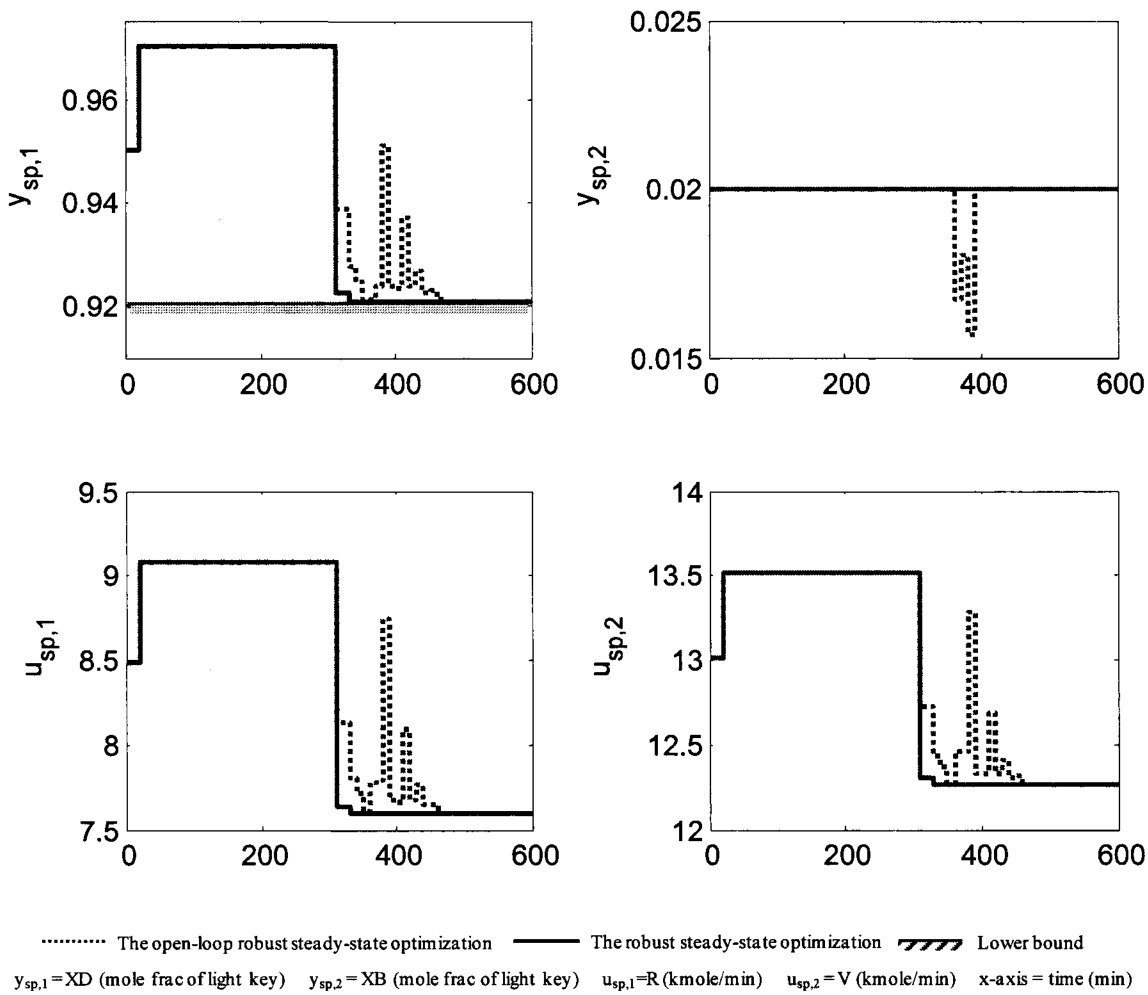


Figure 4.5 Set points calculated by the two robust steady-state optimization methods at each time step (first situation) - binary distillation control system 1

closed-loop uncertainty prediction accounts for the correcting action of the feedback controller and correctly predicts a much smaller uncertainty.

In the second situation, there is a step change of the y_1 reference of -0.03 mole fraction. Figure 4.6 shows the closed-loop dynamics of the system under the robust MPC. Although the y_2 reference does not change, y_2 is moved away from its reference during the transient for both methods. This is because the movement of the manipulated

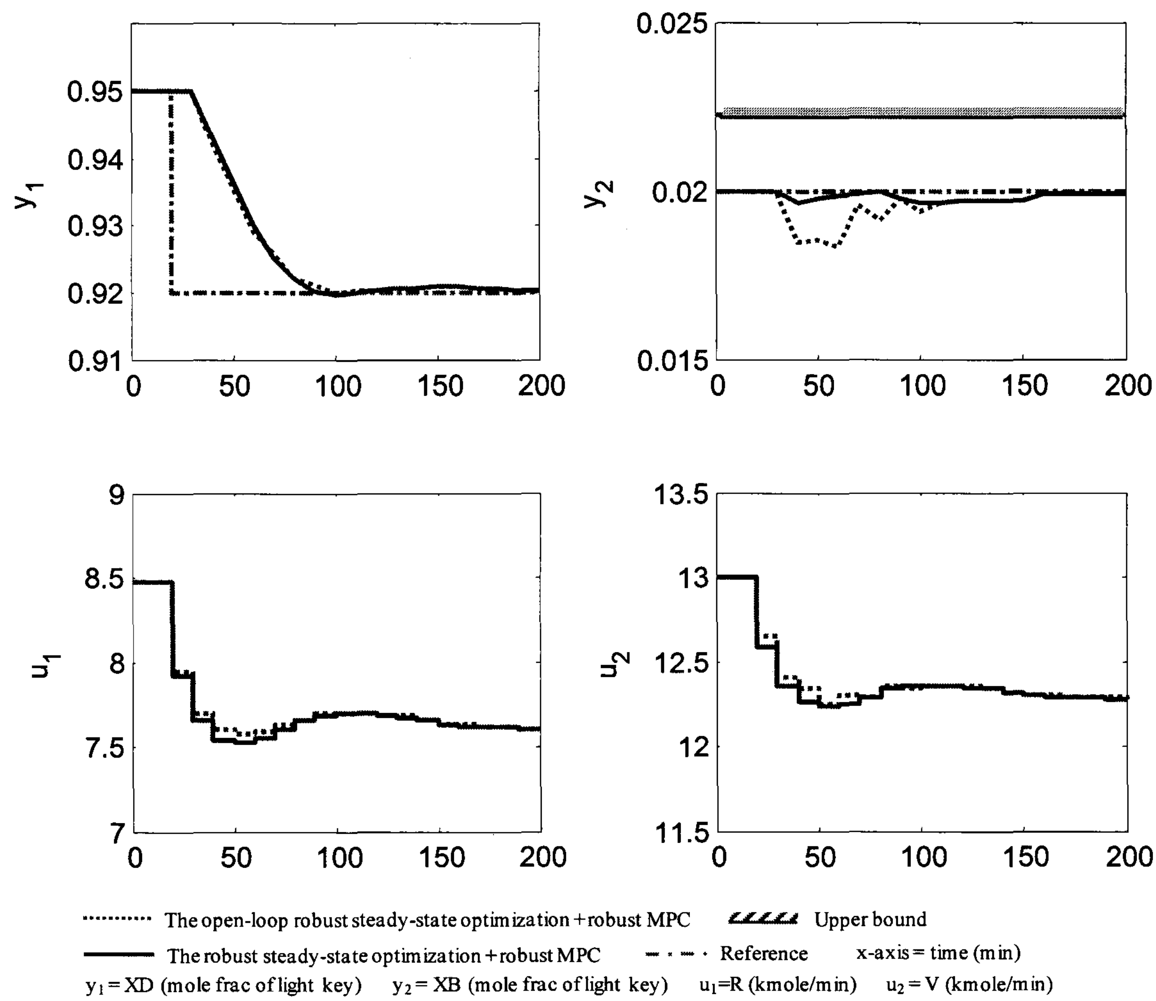


Figure 4.6 Closed-loop dynamics under the robust MPC and the two robust steady-state optimization methods (second situation) - binary distillation control system 1

variables for tracking the reference of y_1 influences the nominal value and the uncertainty of y_2 through the interaction in the MIMO system.

These results show that if the open-loop robust steady-state optimization is used, y_2 is moved further away form its reference, because the y_2 set points calculated by this method is further away from the y_2 upper bound due to the overestimation of the steady-state uncertainty This is shown clearly in Figure 4.7, which compares the set points calculated by the two robust steady-state optimization methods.

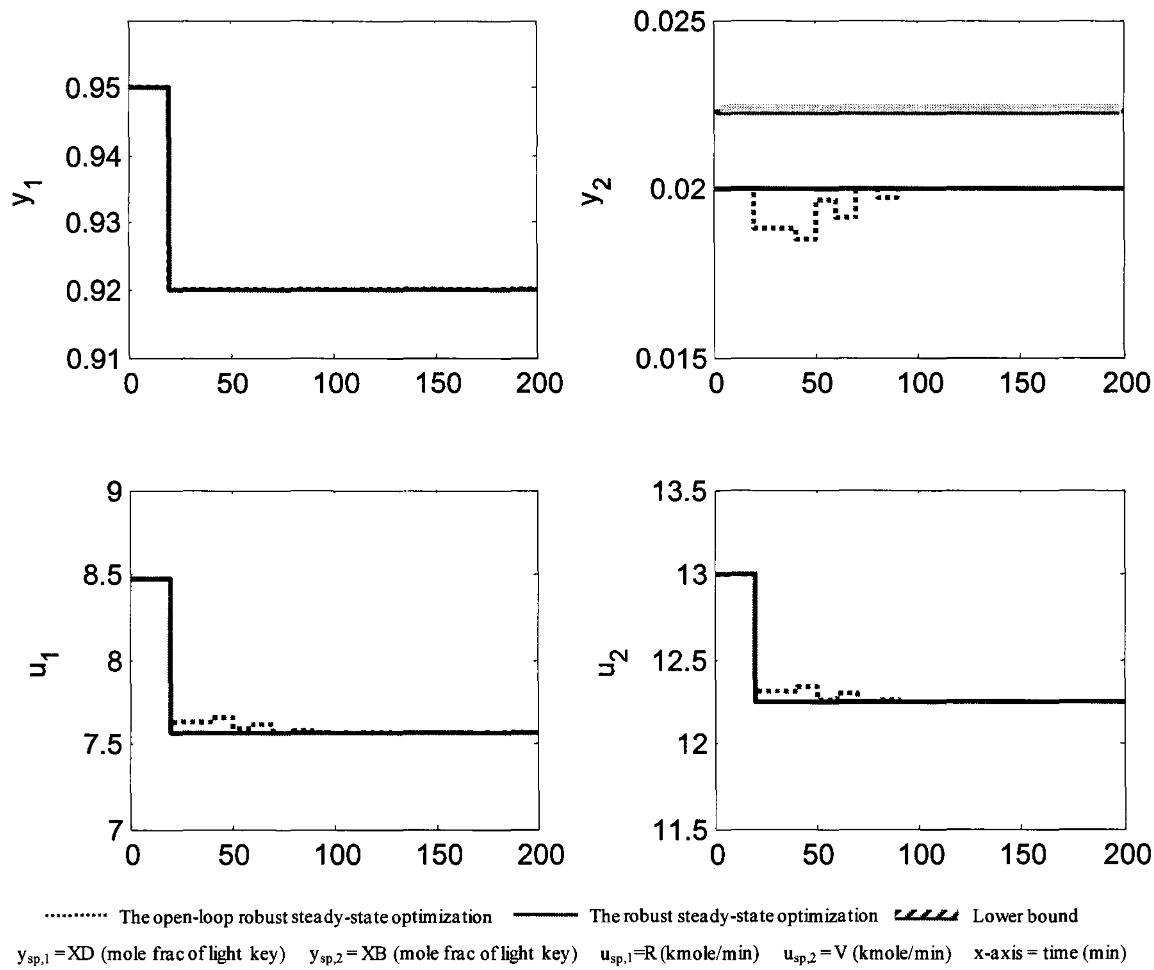


Figure 4.7 Set points calculated by the two robust steady-state optimization methods at each time step (second situation) - binary distillation control system 1

4.3.2.3 Set point tracking while observing output bounds

The simulations here are performed to demonstrate the advantage of using the robust steady-state optimization method (developed in this chapter) and the robust dynamic MPC method (developed in Chapter 3) over the traditional way of using nominal steady-state optimization and nominal dynamic optimization on observing output bounds. The robust formulations employed use recommended closed-loop uncertainty at both the steady-state and the dynamic optimization layers.

In this simulation, the feed flow rate in the plant is $F = 4.9438$ (kmole/min) which is different from its nominal value, so the plant is not the nominal model shown in equation (4.60) but the following model

Plant:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{0.2755e^{-10s}}{89.90s+1} & \dfrac{-0.2682e^{-10s}}{109.92s+1} \\ \dfrac{0.1190e^{-10s}}{64.32s+1} & \dfrac{-0.1369e^{-10s}}{57.04s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \tag{4.61}$$

Figure 4.8 compares the system dynamics under nominal and robust methods when a step change in the y_1 reference of –0.038 mole fraction is entered toward the y_1 lower bound occurs. Clearly the robust method observes the constraints while the nominal method leads to the violation of y_1 lower bound. This is due to the mismatch between the plant and the nominal model.

4.3.2.4 Set point tracking while observing hard input bounds

In this simulation, there is a step change in the y_1 reference of 0.03 mole fraction, and the plant equals the nominal model. We compare in Figure 4.9 the closed-loop dynamics under the robust steady-state optimization and the two robust MPC methods: the robust MPC and the unconstrained robust MPC that does not include input saturation in its optimization.

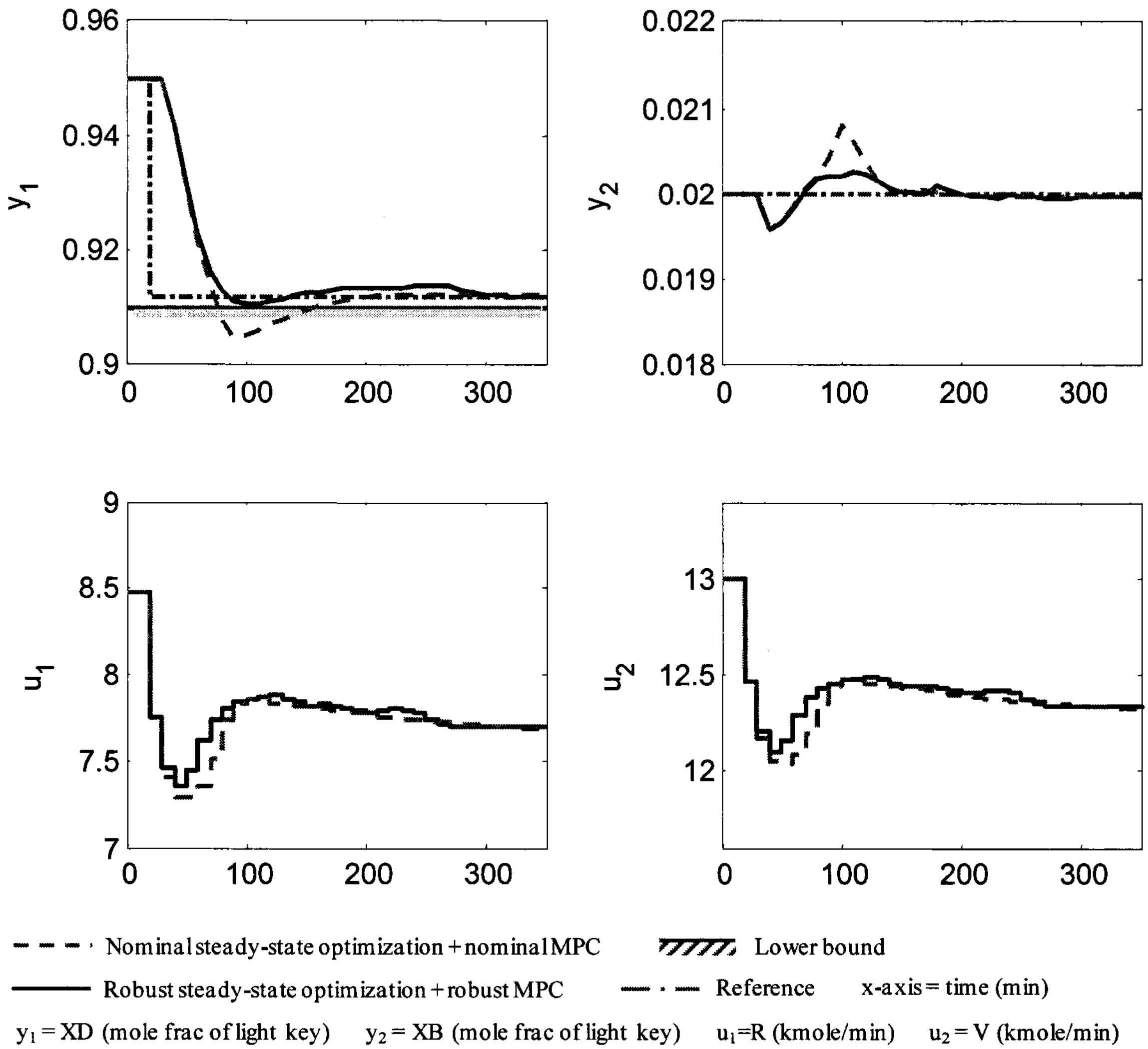
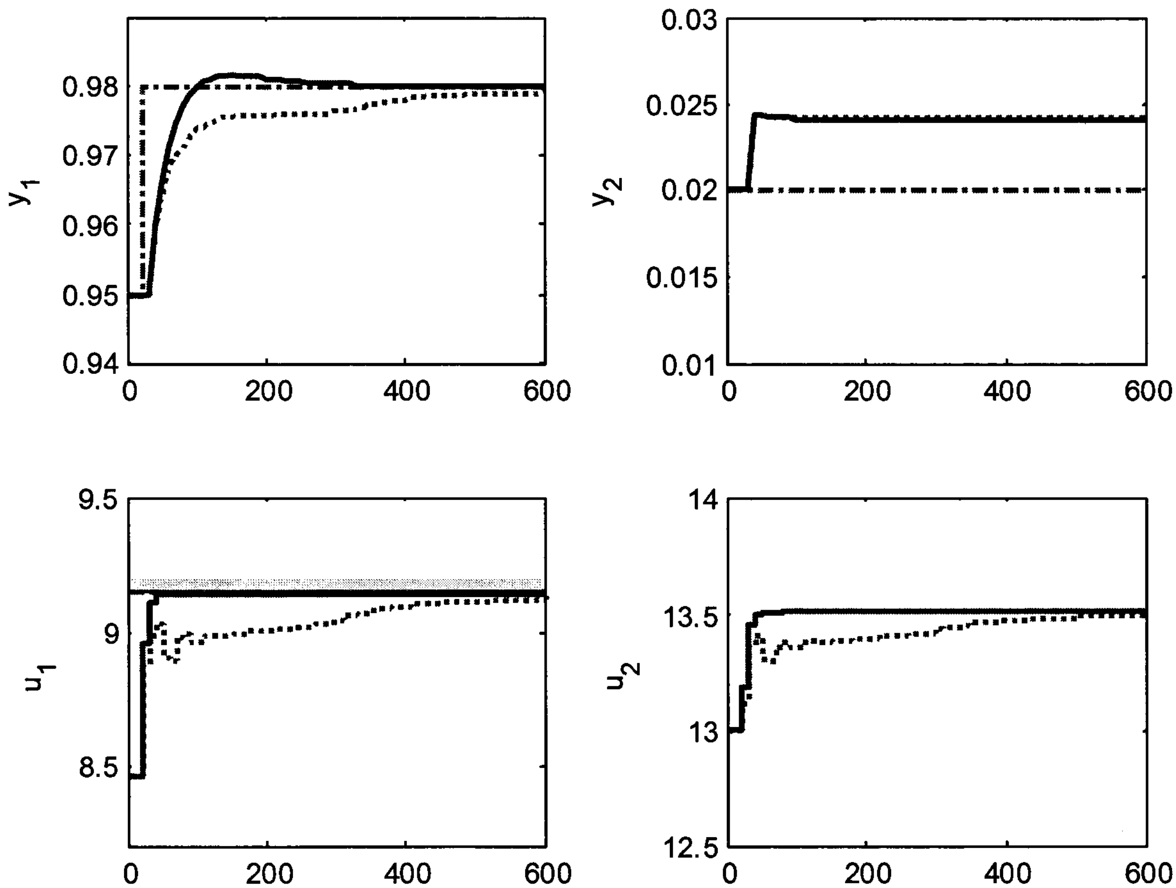


Figure 4.8 Set point tracking while observing output bounds
- binary distillation control system 1

First, we observe that the u_1 is limited by its upper bound so that the system cannot reach the desired steady-state reference values for both controlled variables. Due to the interaction in the system and the different importance of controlled variables (the cost of deviating from the reference for the top composition is ten times larger than for the bottoms composition), y_1 reaches the reference at the steady state and y_2 does not.

Second, we observe that when using the unconstrained robust MPC, u_1 approaches its upper bound very slowly because the closed-loop uncertainty is overestimated. Thus y_1 reaches its reference much slower.



Robust steady-state optimization + unconstrained robust MPC   Upper bound
Robust steady-state optimization + robust MPC   Reference   x-axis = time (min)
y_1 = XD (mole frac of light key) y_2 = XB (mole frac of light key) u_1=R (kmole/min) u_2 = V (kmole/min)

Figure 4.9 Set point tracking while observing hard input bounds - binary distillation control system 1

4.3.3 Binary distillation control system 2

Figure 4.10 shows the diagram of the second binary distillation control system. The binary distillation column is the same as in the previous two distillation control systems. Details of its model and initial conditions are described in Appendix G. The controlled variables are the distillate composition of light key XD (y_1) and the bottoms compositions of light key XB (y_2). The manipulated variables are the Reflux rate R (u_1) and the reboiler rate V (u_2). The controller will also receive the measured information of the feed flow rate F at the beginning of each controller execution period, which is deemed as measured disturbance (d) by the controller for feedforward compensation. We assume it takes $\theta = 10$ minutes for the distillate and bottoms outlet flows to reach the component analyzer and get analyzed, which introduces the time delay of 10 minutes between the controlled and manipulated variables.

The uncertainty of the system comes from the slowly varying feed composite of light key Z_0, whose value is assumed to obey the normal distribution with mean mean 0.5 mole fraction and standard deviation 0.033 mole fraction. The nominal linearized plant

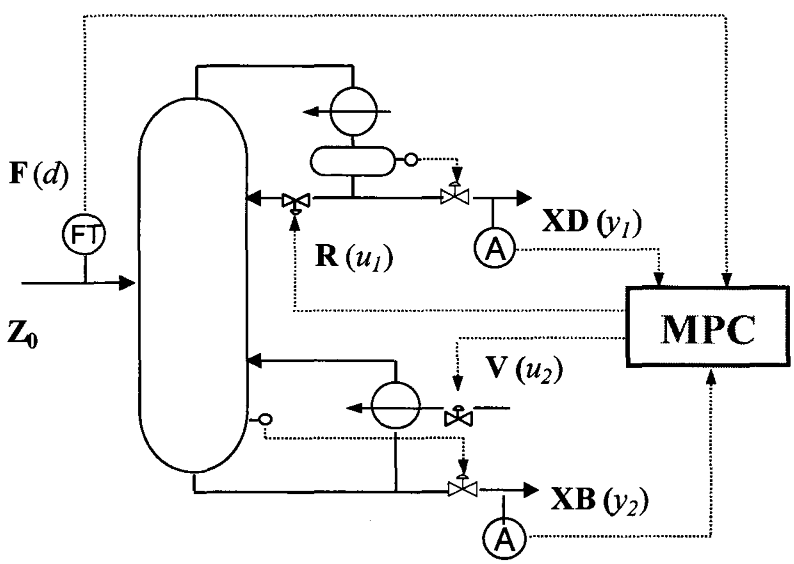


Figure 4.10 Binary distillation control system 2

model is derived at Z_0=0.5 mole fraction as,

Nominal Model:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{0.1491e^{-10s}}{45.58s+1} & \dfrac{-0.1386e^{-10s}}{53.28s+1} \\ \dfrac{0.0649e^{-10s}}{34.13s+1} & \dfrac{-0.0775e^{-10s}}{31.33s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} + \begin{bmatrix} \dfrac{0.0570e^{-10s}}{74.18s+1} \\ \dfrac{0.0562e^{-10s}}{30.07s+1} \end{bmatrix} d(s) \tag{4.62}$$

The case study evaluates the ability of the robust method to observe the bounds when rejecting the measured disturbances. We assume feed composition Z_0=0.5942 mole fraction, then the plant is

Plant:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{0.1163e^{-10s}}{43.87s+1} & \dfrac{-0.1099e^{-10s}}{55.11s+1} \\ \dfrac{0.0940e^{-10s}}{41.50s+1} & \dfrac{-0.1075e^{-10s}}{38.71s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} + \begin{bmatrix} \dfrac{0.0567e^{-10s}}{77.08s+1} \\ \dfrac{0.0857e^{-10s}}{38.39s+1} \end{bmatrix} d(s) \tag{4.63}$$

which is different from the nominal model.

The controller execution period for this system is selected to be 10 minutes. The linearized model of the system is discretized with sampling time of 10 minutes and transformed into state-space model using the MATLAB control system toolbox. The state-space form of the distillation model *without time delays and feedback variables* has the state vector x with 4 element and the system is controllable and observable. The feedback scheme assumes the unmeasured disturbance enters the system through the input channel (See Appendix A for discussion on the selection of unmeasured disturbance model), which introduces the unmeasured disturbance vector e with 2 element. So the augmented system with x and e has 6 states, and it is detectable. Furthermore, the time delay between y and u is described by 3 additional states using the method introduced in Appendix C. Since these 3 states denote the u and d in the last time step, they are known, and no observer gain is need for them. The MPC controllers are tuned according to the methods described in Chapter 3, and Table 4-2 shows the tuning parameters.

The deterministic SOCP subproblem solved by the proposed robust MPC method for this system has 20 decision variables, 230 linear constraints and 140 second order cones. This problem is typically solved in 0.04 CPU seconds. When applying the active set heuristic, the maximum number of SOCP subproblems solved for this system is 10 (the number of time steps in the control horizon), so the robust MPC costs at most 0.04×10=0.4 CPU seconds.

Table 4-2 Tuning parameters of the MPC controllers for distillation control system 2

Tuning Parameter	Value
Control horizon, n	10
Prediction horizon, p	25
Estimation horizon, $p\_$	25
Observer gain for $[x^T, e^T]^T$, L	$\begin{bmatrix} 9.5 & 28 & 8.1 & 14 & 2.9 & 1.3 \\ -10 & -29 & -17 & -30 & -2.2 & -4.4 \end{bmatrix}^T$
Weights for controlled variables, $[q_1, q_2]$	[1, 10]
Weights for manipulated variables, $[r_1, r_2]$	[0.001, 0.001]
Move suppression weights $[w_1, w_2]$	[0.1, 0.1]
Penalty on slack variables $[w_{s,1}, w_{s,2}]$	$[10^5, 10^6]$
Cost of controlled variables c_y	$[10, 1]^T$
Cost of manipulated variables c_u	$[0, 0]^T$
Confidence of each stochastic bound in robust steady-state optimization, α_{ss}	99.7%
Confidence of each stochastic bound in robust MPC, α	99.9%

Figure 4.11 shows the simulation results after a step change of the feed flow rate of 0.44 kmole/min. The closed-loop system dynamics under robust steady-state optimization and robust MPC and that under nominal steady-state optimization and nominal MPC are compared. When using the nominal methods, y_2 not only goes far away from its reference during the transient, but also violates its upper bound because of the plant/model mismatch. When using the robust methods that address the uncertainty explicitly, y_2 is driven far away from its upper bound during early stage of the transient to prevent the potential constraint violation for the "worst case" realization in this scenario.

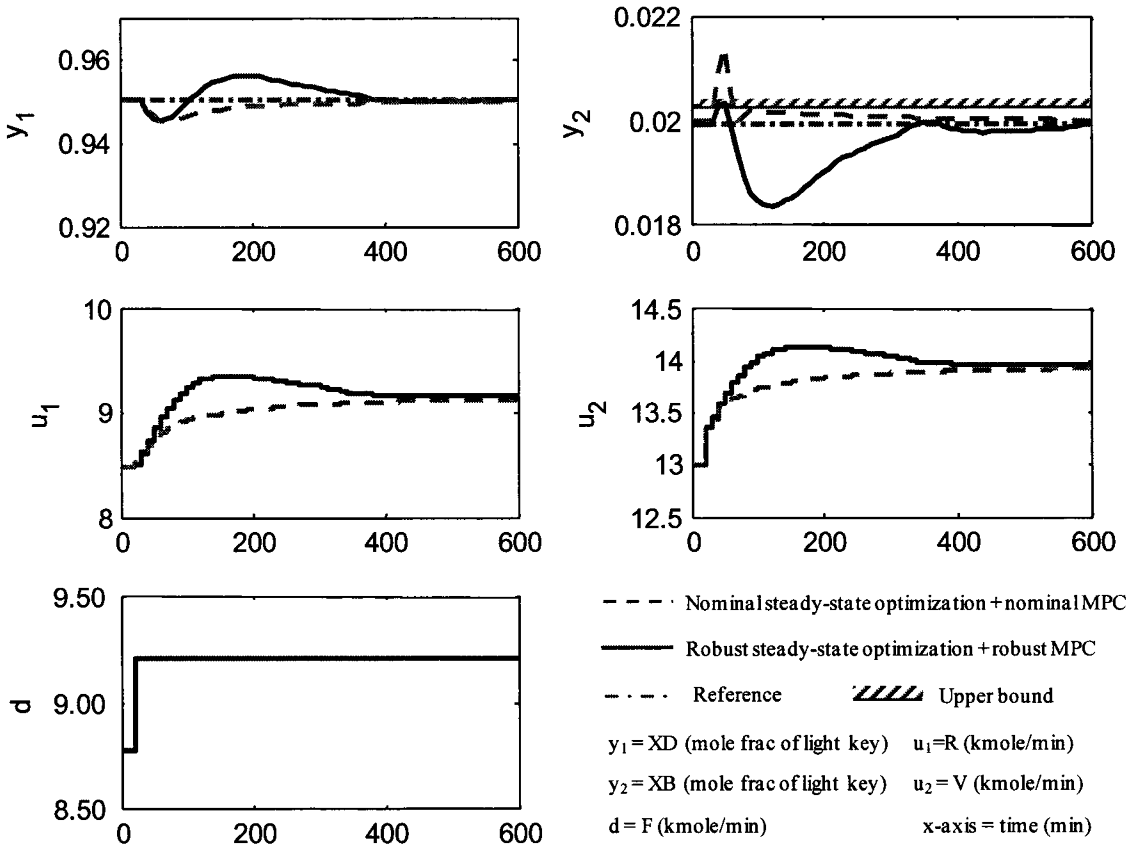


Figure 4.11 Disturbance rejection while observing output bound - binary distillation control system 2

4.3.4 Advantages of robust MPC minimizing robust performance

The purpose of the case studies in this section is to compare the robust MPC minimizing either nominal or robust performance. Two CSTR control systems are studied in the following two subsections 4.3.4.1 and 4.3.4.2, respectively. Both systems are non-square with one controlled variable and two manipulated variables. When a non-square system has more manipulated than controlled variables, opportunity exists to tailor the dynamic behavior to suit the objectives by considering the relative costs, dynamics and *uncertainties* of the manipulated variables. Both control systems have a robust steady-state optimization unit and a robust trajectory optimization unit. We will keep the robust steady-state optimization method the same as we developed in this chapter for all the case studies in this section (note the objective of the method is a nominal cost function and no "robust" cost functions are developed in this thesis), and we will compare the robust MPC (trajectory optimization) methods minimizing nominal and robust performance. The simulation results will show that minimizing robust performance in robust MPC leads to the preferred closed-loop control behavior when significant differences in uncertainties exist for different manipulated variables.

4.3.4.1 CSTR control system 3

Here we look at the CSTR control system 3 shown in Figure 4.12. The details of the parameters and the initial conditions of this process can be found in Table F-1 in Appendix F. The controlled variable of the system is the outlet concentration of reactant A, C_A (y). C_A is measured by an on-stream analyzer, and it takes $\theta = 0.9$ minutes for the outlet flow to reach the analyzer, which introduces the delay of 0.9 minutes between u and y. The temperature in the reactor, T, is measured without delay, for use by the observer for providing the estimates to the controller. Temperature is not controlled.

C_A is controlled by adjusting the inlet concentration of A, C_{A0}. This can be realized by changing the flows with high concentrations of A that are mixed with solvent to generate the reactor inlet flow. In this system, two flows of high concentration of A, $F_{A,1}$, $F_{A,2}$, are available. $F_{A,1}$ is cheap, but its concentration of A is slowly changing;

therefore, it has a significant uncertainty. $F_{A,2}$ is expensive, but its concentration of A is constant and known accurately.

For the simplicity of the discussion, let us define the manipulated variables of the control system to be u_1 and u_2 that denote the nominal C_{A0} by adjusting $F_{A,1}$ and $F_{A,2}$ respectively. Since $F_{A,1}$ is cheaper but uncertain, variable u_1 has a lower cost and greater uncertainty in its effect on C_{A0}; u_2 has a higher cost and negligible uncertainty in its effects on C_{A0}. The block diagram of this system is shown in Figure 4.13, where the gain between u_1 and C_{A0}, K_{CA0}, is uncertain. We assume K_{CA0} obeys the normal distribution with mean 1 and standard deviation 0.233.

The model of this CSTR system is the following.

$$y(s)=\frac{s+0.8078}{s^2+1.925s+1.143}K_{CA0}e^{-0.9s}u_1(s)+\frac{s+0.8078}{s^2+1.925s+1.143}e^{-0.9s}u_2(s) \tag{4.64}$$

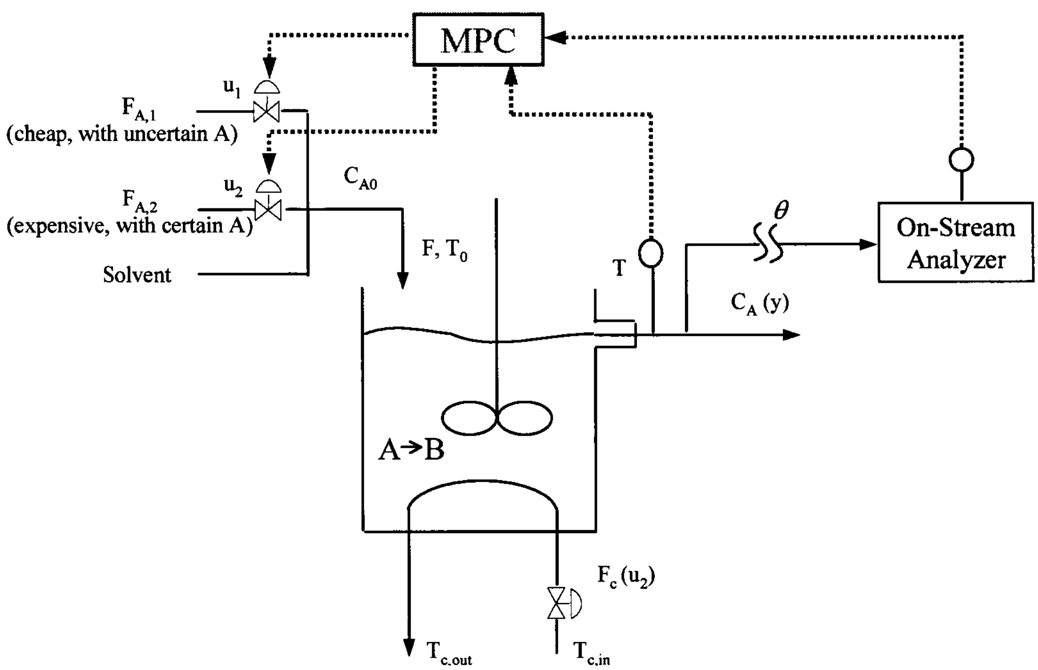


Figure 4.12 CSTR control system 3

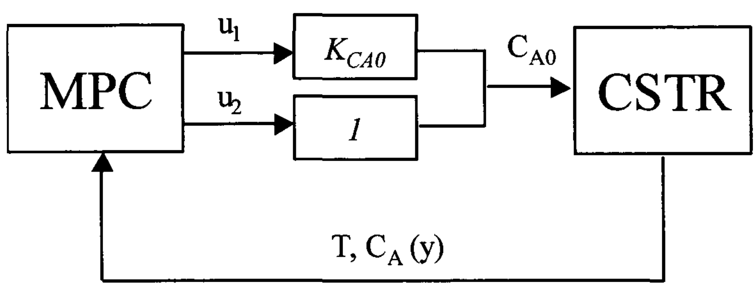


Figure 4.13 The diagram of the control structure of CSTR control system 3

The closed-loop control employs the robust steady-state optimization method developed in this chapter for steady-state optimization and the robust MPC method developed in Chapter 3 for dynamic control. The robust MPC method will minimize the following two objectives for each simulation case respectively:

1) The nominal performance + the nominal cost u:

$$(\mathbf{y}-\mathbf{y}_{sp})^T\tilde{Q}(\mathbf{y}-\mathbf{y}_{sp})+(\mathbf{u}-\mathbf{u}_{sp})^T\tilde{R}(\mathbf{u}-\mathbf{u}_{sp})+\Delta\mathbf{u}^T\tilde{W}\Delta\mathbf{u}+C_u\,|\,\mathbf{u}-\mathbf{u}_{sp}\,| \tag{4.65}$$

2) The robust performance (expected performance) + the nominal cost of u:

$$E\left[(\mathbf{y}_r-\mathbf{y}_{sp})^T\tilde{Q}(\mathbf{y}_r-\mathbf{y}_{sp})+(\mathbf{u}_r-\mathbf{u}_{sp})^T\tilde{R}(\mathbf{u}_r-\mathbf{u}_{sp})+\Delta\mathbf{u}_r^T\tilde{W}\Delta\mathbf{u}_r\right]+C_u\,|\,\mathbf{u}-\mathbf{u}_{sp}\,| \tag{4.66}$$

Note that the term $C_u\,|\mathbf{u}\text{-}\mathbf{u}_{sp}|$ represents the economic cost that is linear with respect to the manipulated variables (associated with the flow rates of A). The robust performance function in the objective function (4.66) is a special case of the robust objective function (4.49) developed in Section 4.2 (containing expected performance only), so we can transform it into a convex and quadratic function of **t** as discussed in Section 4.2.

The execution period for both the steady-state optimization and the dynamic control is selected to be 0.3 minute because the closed-loop settling time was about six minutes. The linearized continuous model of the system is discretized with sampling time of 0.3 minute and transformed into a state-space model using the MATLAB control system toolbox. The state-space form of the reactor model *without time delays and feedback variables* has the state vector x with 2 elements, and the system is controllable and observable. The feedback scheme assumes the unmeasured disturbance enters the system through the u_1 channel (See Appendix A for more discussion on the selection of unmeasured disturbance model), which introduces the unmeasured disturbance vector e with 1 element. So the augmented system with x and e has 3 states, and it is detectable. Furthermore, the time delay between y and u is described by 3 additional states using the method introduced in Appendix C. Since these 3 states denote the u_1 in the last 3 time steps, they are known, and no observer gain is need for them.

Since the case studies on this system focus on comparing the different objective functions instead of constraint handling, we pose loose inequality constraints on the system so that all the inequality constraints are inactive in all the simulations. The robust MPC controllers are tuned according to the methods described in Chapter 3, and the tuning parameters are shown in Table 4-4.

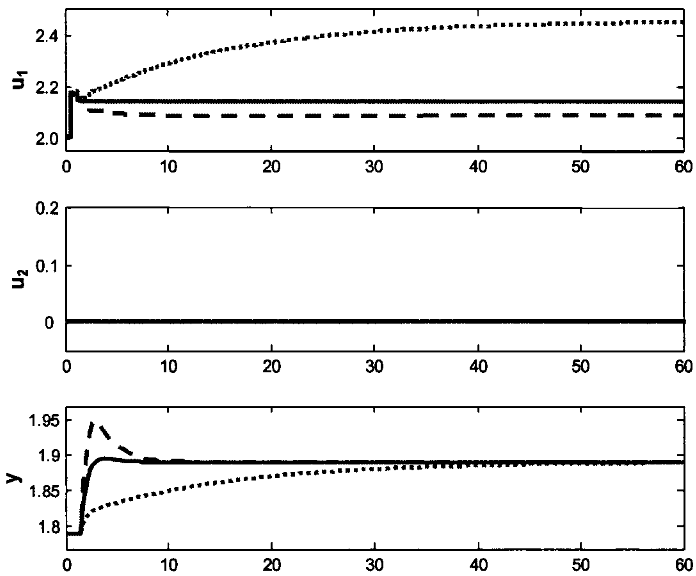
The deterministic SOCP subproblem solved by the proposed robust MPC method for this system has 16 decision variables, 184 linear constraints and 112 second order cones. This problem is typically solved in 0.02 CPU seconds. When applying the active set heuristic, the maximum number of SOCP subproblems solved for this system is 8 (the number of time steps in the control horizon), so the robust MPC costs at most 0.02×8=0.16 CPU seconds.

Figure 4.14 compares the system dynamic responses, to a step increase in the y reference from an upper-level optimizer of 0.10 kmole/m^3, which also results in a change in the u_1 reference of 0.14 kmole/m^3, with the robust MPC minimizing nominal and robust performance. Because the reference values are obtained by an optimizer, they are feasible for the nominal plant realization. The closed-loop system dynamics for three separate three realizations are plotted: nominal plant realization (K_{CA0} = 1), plant realization 1 (K_{CA0}= 0.32), and plant realization 2 (K_{CA0}= 1.61). Figure 4.14 (a) shows that when the robust MPC minimizes the nominal performance, only u_1 is moved and u_2 is unchanged for the set point tracking. This is because both u_1 and u_2 give the same

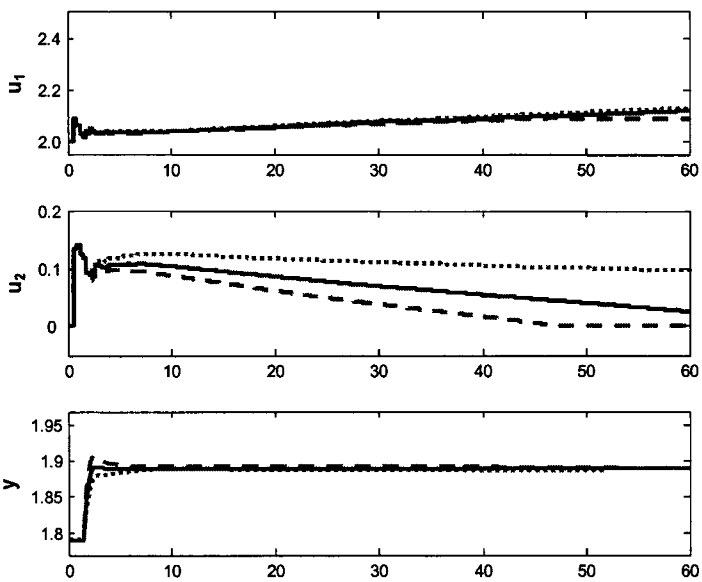
Table 4-4 Tuning parameters of the robust MPC controllers for CSTR control system 3

Tuning Parameter	Value
Control horizon, n	8
Prediction horizon, p	20
Estimation horizon, $p\_$	20
Observer gain for $[x^T, e^T]$, L	$\begin{bmatrix} 1 & 0 & 0.30 \\ 0 & 1 & 0.044 \end{bmatrix}^T$
Weight for controlled variable, q	10
Weight for manipulated variables, $[r_1, r_2]$	[0.01, .0.01]
Move suppression weights, $[w_1, w_2]$	[1, 1]
Costs of manipulated variables, $[c_{u,1}, c_{u,2}]$	[0.0001, 0.001]
Cost of the controlled variable, c_y	0.01
Confidence of each stochastic bound in robust steady-state optimization, α_{ss}	99.7%
Confidence of each stochastic bound in robust MPC, α	99.9%

nominal performance and u_2 is much more expensive than u_1. For the nominal plant realization where there is no plant/model mismatch, y is quickly driven to the new reference. However, for the other two plant realizations where there is severe plant/model mismatch, y either overshoots the reference or approaches the reference slowly. This performance is typical for nominal MPC and occurs for robust MPC as well when minimizing the nominal performance, because the controller does not consider the effects of uncertainty in the feedback on the objective. The performance is not very good, leaving potential for improvement.



(a) Robust MPC minimizing the nominal performance, equation (4.65)



(b) Robust MPC minimizing the robust performance, equation (4.66)

——— Nominal plant realization ·········· Plant realization 1 – – – Plant realization 2 y = measured C_A kmole/m^3

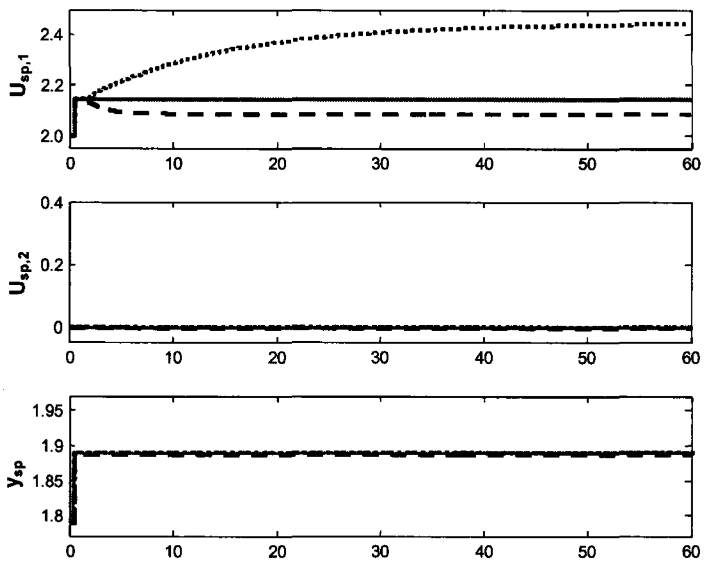
u_1 = Nominal C_{A0} by adjusting $F_{A,1}$ (kmole/m^3) u_2 = Nominal C_{A0} by adjusting $F_{A,2}$ (kmole/m^3) x-axis = time (min)

Figure 4.14 Comparing robust MPC with different objectives
– The system dynamic responses after reference step changes of CSTR control system 3

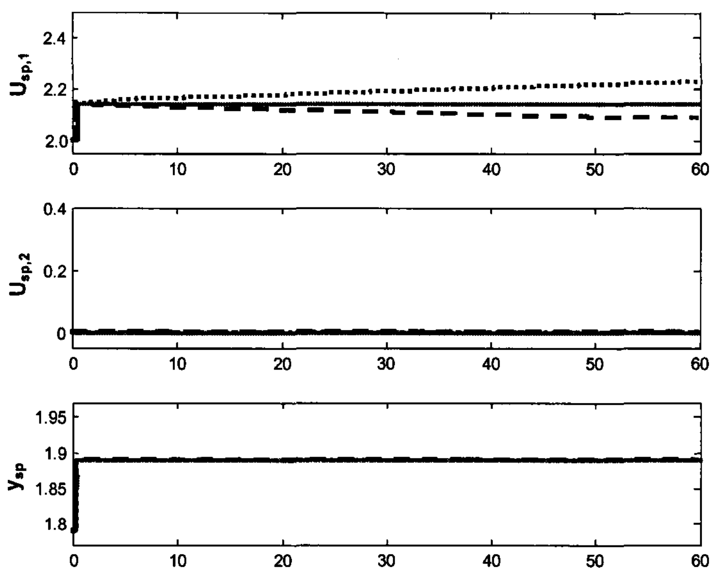
Figure 4.14 (b) shows that when the robust MPC minimizes the robust performance, it manipulates (increases) u_2 at the beginning of the transient. Once y is close to the reference, it decreases u_2 and increases u_1 slowly and simultaneously until u_2 is moved back to its initial position. By doing this, the robust MPC leads to good dynamic performance for all the three plant realizations. The robust MPC adjusts the certain u_2 instead of the uncertain u_1 at the beginning of the transient to avoid introducing large uncertainty into the system so that a good dynamic performance can be achieved. Once the system is close to the steady state, the more expensive manipulated variable u_1 can be replaced by the cheaper manipulated variable u_2 *slowly* without introducing large uncertainty. Therefore, by minimizing the robust performance in this case, the robust MPC properly chooses different manipulated variables for this non-square system at different stages of the transient.

Figure 4.15 shows the references to the system from an upper-level optimizer and the set points calculated by the steady-state optimization during the transient with the different robust MPC. In Figure 4.15 (a) with the robust MPC minimizing the nominal performance, the set point of y is the same as its reference value throughout the transient for all the three plant realizations, which is because reaching the y reference value at the steady state is more important than reaching the input reference values, and the this is feasible for the realizations simulated. The u_2 set point is also the same as its reference value, 0.0 kmole/m^3, because u_2 is more expensive than u_1 and it is to be kept at zero at steady state. The u_1 set point is the same as its reference value under the nominal plant realization, but it is different from this value when plant model mismatch occurs (under plant realizations 1 and 2). This is because when minimizing the nominal performance, y is controlled by manipulating uncertain input u_1, which introduces the uncertainty into the system. So the feedback variable (e) is different for different plant realization and different for different u_1 values during the transient, which leads to different set points through the steady-state optimization.

Figure 4.15 (b) shows that when the robust MPC minimizes the robust performance, the variation in the u_1 set points over different plant realizations is much less than that in Figure 4.15 (a). This is because here u_2 is manipulated to adjust y at the



(a) Set points with robust MPC minimizing the nominal performance, equation (4.65)



(b) Set points with robust MPC minimizing the robust performance, equation (4.66)

Nominal plant realization .... Plant realization 1 — — — · Plant realization 2

— · — · Reference y_{sp} = Set point of C_A kmole/m^3 x-axis = time (min)

$u_{sp,1}$ = Set point of nominal C_{A0} by adjusting $F_{A,1}$ (kmole/m^3) $u_{sp,2}$ = Set point of nominal C_{A0} by adjusting $F_{A,2}$ (kmole/m^3)

Figure 4.15 Comparing robust MPC with different objectives
– The system set points after reference step changes of CSTR control system 3

Table 4-4 Monte-Carlo Simulation Results of case study in Figure 4.14

	Average IAE [1]	Worse IAE
Robust MPC minimizing nominal performance	0.7353	1.8391
Robust MPC minimizing robust performance	0.4196	0.5813

Note: (1) IAE denotes Integrated Absolute Error.

beginning of the transient to avoid introducing large uncertainty in the dynamic performance; as a result, smaller changes in u_1 set points were required. Figure 4.15 (b) also shows that the set points of y and $u2$ are the same as their reference values throughout the transient, which is the same as in Figure 4.14 (a).

One hundred case studies of this closed-loop system with the Robust MPC using the two different objective functions have been run with Monte Carlo sampling of the plant realizations. The results are summarized in Table 4-4. We can find that when minimizing the robust performance, the better performance occurs for both the average performance and the worst-case metrics. We conclude that the robust MPC with the robust performance objective function can provide superior dynamic behaviour when a process has manipulated variables with different costs and uncertainties.

4.3.4.2 CSTR control system 4

Figure 4.16 shows the CSTR control system 4. The CSTR process is from page 438-439 of Marlin (2000). Details on the parameters and initial conditions of this CSTR process can be found in Table F-2 in Appendix F. The controlled variable (y) of the system is the outlet concentration of A, C_A. Both C_A and the temperature in the reactor T are measured and used by the observer for providing estimates to the controller. C_A can be controlled by a) adjusting the inlet concentration of A, C_{A0} or b) adjusting the cooling

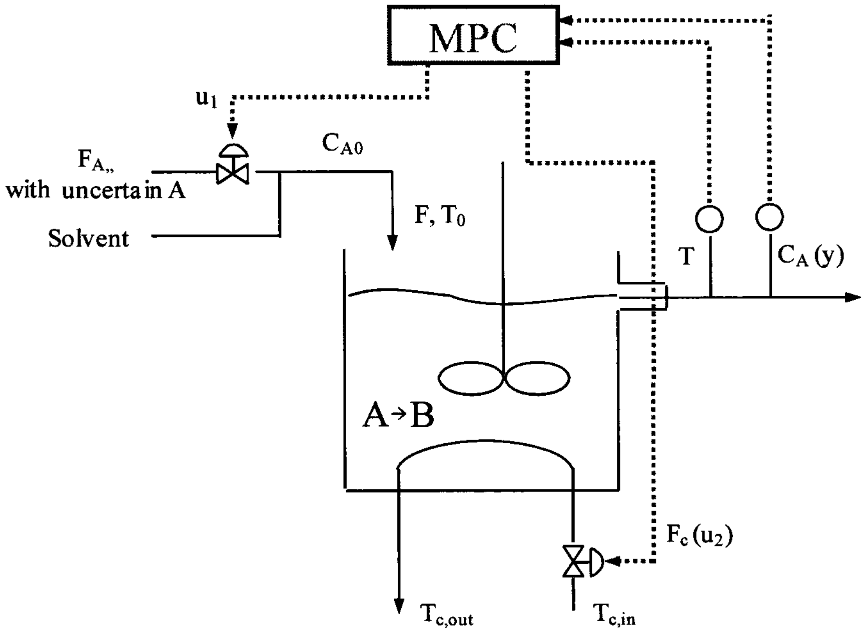


Figure 4.16 CSTR control system 4

follow rate Fc. The first manipulated variable (u_1) has faster dynamics, but there is large uncertainty associated with it. This uncertainty is from the slowly varying concentration of A in flow F_A that is mixed with solvent to generate the inlet flow. For the simplicity of the discussion, let us define u_1 is the nominal value of C_{A0} the controller achieves by changing the flow F_A. The second manipulated variable (u_2) has slower dynamics, but there is no uncertainty associated with it. We define u_2 the cooling flow rate, Fc.

Figure 4.17 gives a diagram of the structure of the CSTR control system 4, where K_{CA0} is the uncertain gain between u_1 and the actual C_{A0}. We assume K_{CA0} obeys the normal distribution with mean 1 and standard deviation 0.25. According to the parameters and the initial condition shown in Appendix F, the linearized model of this system is

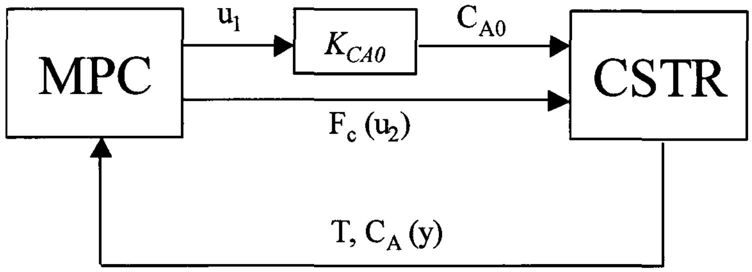


Figure 4.17 The diagram of the control structure of CSTR control system 4

$$y(s) = \frac{0.04048}{s + 0.07808} K_{CA0} u_1(s) + \frac{0.0123}{s^2 + 0.1617s + 0.00653} u_2(s) \tag{4.67}$$

The closed-loop control employs the robust steady-state optimization method developed in this chapter for steady-state optimization and the robust MPC method developed in Chapter 3 for dynamic control. The robust MPC method will minimize the following two objectives for each simulation case respectively:

1) <u>The nominal performance</u>:

$$(\mathbf{y} - \mathbf{y}_{sp})^T \widetilde{Q}(\mathbf{y} - \mathbf{y}_{sp}) + (\mathbf{u} - \mathbf{u}_{sp})^T \widetilde{R}(\mathbf{u} - \mathbf{u}_{sp}) + \Delta\mathbf{u}^T \widetilde{W} \Delta\mathbf{u} \tag{4.68}$$

2) <u>The robust performance (expected performance + weighted variance of y)</u>:

$$E\left((\mathbf{y}_p - \mathbf{y}_{sp})^T \widetilde{Q}(\mathbf{y}_p - \mathbf{y}_{sp}) + (\mathbf{u}_p - \mathbf{u}_{sp})^T \widetilde{R}(\mathbf{u}_p - \mathbf{u}_{sp}) + \Delta\mathbf{u}_p{}^T \widetilde{W} \Delta\mathbf{u}_p\right) + \cdot \sum_{l=1}^{n_y p} \gamma_{y,l} Var(y_{p,l}) \tag{4.69}$$

Note that we assume the economic costs associated with the manipulated variables are negligible for dynamic control, so that we do not include the linear cost in the objective function. The robust performance objective function (4.69) is a special case of the robust objective function (4.49) developed in Section 4.2 (without variances of inputs and input changes), so we can transform it into a convex and quadratic function of **t** as discussed in Section 4.2.

The controller execution period for this system is selected to be 3 minutes because the closed-loop settling time is about 60 minutes. The above linear model is discretized with sampling time of 3 minutes and transformed into state-space model using the MATLAB control system toolbox. The state-space form of the reactor model *without feedback variables* has the state vector x with 2 elements, and the system is controllable and observable. The feedback scheme assumes the unmeasured disturbance enters the system through the u_1 channel (See Appendix A for more discussion on the selection of unmeasured disturbance model), which introduces the unmeasured disturbance vector e with 1 element. So the augmented system with x and e has 3 states, and it is detectable.

Again, since the case studies on this system focus on comparing the different objective functions instead of constraint handling, we pose loose constraints on the system so that all the inequalities constraints are inactive in all the simulations. The robust MPC controllers are tuned according to the methods described in Chapter 3 and the tuning parameters are shown in Table 4-5.

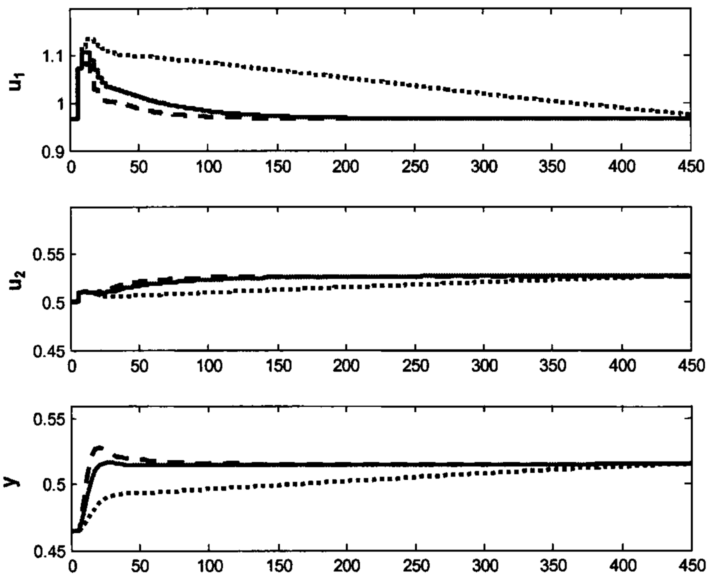
The deterministic SOCP subproblem solved by the proposed robust MPC method for this system has 16 decision variables, 214 linear constraints and 132 second order cones. This problem is typically solved in 0.03 CPU seconds. When applying the active set heuristic, the maximum number of SOCP subproblems solved for this system is 8 (the number of time steps in the control horizon), so the robust MPC costs at most 0.03×8=0.24 CPU seconds.

Figure 4.18 compares the system dynamic responses, to a step increase in the y reference of 0.050 kmole/m^3 and in the u_2 reference of 0.027 m^3/min, with the robust MPC minimizing nominal and robust performance. The closed-loop system dynamics are plotted for three realizations: nominal plant realization (K_{CA0}=1), plant realization 1 (K_{CA0}=0.26), and plant realization 2 (K_{CA0}=1.71). Figure 4.17 (a) shows that when the robust MPC minimizes the nominal performance, u_1 is manipulated relatively quickly, while u_2 is changed very slowly. This is because nominal u_1 can drive y to the reference much quicker than u_2 does. So, y is well controlled to the new reference value for the nominal plant realization. However, for the other two plant realizations where there is

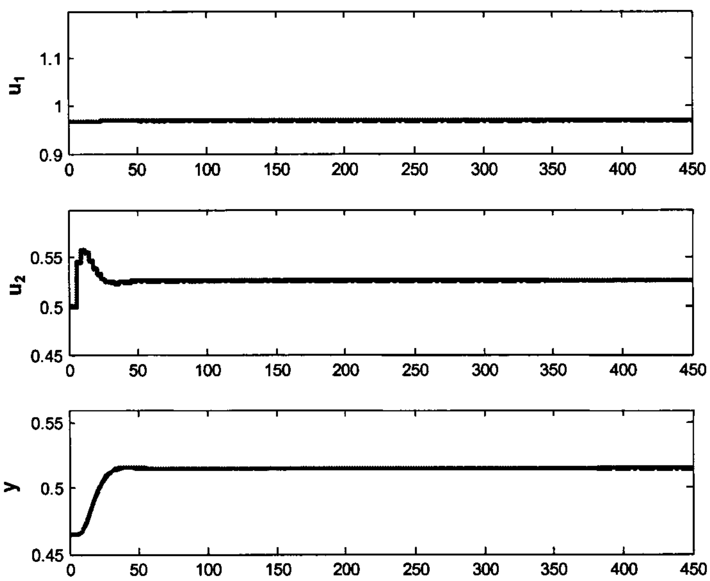
Table 4-5 Tuning parameters of the MPC controllers for CSTR control system 4

Tuning Parameter	Value
Control horizon, n	8
Prediction horizon, p	25
Estimation horizon, $p\_$	25
Gains of the state observer, L	$\begin{bmatrix} 1 & 0 & 0.01 \\ 0 & 1 & 0 \end{bmatrix}^T$
Weights for controlled variables, q	10
Weights for manipulated variables, $[r_1, r_2]$	[0.01, 0.1]
Move suppression weights, $[w_1, w_2]$	[1, 10]
Costs of the manipulated variables (used in robust steady-state optimization only), $[c_{u,1}, c_{u,2}]$	[0.01, 0.01]
Cost of the controlled variable (used in robust steady-state optimization only), c_y	10
Weight for variance of controlled variable, $\gamma_{y,l}$	1000
Confidence of each stochastic bound in robust steady-state optimization, α_{ss}	99.7%
Confidence of each stochastic bound in robust MPC, α	99.9%

severe plant/model mismatch, y either overshoots the reference substantially or approaches the reference slowly. This performance is typical for nominal MPC and occurs for robust MPC as well when minimizing the nominal performance, because it does not distinguish between feedback paths with large and small uncertainty. This performance is not very good, leaving potential for improvement.



(a) Robust MPC minimizing the nominal performance, equation (4.68)



(b) Robust MPC minimizing the robust performance, equation (4.69)

Nominal Realization ········· Plant Realization 1 – – – · Plant realization 2

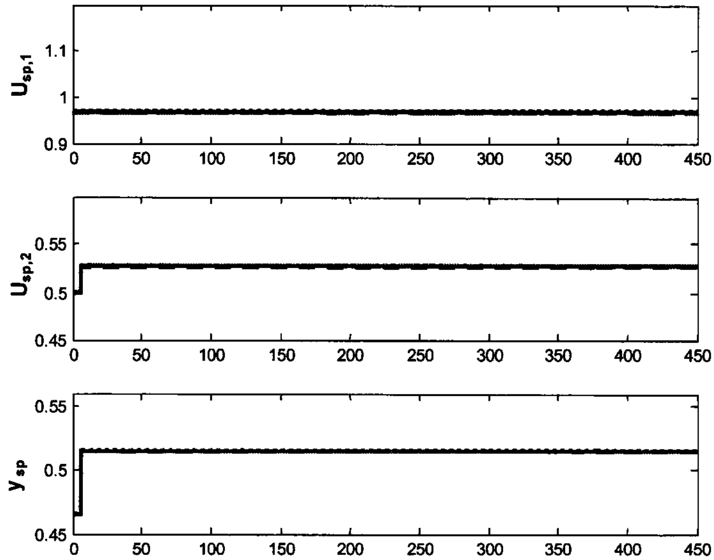
$y = C_A$ (kmole/m^3) u_1 = Nominal C_{A0} by adjusting flow F_A (kmole/m^3) $u_2 = F_c$ (m^3/min) x-axis = time (min)

Figure 4.18 Robust MPC with different objectives
– The system set points after reference step changes of CSTR control system 4

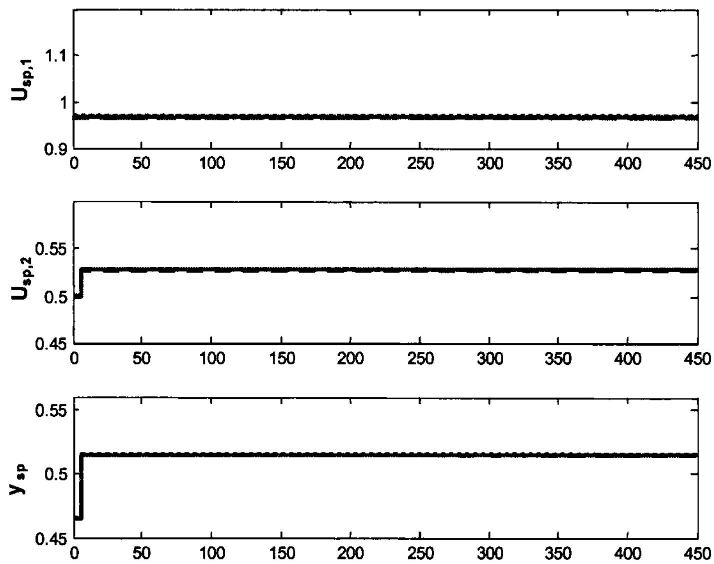
Figure 4.18 (b) shows that when the robust MPC minimizes the robust performance, it increases u_2 aggressively and keeps u_1 almost unchanged. This is because the controller avoids using the manipulated variable with large uncertainty, so that y trajectory does not vary significantly for different plant realizations. This strategy results in dynamic performance that is the better than nominal MPC for many plant realizations but cannot be guaranteed better for every plant realization. This robust strategy would be preferred in the situation where reducing the variation of the performance is more important than improving the performance, that is, where *consistent* closed-loop plant behavior is required.

Figure 4.19 shows the references of the controlled variable and the manipulated variables and compares the set points calculated by the steady-state optimization during the transient with the different robust MPC. We observe that in both Figure 4.19 (a) and Figure 4.19 (b), all the set points calculated are the same as their reference values (in spite of the different plant realizations and different dynamic performance in the objective of robust MPC). This is because 1) u_2 is cheaper to manipulate than u_1 is to influence y, concerning the gains of the two inputs (the gain of u_2 is bigger than the gain of u_1), so the steady-state optimization chooses to change the steady state value of u_2 (instead of u_1) to maintain y at the new reference value; 2) u_2 is known to have no uncertainty in its effect on y, so the steady-state settling point is constant in this case (while the dynamic responses of the system may be different for different plant realizations and different dynamic performance in the objective of robust MPC.

One hundred cases of the closed-loop system with the robust MPC using the different objective functions have been run with Monte Carlo sampling of the plant realizations. The results are summarized in Table 4-6. We find that minimizing the nominal performance leads to the worse performance for both the average performance and the worst-case performance. From this study we conclude that the robust MPC minimizing the robust performance can provide almost the same good performance for all the sampled plant realizations when differences in dynamics and uncertainty exist. When consistent quality is essential, the robust MPC with robust objective could provide substantially better dynamic behaviour.



(a) Set points with robust MPC minimizing the nominal performance, equation (4.68)



(b) Set points with robust MPC minimizing the robust performance, equation (4.69)

——— Nominal Realization ·········· Plant Realization 1 – – – Plant realization 2 –·–· Reference x-axis = time (min)
y_{sp} = Set point of C_A (kmole/m^3) $u_{sp,1}$ = Set point of nominal C_{A0} by adjusting flow F_A (kmole/m^3) $u_{sp,2}$ = Set point of F_c (m^3/min)

Figure 4.19 Comparing robust MPC with different objectives
– The system set points after reference step changes of CSTR control system 4

Table 4-6 Monte-Carlo Simulation Results of case study in Figure 4.17

	Average IAE [1]	Worse IAE
Robust MPC minimizing nominal performance	0.2167	2.3551
Robust MPC minimizing robust performance	0.1862	0.1864

Note: (1) IAE denotes Integrated Absolute Error.

4.4 Conclusions

In this chapter, we extend the robust MPC method from Chapter 3 with the three additional features that are important for applications in process control: the first is a new robust steady-state optimization method; the second is a novel steady-state deviation model developed for robust steady-state optimization with time-invariant uncertainty; and the third is a new objective function minimizing the dynamic performance robustly.

The new robust steady-state optimization method developed in this chapter includes the features that follow the developments in Chapter 3 for dynamic optimization:

1) Correlated parametric uncertainty of the closed-loop system at steady state (with deviation model, see explanation below);
2) An active set heuristic that is used to obtain the active hard bounds on manipulated variables at steady state in an iterative way;
3) Tractable solution for real-time implementation through a limited number of (convex) SOCPs.

A novel deviation model formulation is obtained by the deviation of the variables from a virtual steady state of them (determined by the latest implemented manipulated variables). This formulation is used for time-invariant uncertainty to reduce the conservativeness in uncertainty prediction by limiting the effects of plant uncertainty to *changes* in the input variables.

The proposed new objective function includes the expected dynamic performance and the variances of the controlled variables. The new objective function is convex and quadratic with respect to the degrees of freedom **t**, so the resulting robust MPC formulation is still a SOCP that can be relatively easily solved in real time.

The case study results shown in Section 4.3 demonstrate that:

1) The new robust steady-state optimization method using closed-loop uncertainty is better than the method using open-loop uncertainty because it is more accurate in modelling the closed-loop system and less conservative in determining the set points of the controlled variables. When integrated with the robust trajectory optimization (robust MPC), the steady-state closed-loop method achieves better dynamic performance than the open-loop method.
2) The robust method (including steady-state and trajectory optimization) outperforms the nominal method (including steady-state and trajectory optimization) on handling the constraints on controlled variables.
3) In the situations where the uncertain non-square system has alternative ways to adjust manipulated variables, optimizing a robust measure of robust dynamic performance is better than optimizing the nominal dynamic performance. This is because the new method includes uncertainty in evaluating future performance and enables the controller to trade off uncertainty, economics, and nominal feedback dynamics.

This chapter has tailored the basic robust MPC method in Chapter 3 to applications in process control. The next chapter will tailor the basic robust MPC method for applications in supply chain optimization, which is another important uncertain dynamic system with feedback.

Chapter 5

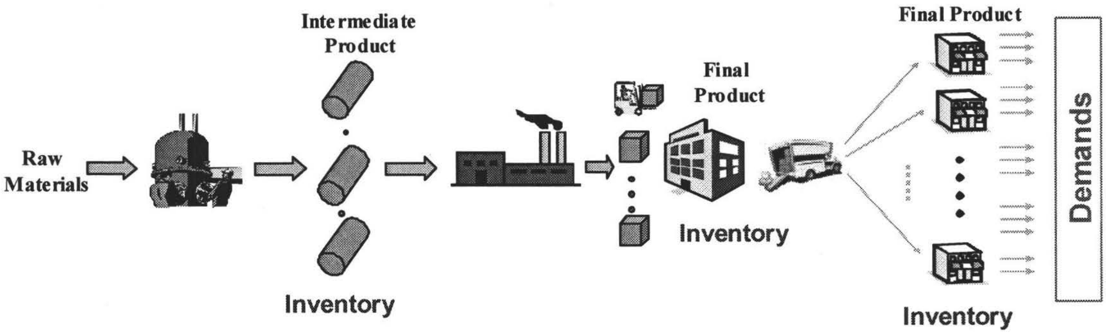
Robust MPC for Supply Chain Optimization

This chapter addresses the application of the robust MPC method to the optimization of supply chain operation under uncertainty. The general robust MPC framework developed in Chapter 3 is adapted for the application, with the formulation tailored for supply chain optimization. The need to tailor the formulation is introduced through a real supply chain optimization problem from industry. The industrial supply chain system contains manufacturing of the intermediate and final products, transportation of the final products and storage units located in different parts of the supply chain, so its structure is typical of many supply chain systems. As a result, the method developed for this system is applicable to many other supply chain systems.

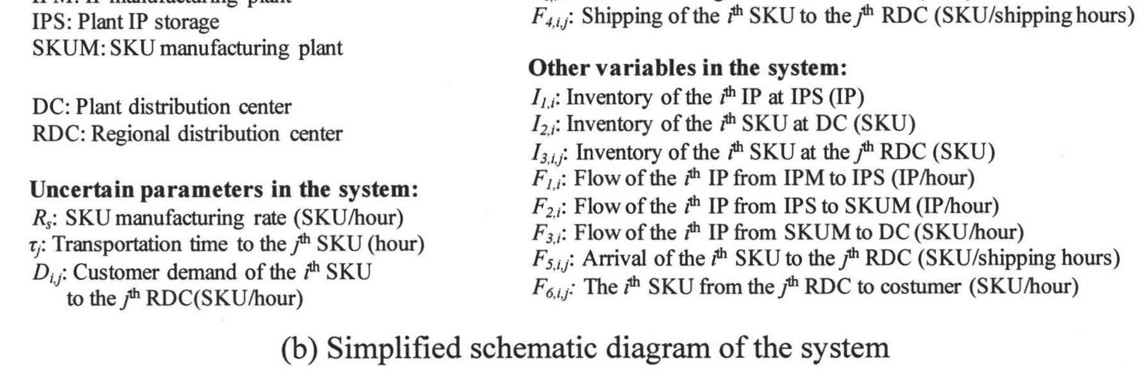
Section 5.1 introduces the industrial supply chain system and the goal of the optimization. Section 5.2 describes the modeling of the dynamic system with a discrete, state-space model and the nominal MPC formulation. Section 5.3 discusses the robust MPC formulation with emphasis on: 1) the modeling of parametric uncertainty; 2) a bilevel formulation for closed-loop optimization; 3) a tailored chance constrained program for the non-normally distributed customer demands. Section 5.4 discusses the supply chain model for simulation, which enforces integer values when needed. Section 5.5 shows the advantage of the robust MPC over nominal MPC on reducing the back orders though the case study results of the industrial supply chain optimization problem. Issues of the tuning and the computational complexity of the robust MPC method are also addressed in Section 5.5. Section 5.6 summarizes the chapter with conclusions.

5.1 The Industrial Supply Chain Optimization Problem

Let's see the industrial multi-echelon supply chain system in Figure 5.1 first. The sketch of the system is shown in Figure 5.1(a) and its schematic diagram (with defined symbols and variables) is shown in Figure 5.1 (b).



(a) The sketch of the system



(b) Simplified schematic diagram of the system

Figure 5.1 The industrial multi-echelon supply chain system

The design of this supply chain was completed after discussions with an industrial company, so that many of the specific parameters, e.g., decision frequency, are close to the parameters used by the company. Naturally, these parameters could be changed, but this design gives a reasonable basis for evaluating the robust optimization methods developed in this thesis.

In this system, different types of unlimited raw materials are processed in the plant IPM into different Intermediate Products (denoted by IP). The different types of raw materials and the associated products are indexed by i=1,...,n_i. P_i (IP/84 hours) denotes the manufacturing rates of the IP that is associated with the i^{th} material type (hereafter, referred to as the i^{th} IP). The decisions of the IP manufacturing are made once every 84 hours or 3.5 days.

The intermediate products are stored in the Plant IP storage (denoted by IPS). $I_{1,i}$ (IP) is the inventory of the i^{th} IP. The intermediate products are then processed into different final products, called Stock Keeping Units (denoted by SKU), in plant SKUM. $F_{2,i}$ (IP/hour) denotes the processing rate of i^{th} IP into the associated SKU (hereafter, referred to as the i^{th} SKU), which is determined by the machine running time $T_{s,i}$ for the i^{th} IP. The decisions on the SKU manufacturing are made once every day.

The SKUs are sent to the plant Distribution Center (denoted by DC), where the inventory of the i^{th} SKU is $I_{2,i}$ (SKU). Then, they are shipped by truck to different Regional Distribution Centers (denoted by RDC), which are indexed by j =1,...n_j. $F_{4,i,j}$ (SKU/shipping hours) denotes the quantity of the i^{th} SKU to be shipped to the j^{th} RDC and τ_j denotes the transportation time for the shipment from DC to the j^{th} RDC. The decisions on the shipments are made at different frequencies for different regional centers, and the frequencies are between once per day to three times per day. The unit cost of SKU shipment is constant, because if the SKUs do not fill up a truck, other products can be transported to fill the truck. $I_{3,i,j}$ (SKU) denotes the inventory of the i^{th} SKU at the j^{th} RDC.

The SKUs are sold to the customers at different RDCs. The customer demands $D_{i,j}$ (SKU/simulation period) of the i^{th} SKU to the j^{th} RDC can be estimated from historical data, but a large variability exists in demands. If a demand of the i^{th} SKU at the j^{th} RDC cannot be satisfied immediately, a stock-out occurs, and the unfilled part of the order is recorded as a back order $O_{i,j}$ (SKU), which must be satisfied by later shipments before new demands will be satisfied.

In all cases considered, plants IPM and SKUM have sufficient capacity to satisfy the total customer demands over the time horizon (14 days in the cases studies in this Chapter). However, a capacity limit can be encountered during a short period of the horizon (e.g., 1 day). The maximum storage capacities are unlimited, but naturally have lower bounds of zero.

The goal of this supply chain optimization is to minimize the total system cost of the supply chain (including inventory cost, manufacturing cost and transportation cost) while satisfying customer demands (if possible) by making decisions on the IP manufacturing rates P_i, SKU manufacturing machine running times $T_{s,i}$ (hour), and the SKU transportation quantities $F_{4,i,j}$. The uncertainties in the system include the SKU manufacturing rate R_s (SKU/hour), the product transportation time τ_i (hour) and the customer demands $D_{i,j}$.

We make the following assumptions for the model used in optimization based on the real industrial problem and the needs for real-time computing:

1) In each manufacturing decision interval, the IP or SKU manufacturing is continuous and the manufacturing rates are constant. The production scheduling, control and optimization (if needed) are assumed to be solved locally, which are out of the scope of this supply chain optimization. This decomposition is typical (e.g., Pinedo, 2000).
2) The SKU shipments to RDC only occur at predetermined time points each day. For example, if the shipment is once every 12 hours, it can only occur at 12am and 12pm.
3) The daily quantities of customer demands are assumed to be continuous, and the demand rate is assumed to be constant within each day.
4) Fractional numbers are allowed in the solution of the supply chain optimizer and are rounded to an integer for implementation.

5.2 The Nominal MPC Formulation

This section gives the detailed information on the nominal MPC formulation of this supply chain optimization problem. The nominal MPC will be extended in the following Section 5.3 to include robust performance using the method developed in Chapter 3. Also, This nominal MPC will be used for comparison with the robust MPC in case studies in Section 5.5. The nominal MPC is composed of three parts: the nominal state-space dynamic model of the supply chain system, the economic objective function and the constraints on the variables. We start with the modeling of a nominal state-space dynamic model for the system.

5.2.1 The discrete-time nominal state-space model

5.2.1.1 Handling of different feedback and implementation periods

In process control, often the controlled variables are measured and the manipulated variables are decided at the same time point, so the sampling period and the decision implementation period are same. We can choose this period to be simulation time period (the length of a discrete time step) of the discrete system, and then depict this system with a canonical discrete state-space model. In this supply chain optimization problem, however, the implementation periods of the decision variables are not all the same, and they are not consistent with the frequency of measuring the different feedback information. So we have to determine the simulation time period in another way.

It is logical to execute the supply chain optimizer only when new measured information on inventories is available. Since all the inventories in this system are measured once per 24 hours, the controller (optimizer) execution period of this system, ΔT_c, is taken to be 24 hours. However, the decisions on the IP manufacturing (P_i) are determined once per 84 hours, the decisions on the SKU manufacturing ($T_{s,i}$) are determined once per 24 hours, and the decisions on the SKU transportation ($F_{4,i,j}$) are determined at different frequencies for different RDCs. Here, we choose the simulation period, ΔT, to be the greatest common devisor of the different periods so that all the

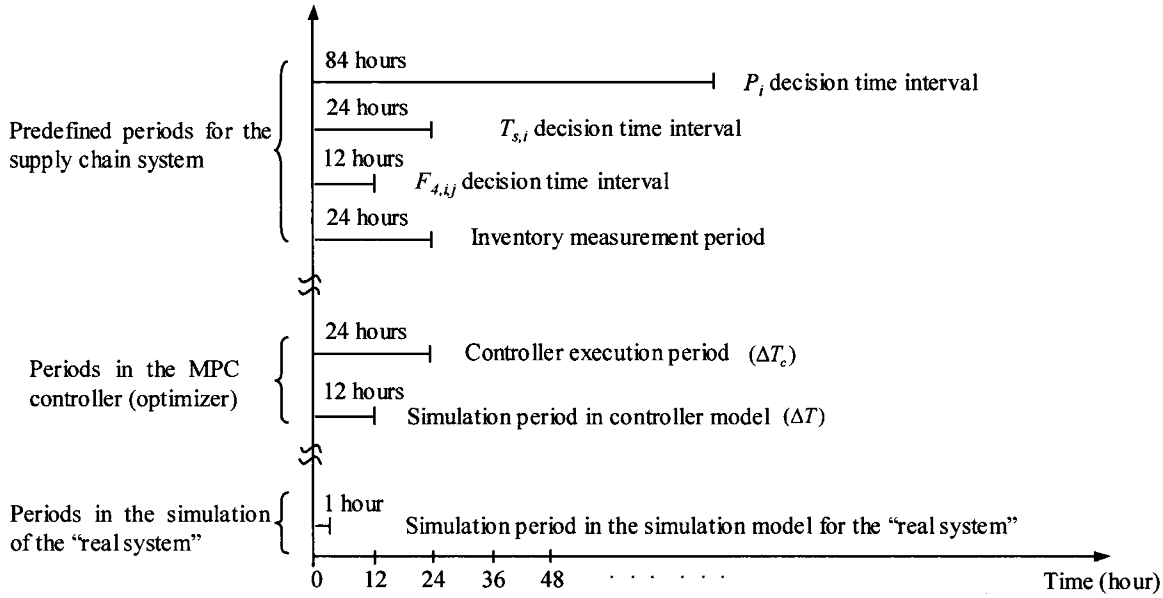


Figure 5.2 The different time periods for the supply chain system

periods can be addressed with a canonical discrete state-space model with this ΔT. For example, since the shortest $F_{4,i,j}$ decision periods are 12 hours, we set ΔT=12 hours. We also choose 1 hour to be the simulation period for the simulation model that presents the behavior of the real supply chain (refer to Section 5.4 for more discussion on this model). The different periods set for the supply chain system are shown in Figure 5.2.

5.2.1.2 Dynamic model based on mass balances

Now, we discuss the nominal dynamic model used by the controller. This model can be built based on the dynamic mass balance for each inventory in the supply chain system, which is discretized as given in the following.

IPS inventory:

$$I_{1,i,k+1} = I_{1,i,k} + F_{1,i,k}\Delta T - F_{2,i,k}\Delta T , \qquad i = 1,\dots,n_i \tag{5.1}$$

where the k subscript denotes the sequence number of sampled time steps.

The flow of the i^{th} IP coming out of plant IPM, $F_{1,i,k}$, is determined by the manufacturing decision of the i^{th} IP, P_i, at the current time step or a prior time step, according to the following relationship

$$F_{1,i,k} = P_{i,m_1} / \Delta T_P, \qquad i = 1,\ldots,n_i \tag{5.2}$$

where P_{i,m_1} is the decision variable giving the total i^{th} IP production during its decision time period ΔT_P (equal to 84 hours) beginning at time step m_1 $(m_1 \le k)$.

The flow of the i^{th} IP sent to the plant SKUM, $F_{2,i,k}$, is determined by the SKU manufacturing decision, processing time $T_{s,i}$, at the current time step or a prior time step, according to the following relationship

$$F_{2,i,k} = C_{IP-SKU,i} R_{s,i,k} T_{s,i,m_2} / \Delta T_c, \qquad i = 1,\ldots,n_i \tag{5.3}$$

where T_{s,i,m_2} denotes the decision variable that determines the i^{th} SKU manufacturing time during its decision time period ΔT_c (equal to 24 hours) at time step m_2 $(m_2 \le k)$; $R_{s,i,k}$ is the (uncertain) production rate of the i^{th} SKU at time step k; $C_{IP-SKU,i}$ converts the units of the i^{th} IP i to the i^{th} SKU, that is, how much of the intermediate product is required for one SKU.

<u>DC inventory:</u>

$$I_{2,i,k+1} = I_{2,i,k} + F_{3,i,k}\Delta T - \sum_{j=1}^{n_j} F_{4,i,j,k}\Delta T, \qquad i = 1,\ldots,n_i, \tag{5.4}$$

where the SKU mass flow $F_{3,i,k}$ is equal to the IP flow $F_{2,i,k}$ because there is no inventory accumulated in the manufacturing plant SKUM, so

$$F_{3,i,k} = F_{2,i,k} / C_{IP-SKU,i} = R_{s,i,k} T_{s,i,m_2} / \Delta T_c, \qquad i = 1,\ldots,n_i \tag{5.5}$$

RDC inventory:

$$I_{3,i,j,k+1} = I_{3,i,j,k} + F_{5,i,j,k}\Delta T - F_{6,i,j,k}\Delta T, \qquad i=1,\dots,n_i, \;\; j=1,\dots,n_j \tag{5.6}$$

$$O_{i,j,k+1} = O_{i,j,k} + D_{i,j,k}\Delta T - F_{6,i,j,k}\Delta T, \qquad i=1,\dots,n_i, \;\; j=1,\dots,n_j \tag{5.7}$$

Equation (5.7) denotes the back order balance. Let

$$I_{3,i,j,k}{}^* = I_{3,i,j,k} - O_{i,j\,k},$$

then equations (5.6-5.7) can be written into

$$I_{3,i,j,k+1}{}^* = I_{3,i,j,k}{}^* + F_{5,i,j,k} - D_{i,j,k}, \qquad i=1,\dots,n_i, \;\; j=1,\dots,n_j \tag{5.8}$$

where $I_{3,i,j,k+1}{}^*$ can be negative (when back orders exist), zero, or positive. The purpose of the variable $I_{3,i,j,k+1}{}^*$ is to accumulate backorders that must be serviced as soon as possible. In the case studies, orders are not lost if not satisfied immediately, and we will monitor the magnitude of backorders when evaluating the performance of supply chain optimization.

Also, the arrival shipment of the i^{th} SKU to the j^{th} RDC, comes from the departure shipment of the i^{th} SKU to the j^{th} RDC, so

$$F_{5,i,j,k} = F_{4,i,j,k-(\tau_j/\Delta T)}, \qquad i=1,\dots,n_i, \;\; j=1,\dots,n_j \tag{5.9}$$

The time delay between the departure shipment and the arrival shipment in equation (5.8) is caused by the transportation time τ_j. To maintain the model in state space form, we can express equation (5.9) in an equivalent form by introducing additional variables $S_{i,j} = (s_{1,i,j},\dots,s_{\tau_j/\Delta T,i,j})^T$ that denote the quantities of the i^{th} SKU in the transportation to the j^{th} RDC during a time step:

$$\begin{pmatrix} F_{5,i,jk} \\ s_{1,i,j,k} \\ \vdots \\ s_{\tau_j/\Delta T,i,j,k} \end{pmatrix} = \begin{pmatrix} s_{1,i,jk-1} \\ s_{2,i,j,k-1} \\ \vdots \\ F_{4,i,j,k} \end{pmatrix}, \qquad i = 1,\ldots,n_i, \quad j = 1,\ldots,n_j \tag{5.10}$$

Equations (5.1-5.5), (5.8) and (5.10) can be combined into the following state-space model

$$x_{k+1} = Ax_k + Bu_k + B_d d_{m,k} \tag{5.11}$$

where $x = \left(S_{1,1}^T,\ldots,S_{n_i,n_j}^T, I_{1,1},\ldots, I_{1,n_i}, I_{2,1},\ldots,I_{2,n_i}, I_{3,1,1}{*},\ldots,I_{3,n_i,n_j}{*}\right)^T$ contains the state variables, $u = \left(P_1,\ldots,P_{n_i}, T_{s,1},\ldots T_{s,n_i}, F_{4,1,1},\ldots F_{4,n_i,n_j}\right)^T$ contains all decision (manipulated) variables and $d_m = \left(D_{1,1},\ldots,D_{n_i,n_j}\right)^T$ denotes the predicted disturbances which are forecast customer demands.

5.2.2 The economic objective and the constraints

According to the goal of the supply chain optimization, the economic objective of the nominal MPC can be written as

$$\begin{aligned} \min_{u_k, O_{i,j,k+1}} \quad & \sum_{i,k} C_{I_1,i,k+1} I_{1,i,k+1} + \sum_{i,k} C_{I_2,i,k+1} I_{2,i,k+1} + \sum_{i,j,k} C_{I_3,i,j,k+1} I_{3,i,j,k+1} \\ & + \sum_{i,j,k} C_{S,i,j,k+1} S_{i,j,k+1} + \sum_{i,k} C_{P,i,k} P_{i,k} + \sum_{i,k} C_{T_s,i,k} T_{s,i,k} \\ & + \sum_{i,j,k} C_{F_4,i,j,k} F_{4,i,j,k} + \sum_{i,j,k} C_{O,i,j,k+1} O_{i,j,k+1} \end{aligned} \tag{5.12}$$

where $C_{I_1,i,k+1}$, $C_{I_2,i,k+1}$, $C_{I_3,i,j,k+1}$ denote the costs of inventory $I_{1,i,k}$, $I_{2,i,k}$, $I_{3,i,j,k}$ respectively; $C_{S,i,j,k+1}$ denotes the cost of the SKU in the transportation; $C_{P,i,k}$, $C_{T_s,i,k}$,

$C_{F_4,i,j,k}$ denote the costs of implementing the decisions $P_{i,k}$, $T_{s,i,k}$, $F_{4,i,j,k}$; $C_{O,i,j,k+1}$ denotes the penalty on the back order $O_{i,j,k+1}$. According to the definition of the state vector x_k and the decision vector u_k, the objective function (5.12) can be also written in the following form

$$\min_{u_k, O_{k+1}} \quad \sum_k C_{x,k+1}^T x_{k+1} + \sum_k C_{u,k}^T u_k + \sum_k C_{IO,k+1}^T O_{k+1} \tag{5.13}$$

where $C_{x,k+1}^T$, $C_{u,k}^T$, $C_{IO,k+1}^T$ contain appropriate costs and penalty coefficients, and the back order vector is defined as $O_{k+1} = \left(O_{1,1,k+1}, \ldots, O_{n_i,n_j,k+1}\right)^T$.

Now let's discuss the inequality constraints on the variables. First, all the inventories and the SKUs in transportation should be nonnegative, so

$$I_{1,i,k+1}, I_{2,i,k+1}, I_{3,i,j,k+1}, S_{i,j,k+1} \geq 0, \qquad i = 1,\ldots,n_i, \;\; j = 1,\ldots,n_j \tag{5.14}$$

According to the definition of x_k, constraints (5.14) can be written into the form of

$$x_{k+1} \geq x_{\min,k+1} - B_o O_{k+1} \tag{5.15}$$

where B_o is a diagonal matrix whose diagonal elements are 1 when associated to $I_{3,i,j,k}^*$ and 0 otherwise. Second, all the back orders should be nonnegative, so

$$O_{k+1} \geq 0 \tag{5.16}$$

Third, the decision variables are all nonnegative and are subject to upper limits, so

$$u_k \geq 0 \tag{5.17}$$

$$\sum_{i=1}^{n_i} T_{s,i,k} \leq T_{s,\max,k}, \tag{5.18}$$

$$\sum_{i=1}^{n_i} F_{4,i,j,k} \leq F_{4,\max,j,k}, \tag{5.19}$$

where the constraints (5.18) denote the total SKU manufacturing time at the *k*th controller execution period ΔT_c cannot exceed $T_{s,\max,k}$ (i.e., $T_{s,\max,k}=\Delta T_c=24$ hours), and the constraints (5.19) denote the quantities of different SKUs in a shipment to the j^{th} RDC cannot exceed the available transportation capacity, $F_{4,\max,j,k}$. Since IP manufacturing capacity is larger than any possible optimal solution, no upper bounds are imposed on P_i, $F_{1,i}$., although such bounds could be accommodated in the robust MPC.

Special modeling is required because the IP manufacturing decision P_i is made once every 84 hours and not reconsidered until 84 hours have elapsed, $F_{1,i}$, and the optimizer is executed every 24 hours. Therefore, the IP manufacturing decisions in the subsequent several time steps must be set to the values determined in an earlier optimization. To achieve this, we need to force these $F_{1,i}$ to be the value that has been determined in a previous optimization, which can be realized by setting the upper and lower bounds on these $F_{1,i}$ to be the value in the previous optimization. We can express these special bounds as

$$F_{1,\min,i,k} \leq F_{1,i,k} \leq F_{1,\max,i,k}, \qquad i = 1,\ldots,n_i, \tag{5.20}$$

For the time step when P_i are degrees of freedom that determines $F_{1,i,k}$, the above constraints denote the equipment capacity limits; for the time step when P_i has been determined in an earlier controller execution that determines $F_{1,i,k}$, the bounds $F_{1,\min,i,k} = F_{1,\max,i,k}$ and they are equivalent to the value of $F_{1,i,k}$ determined in the prior optimization.

All the above bounds on decision variables can be written in the following form:

$$u_{\min,k} \le u_k \le u_{\max,k}, \tag{5.21}$$

with the minimum and maximum values determined prior to each optimization as required to achieve the above-discussed strategy.

5.2.3 The nominal MPC formulation

The nominal MPC formulation consists of the objective function (5.13), the nominal dynamic model (5.11) and the constraints (5.15-5.16) and (5.21).

NMPC_SCO:

$$\min_{u_k, O_{k+1}} \quad \sum_k C_{x,k+1}^T x_{k+1} + \sum_k C_{u,k}^T u_k + \sum_k C_{Io,k+1}^T O_{k+1} \tag{5.22a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k + B_d d_{m,k} + e_0 \tag{5.22b}$$

$$u_{\min,k} \le u_k \le u_{\max,k} \tag{5.22c}$$

$$x_{k+1} \ge x_{\min,k+1} - B_o O_{k+1} \tag{5.22d}$$

$$O_{k+1} \ge 0 \tag{5.22e}$$

$$k = 0, \cdots, n-1$$

where n denotes the same control and prediction horizons, the additional variable e_0 in Equation (22.b) denotes the feedback information which contains the difference between the predicted and the measured inventories. Note that for this supply chain system, we assume that all the inventories can be measured at each time step, but the SKUs in transit (i.e., $S_{i,j,k}$ in state vector x_k) cannot be measured. Therefore, we set the feedback elements in e_0, which correspond to $S_{i,j,k}$, to be zero at each time step. This nominal MPC formulation, NMPC_SCO, is a Linear Program (LP) instead of a QP because the cost function is linear.

5.3 The Robust MPC Formulation

This section discusses the development of a robust MPC formulation based on the nominal MPC formulation (5.22) to explicitly address the uncertainties in the system. We can develop a closed-loop model for uncertainty using the same approach as in Chapter 3, so here we will not repeat the general development. However, some additional issues need to be addressed for the application of the robust MPC to this supply chain optimization problem, which are discussed in the following subsections respectively.

5.3.1 Description of uncertainties with uncertain parameters

The uncertainties in the supply chain system include the customer demands, the SKU manufacturing rate in the plant SKUM and the SKU transportation time. We will describe these uncertainties and show how to characterize them using uncertain parameters.

5.3.1.1 Costumer demands

The uncertainty in the customer demands can be estimated from the historical data. Figure 5.3 shows the histogram of the daily demand of the 1st SKU to the 1st RDC during years of 2004 and 2005, where we can find that the uncertain demand does not obey a normal distribution but a distribution close to exponential distribution (Balakrishnan and Basu, 1996). The demands of other products (to other RDCs) follow similar distributions (see Appendix H for the histograms of their demands).

Furthermore, correlations exist between the uncertain demands. Figure 5.4 shows the normalized covariance matrix of the demands of the 1st and 2nd SKUs to the 1st and the 2nd RDCs in the two successive days, which is calculated according to the historical data from years of 2004 and 2005 using the standard technique (Box et al., 2008). We observe that the correlations between the demands in the same day are more significant, especially between demands of the 1st SKU to the two RDCs, although the correlations between the demands in different days are much weaker.

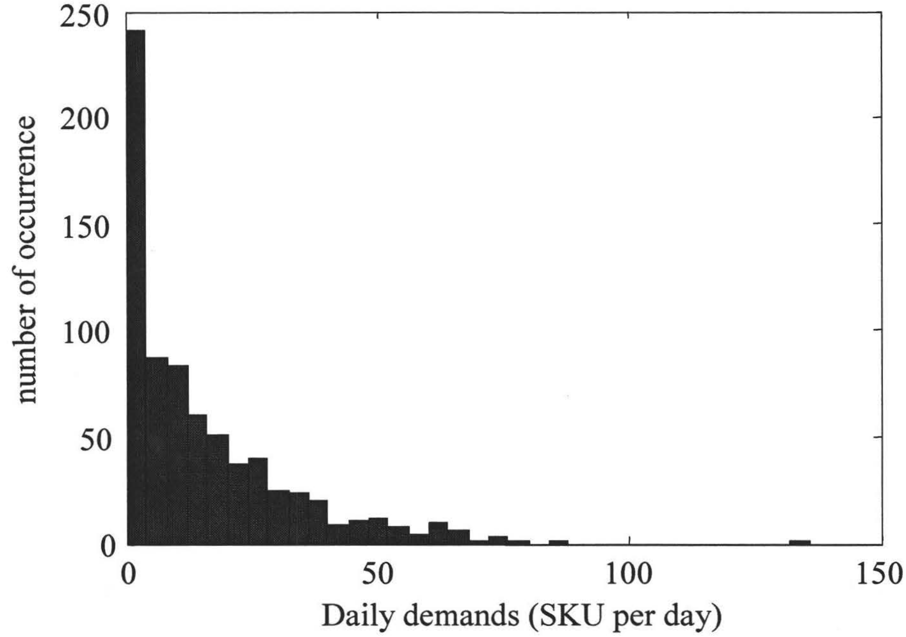


Figure 5.3 The histogram of the daily demand of the 1st SKU to the 1st RDC

	$D_{1,1,1}$	$D_{1,2,1}$	$D_{2,1,1}$	$D_{2,2,1}$	$D_{1,1,2}$	$D_{1,2,2}$	$D_{2,1,2}$	$D_{2,2,2}$
$D_{1,1,1}$	1.00	0.50	0.29	0.23	0.24	0.31	0.16	012
$D_{1,2,1}$	0.50	1.00	0.21	0.26	0.16	0.31	0.14	0.09
$D_{2,1,1}$	0.29	0.21	1.00	0.21	0.14	0.12	0.10	0.05
$D_{2,2,1}$	0.23	0.26	0.21	1.00	0.07	0.05	0.10	0.11
$D_{1,1,2}$	0.24	0.16	0.14	0.07	1.00	0.50	0.29	0.23
$D_{1,2,2}$	0.31	0.31	0.12	0.05	0.50	1.00	0.21	0.26
$D_{2,1,2}$	0.16	0.14	0.10	0.10	0.29	0.21	1.00	0.21
$D_{2,2,2}$	0.12	0.09	0.05	0.11	0.23	0.26	0.21	1.00

Figure 5.4 Normalized covariance matrix of demands of the SKUs to RDCs ($D_{i,j,k}$ denotes demand of the i^{th} SKU to the j^{th} RDC in day k)

The uncertainty in customer demands can be directly described by the uncertainty in parameter d_k in the process model (5.11). According the above discussion, one way to characterize d_k uncertainty is to use the analytical expression of a joint (multi-variate) exponential distribution. However, the existing work on multi-variate exponential distribution (e.g. Bemis et. al., 1972; Marshall and Olkin, 1967; Proschan and Sullo, 1976) is limited for specific correlations between the variables only. So in this thesis, we propose to characterize the demand uncertainty with sampling from the historical data and use the results to build the chance-constrained program. The details of this method are presented in Section 5.3.3.

5.3.1.2 SKU manufacturing rate

We will take estimates of the SKU manufacturing rate $R_{s,k}$ as their nominal value $\pm$ 25% (from 13.3 to 22.2 SKU/hour) with 90% confidence. We assume that $R_{s,k}$ obeys a normal distribution and that there is no correlation between $R_{s,k}$ in different days. Because the manufacturing rate appears on the left-hand side in equations (5.4-5.5), the uncertainty in $R_{s,k}$ leads to the uncertainty in the process gain B in the process model (5.11).

5.3.1.3 SKU transportation time

We know the approximate ranges of the SKU transportation time $\tau_{j,k}$ to different RDCs with 90% confidence from discussions with the supplier of the case study. We assume that $\tau_{j,k}$ obeys a normal distribution and that there is no correlation between $\tau_{j,k}$ in different shipments.

The uncertainty in SKU transportation time makes the *structure* of the model uncertain, because it changes the number of states in (5.10). To model the uncertainty would require disjunctive programming and integer variables, and no method is available for modeling continuous uncertainty across multiple models. Therefore, we propose to use an alternative disjunctive modeling approach with uncertain parameters to approximate this structural uncertainty. The concept is illustrated in Figure 5.5. The shipment from DC is

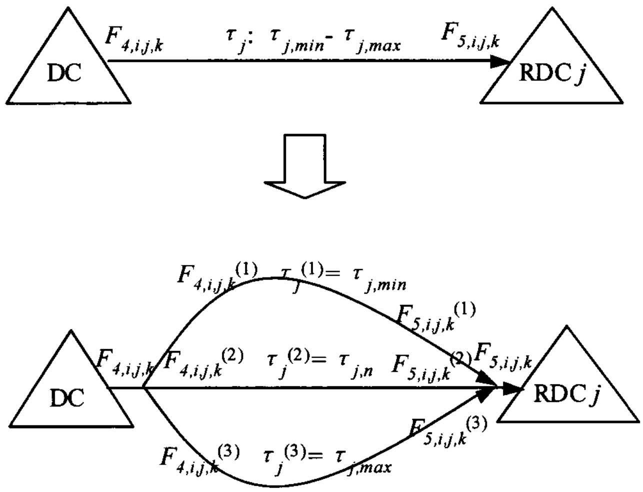


Figure 5.5 Approximate disjunctive model for SKU transportation time uncertainty

assumed to reach the j^{th} RDC through several, in this study three, different virtual routes with different but known transportation times, $\tau_j^{(1)}=\tau_{j,min}$ (minimum transportation time with 90% confidence, typically 132 hours), $\tau_j^{(2)}=\tau_{j,n}$ (nominal transportation time, typically 144 hours) and $\tau_j^{(3)}=\tau_{j,max}$ (maximum transportation time with 90% confidence, typically 156 hours). (Refer to Appendix H for a more complete information of $\tau_{j,min}$, $\tau_{j,n}$ and $\tau_{j,max}$ for different RDCs.) Then, we have

$$F_{4,i,j,k}{}^{(l)} = \beta_l F_{4,i,j,k}\,, \quad 0 \le \beta_l \le 1\,, \ \sum_l \beta_l = 1 \tag{5.23}$$

$$F_{5,i,j,k}{}^{(l)} = F_{4,i,j,k-(\tau_j^{(l)}/\Delta T)}{}^{(l)} \tag{5.24}$$

$$F_{5,i,j,k} = \sum_l F_{5,i,j,k}{}^{(l)} \tag{5.25}$$

$$l = 1, 2, 3$$

where equation (5.24) describes the time delay between the departure shipment $F_{4,i,j,k}^{(l)}$ and the arrival shipment $F_{5,i,j,k}^{(l)}$ due to the transportation time $\tau_j^{(l)}$, and this equation can be transformed into a state-space model of the form of equation (5.10) with additional variables. Note the ratios β_l are uncertain parameters, so the uncertainty in SKU transportation time can be approximately described by the uncertain elements in B that depend on β_l. In this chapter, we assume the nominal values of different β_l are β_1=0, β_2=1, β_3=0, and the uncertain values of β_l are generated as follows:

1) Generate values for 3 random variables β_1^*, β_2^*, β_3^*, each of which obtains nominal distribution with mean 0.5 and standard deviation 1/6.
2) Calculate each β_l by the formula $\beta_l = \beta_l/(\beta_1^*+\beta_2^*+\beta_3^*)$ (so that each β_l is between 0 and 1 and their summation equals to 1).

The case study results will demonstrate that this approximation provides good supply chain optimization for the different cases.

5.3.2 Closed-loop optimization with approximating inner QP problems

The robust MPC formulation in Chapter 3 can be adapted for the supply chain robust optimizer. Here, we present the robust controller formulation, including the change in the formulation to give an inner QP problem for the robust supply chain optimization, even though the nominal objective function is linear.

With the presence of uncertainty described in Section 5.3.1, we can use the following model to describe the uncertain supply chain system,

$$x_{r,k+1} = Ax_{r,k} + B_{r,k}u_{r,k} + B_{dr}d_{r,k} + e_0 \tag{5.26}$$

where $u_{r,k}$, $x_{r,k+1}$, $e_{r,k+1}$, denote the decision (manipulated) variables, state variables and feedback variables, respectively. The model (5.26) can be built in the same approach we followed for the nominal model (5.11). However, additional state variables are needed to model the uncertainties in the transportation time (as we discussed in Section 5.3.1.3). The parameters $B_{dr,k}$ and $d_{r.k}$ in the model (5.26) are uncertain.

The robust MPC for this supply chain optimization problem addresses the closed-loop uncertainty of the system. So, its formulation includes not only the uncertain system model (5.26); it also includes the effects of the controller (optimizer) on the system dynamics during each time step in the future. As in Chapter 3, we will use a nominal MPC for the future controllers in the model. Thus, the robust MPC requires to solving the following bilevel stochastic optimization problem RMPC_SCO-CL

RMPC_SCO-CL:

$$\min_{\tilde{\mathbf{x}}_{sp,k+1},\tilde{\mathbf{u}}_{sp,k},O_{k+1}} \quad \sum_k C_{x,k+1}^T x_{k+1} + \sum_k C_{u,k}^T u_k + \sum_k C_{Io,k+1}^T O_{k+1} \tag{5.27a}$$

$$\text{s.t.} \quad u_{r,k} = NMPC\_SCO^*(x_{r,k}, e_{r,k}, \mathbf{d}_{m,k}, \tilde{\mathbf{x}}_{sp,k+1}, \tilde{\mathbf{u}}_{sp,k}) \tag{5.27b}$$

$$x_{r,k+1} = Ax_{r,k} + B_{r,k}u_{r,k} + B_{dr}d_{r,k} + e_0 \tag{5.27c}$$

$$e_{r,k+1} = x_{r,k+1} - (Ax_{r,k} + Bu_{r,k} + B_{dr,k}d_{r,k}) \tag{5.27d}$$

$$u_{\min,k} \le u_{r,k} \le u_{\max,k} \tag{5.27e}$$

$$x_{r,k+1} \ge x_{\min,k+1} - B_o O_{k+1} \tag{5.27f}$$

$$O_{k+1} \ge 0 \tag{5.27g}$$

For all $B_{d,k}$, $d_{t,k}$ in uncertainty region and $k = 0,\ldots, n$-1

where equation (5.27a) means that the robust MPC is to minimize the nominal cost of the system. Equation (5.27c) denotes the uncertain system model with the feedback information e_0. Equation (5.27d) denotes the feedback scheme where the feedback information is the difference between the real states and the nominal states. Equations (5.27e-5.27g) impose the bounds on the variables.

Equation (5.27b) means that the "simulated" control decisions in the future horizon are determined using a nominal MPC formulation, so equation (5.27b) denotes the embedded inner optimization problem. If we use the nominal MPC formulation NMPC_SCO (formulation (5.22) discussed in Section 5.2.3) in equation (5.27b), then the inner optimization problem is an LP. Thus, we cannot solve the bilevel stochastic optimization problem RMPC_SCO-CL using the method developed in Section 3.2 of Chapter 3, which is based on modeling the future control action with QP. Therefore, we need a modified nominal MPC formulation in form of QP to take advantage of the method developed in Chapter 3. We call the modified formulation NMPC_SCO* and show it as the following.

NMPC_SCO*:

$$\min_{x_{sp,k+1},u_{sp,k}} \sum_k (x_{k+1}-x_{sp,k+1})^T Q(x_{k+1}-x_{sp,k+1}) + \sum_k (u_k - u_{sp,k})^T R(u_k - u_{sp,k}) \tag{5.28a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k + B_d d_{m,k} + e_0 \tag{5.28b}$$

$$u_{\min,k} \le u_k \le u_{\max,k} \tag{5.28c}$$

$$k = 0,\ldots,n-1$$

where $x_{sp,k+1}$. $u_{sp,k}$ denote the desired values of the states and the decision variables respectively. Here, the objective (5.28a) is a quadratic function of the deviations of the state and decision variables from their desired values instead of the linear cost functions problem NMPC_SCO, so the problem NMPC_SCO is a QP instead of LP. Q and R are the weighting matrices that can be tuned off-line. We do not include the bounds on the state variables explicitly within the inner QPs because these bounds are addressed with constraints (5.27f) that are outside the inner QPs.

Therefore, the decisions determined at the future kth time step, $u_{r,k}$, are determined by the controller at that time step that solves the QP problem in the form of NMPC_SCO* with the initial conditions $x_{r,k}$, $e_{r,k}$, $\mathbf{d}_{m,k}=(d_{m,k}^T,\ldots,d_{m,k+n-1}^T)^T$ and the desired states and decisions $\tilde{\mathbf{x}}_{sp,k+1}=(x_{sp,k+1}^T, \ldots, x_{sp,k+n}^T)^T$, $\tilde{\mathbf{u}}_{sp,k}=(u_{sp,k}^T, \ldots, u_{sp,k+n-1}^T)^T$. $\tilde{\mathbf{x}}_{sp,k+1}$, $\tilde{\mathbf{u}}_{sp,k}$ are the degrees of freedom of the outer level of the bilevel problem RMPC_SCO-CL. The problem RMPC_SCO-CL can be transformed (approximately) using the approach developed in Chapter 3 by a limited number of linear stochastic optimization problems in the following form

RMPC_SCO-CLT:

$$\min_{\mathbf{t},\mathbf{O}} \quad \widetilde{C}_x^T \mathbf{x} + \widetilde{C}_u^T \mathbf{u} + \widetilde{C}_{IO}^T \mathbf{O} \tag{5.29a}$$

$$\text{s.t.} \quad \mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u \boldsymbol{\omega} \tag{5.29b}$$

$$\mathbf{x}_r = L_{xr}\mathbf{t} + M_{xr}\boldsymbol{\theta} + N_x \boldsymbol{\omega} \tag{5.29c}$$

$$(\mathbf{I} - \mathbf{I}_{\delta}) \cdot \mathbf{t} = \mathbf{u}_c \tag{5.29d}$$

$$\mathbf{u}_{\max} \geq \mathbf{u}_r \geq \mathbf{u}_{\min} \tag{5.29e}$$

$$\mathbf{x}_r \geq \mathbf{x}_{\min} - \widetilde{B}_o \mathbf{O} \tag{5.29f}$$

$$\mathbf{O} \geq 0 \tag{5.29g}$$

For all L_{ur}, M_{xur}, L_{xr}, M_{xr}, $\boldsymbol{\omega}$ in uncertainty region

where

1) Equation (5.29a) denotes the objective of the optimization is to minimize the economic cost of the states and decisions as well as the penalties on the back orders. The degrees of freedom $\widetilde{x}_{sp,k+1}$, $\widetilde{u}_{sp,k}$ are changed into $\mathbf{t} = \left(t_0^T, \cdots, t_{n-1}^T\right)^T$ (which are linear combinations of $\widetilde{x}_{sp,k+1}$, $\widetilde{u}_{sp,k}$) to prevent ill-conditioning. The process variables are $\mathbf{x} = (x_1^T, \ldots, x_{n+1}^T)^T$, $\mathbf{u} = (u_0^T, \ldots, u_n^T)^T$, $\mathbf{O} = (O_1^T, \ldots, O_{n+1}^T)^T$ and coefficients are $\widetilde{C}_x^T = (C_{x,1}^T, \cdots, C_{x,n+1}^T)^T$, $\widetilde{C}_u^T = (C_{u,0}^T, \cdots, C_{u,n}^T)^T$, $\widetilde{C}_{IO}^T = (C_{IO,1}^T, \cdots, C_{IO,n+1}^T)^T$.
2) Equations (5.29b-5.29d) denote the closed-loop model of the system with the known active bounds on the decisions. $\mathbf{u}_r = (u_{r,0}^T, \cdots, u_{r,n-1}^T)^T$, $\mathbf{x}_r = (x_{r,1}^T, \cdots, x_{r,n}^T)^T$, $\boldsymbol{\theta} = (x_0^T, e_0^T, \widetilde{\mathbf{d}}_m^T)^T$ where $\widetilde{\mathbf{d}}_m = (d_{m,0}^T, \ldots, d_{m,n}^T)^T$ denotes the predicted disturbance in the horizon, $\boldsymbol{\omega} = \delta \mathbf{d}_r = (d_{r,0}^T - d_{m,0}^T, \ldots, d_{r,n}^T - d_{m,n}^T)^T$ denotes the difference between the uncertain disturbances and their predicted values. Equation (5.29d) enforces the active bounds, where $\boldsymbol{\delta}$ is determined according to the known active bounds

specified in $\mathbf{u}_c$. $\mathbf{I}_\delta$, $\mathbf{u}_c$ using the active set heuristic explained in Section 3.2.4 in Chapter 3. The uncertain coefficients L_{ur}, M_{ur}, L_{xr}, M_{xr} can be obtained using the same approach developed in Section 3.2.3 of Chapter 3 and Appendix D.

3) Equations (5.29e-5.29g) impose bounds on the process variables, where $\mathbf{u}_{\min} = (u_{\min,0}^T, \cdots, u_{\min,n-1}^T)^T$, $\mathbf{u}_{\max} = (u_{\max,0}^T, \cdots, u_{\max,n-1}^T)^T$, $\mathbf{x}_{\min} = (x_{\min,1}^T, \cdots, x_{\min,n}^T)^T$ and

$$\widetilde{B}_o = \begin{pmatrix} B_o & & \\ & \ddots & \\ & & B_o \end{pmatrix}.$$

The problem RMPC_SCO-CL can be solved by solving the problem RMPC_SCO-CLT iteratively with the active set heuristic developed in Chapter 3. The next subsection will discuss the solution of the linear stochastic optimization problem RMPC_SCO-CLT as a chance-constrained program.

5.3.3 SOCP formulation with tailored chance-constrained program

The equations (5.29b-5.29c) and (5.29e-5.29f) can be combined into the following linear constraints with uncertain parameters,

$$L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \le \mathbf{u}_{\max} \tag{5.30}$$

$$L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \ge \mathbf{u}_{\min} \tag{5.31}$$

$$L_{xr}\mathbf{t} + M_{xr}\boldsymbol{\theta} + N_x\boldsymbol{\omega} \ge \mathbf{x}_{\min} - \widetilde{B}_o\mathbf{O} \tag{5.32}$$

In Chapter 3 we pointed out that the uncertain linear inequalities can be transformed into deterministic constraints using the idea of chance-constrained program. The accuracy of the transformation replies on how close the distribution of uncertain parameters to normal distribution. However, in this supply chain optimization problem, the uncertain parameter $\boldsymbol{\omega}$, which denotes the prediction errors of the uncertain customer demands in the future, has a distribution that is significantly different from normal distribution (as discussed in

Section 5.3.1). Therefore, we propose a revised chance-constrained program approach so that the resulting deterministic constraints can better approximates the uncertain linear inequalities.

For more details, let us consider the lth constraint in (5.30),

$$L_{ur,l}\mathbf{t} + M_{ur,l}\boldsymbol{\theta} + N_{u,l}\boldsymbol{\omega} \leq \mathbf{u}_{\max,l} \tag{5.33}$$

where $L_{ur,l}, M_{ur,l}, N_{u,l}$ denote the lth row of matrices L_{ur}, M_{ur}, N_u respectively and $\mathbf{u}_{\max,l}$ denotes the lth element in $\mathbf{u}_{\max}$. Assume we want to guarantee the feasibility of constraint (5.33) at the given confidence level α, then the constraint can be written as,

$$P_r(L_{ur,l}\mathbf{t} + M_{ur,l}\boldsymbol{\theta} + N_{u,l}\boldsymbol{\omega} \leq \mathbf{u}_{\max,l}) \geq \alpha \tag{5.34}$$

We assume the uncertainties in $L_{ur,l}, M_{ur,l}$ (from the uncertainties in manufacturing rate and transportation time) are independent of the uncertainties in $\boldsymbol{\omega}$ (comes from the uncertainties in demand forecast), thus we can handle them separately.

Define $r_{u,l}^{+}$ such that $P_r(N_{u,l}\boldsymbol{\omega} \leq r_{u,l}^{+}) \geq \alpha^{1/2}$, then constraint (5.34) can be transformed into

$$P_r(L_{ur,l}\mathbf{t} + M_{ur,l}\boldsymbol{\theta} + r_{u,l}^{+} \leq \mathbf{u}_{\max,l}) \geq \alpha^{1/2} \tag{5.35}$$

where we assume the uncertain parameters $L_{up,l}$, $M_{up,l}$ to obey a normal distribution. Then constraint (5.35) is equivalent to the following deterministic constraint (see Lobo et al., 1998 for more details),

$$E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\theta} + r_{u,l}^{+} + \Phi^{-1}(\alpha^{1/2}) \| V_{u,l}^{1/2}(\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \leq \mathbf{u}_{\max,l} \tag{5.36}$$

where $E(\cdot)$ denotes the expected value of the parameters in the brackets, $\Phi^{-1}(\alpha^{1/2})$ denotes the inverse cumulative probability function of normal distribution, $V_{u,l}$ denotes the covariance matrix of $(L_{ur,l}, M_{ur,l}, 1)$, which can be obtained using the method discussed in Chapter 3.

If we transform all the uncertain linear inequalities (5.30-5.32) in the same way, the problem RMPC_SCO-CLT becomes problem RMPC_SCO-CLTSOCP, a deterministic SOCP in the following form

RMPC_SCO-CLTSOCP:

$$\min_{\mathbf{t},\mathbf{O}} \quad \widetilde{C}_x^T \mathbf{x} + \widetilde{C}_u^T \mathbf{u} + \widetilde{C}_{IO}^T \mathbf{O} \tag{5.37a}$$

$$\text{s.t.} \quad \left.\begin{aligned} & E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\theta} + r_{u,l}^{+} \\ & \quad + \Phi^{-1}(\alpha^{1/2}) \| V_{u,l}^{1/2} (\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \le \mathbf{u}_{\max,l} \\ & E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\theta} + r_{u,l}^{-} \\ & \quad + \Phi^{-1}(\alpha^{1/2}) \| V_{u,l}^{1/2} (\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \ge \mathbf{u}_{\min,l} \end{aligned}\right\} \; l = 1, \cdots, n_u n \tag{5.37b, 5.37c}$$

$$\begin{aligned} & E(L_{xr,l})\mathbf{t} + E(M_{xr,l})\boldsymbol{\theta} + r_{x,l}^{-} \\ & \quad + \Phi^{-1}(\alpha^{1/2}) \| V_{x,l}^{1/2} (\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \ge \mathbf{x}_{\min,l} - \widetilde{B}_o \mathbf{O} \end{aligned}, \quad l = 1, \cdots, n_x n \tag{5.37d}$$

$$(diag(\mathbf{I}) - \boldsymbol{\delta}) \cdot \mathbf{t} = \mathbf{u}_c \tag{5.37e}$$

$$\mathbf{O} \ge 0 \tag{5.37f}$$

where $L_{xr,l}, M_{xr,l}$ denote the lth row of matrices L_{xr}, M_{xr} respectively and $\mathbf{u}_{\min,l}$, $\mathbf{x}_{\min,l}$, denotes the lth element in $\mathbf{u}_{\min,}$, $\mathbf{x}_{\min}$ respectively, $V_{x,l}$ denotes the covariance matrix of $(L_{xr,l}, M_{xr,l}, 1)$, $r_{u,l}^{-}$, $r_{x,l}^{-}$ are the parameters such that $P_r(N_{u,l}\boldsymbol{\omega} \ge r_{u,l}^{-}) \ge \alpha^{1/2}$, $P_r(N_{x,l}\boldsymbol{\omega} \ge r_{x,l}^{-}) \ge \alpha^{1/2}$.

Note that $r_{u,l}^{+}$, $r_{u,l}^{-}$, $r_{x,l}^{-}$ are obtained off-line numerically based on sampling of the historical data. For example, $r_{u,l}^{+}$ can be calculated as follows:

1) Randomly select p samples of the customer demands from historical data over the horizon of p time steps, and these samples form the vector $\boldsymbol{\omega}$;
2) Calculate $r_{u,l} = N_{u,l}\boldsymbol{\omega}$;
3) Repeat steps (1) and (2) for 100 groups of samples in the historical data set (from years of 2004 and 2005), and then get values of $r_{u,l}$ for all the 100 groups of samples. Pick the smallest $r_{u,l}$ value such that the percentage of the $r_{u,l}$ values less than it is above $\alpha^{1/2}$, and then set $r_{u,l}^{+}$ at this value.

With these parameters evaluated, we can solve the bilevel stochastic optimization problem RMPC_SCO-CL by solving a limited number of SOCP problems. In this thesis, RMPC_SCO-CLTSOCP was solved using an interior point optimizer, CPLEX.

5.4 The Model for Supply Chain Simulation

In the purpose of simulation case study, we need a model to represent the behavior of the real supply chain. The simulation period of this model is selected to be 1 hour for all the case studies in this chapter. Also, this model enforces all integer values where required, even though factional numbers are allowed in the commands from the optimization. It is non-trivial to round the fractional numbers into the integer variables, because the hard process constraints need to be observed when implementing commands from the optimizer. The following rules are used to implement the commands from the optimizer:

1) The manufacturing rate of the i^{th} IP, P_i, is rounded up to the nearest integer.
2) The processing rate of the i^{th} IP, F_{2},i, is rounded down to the nearest integer if there is enough inventory of the i^{th} IP ($I_{1,i}$) in IPS; Otherwise, $F_{2,i}$ is rounded to the largest integer that is feasible with the existing $I_{1,i}$.
3) The optimization tells how much SKUs to be produced within a SKU manufacturing period, but it does not indicate the sequence of manufacturing the different SKUs within the period. In the simulation model, this sequence is determined according to the inventories of different SKUs at the end of the optimization period, i.e. the smaller the inventory of a type of SKU is, the earlier that type of SKU is

manufactured. (We could choose to look at the inventories at the beginning or the end of the optimization period; here, we choose the end of the period to take the customer demand information into account.)

4) The SKU transportation quantities to different RDCs ($F_{4,i,j}$) are rounded sequentially according to the RDC indices j=1,...n_j and then the materials/product indices i=1,...n_i. For a particular $F_{4,i,j}$, we round it in the following steps:
   a) Round $F_{4,i,j}$ up to the nearest integer and call the result $F_{4,i,j}^{(a)}$
   b) If $F_{4,i,j}^{(a)}$ is not feasible with the existing available inventory of SKU i in DC, $I_{2,i}$, decrease it to the largest integer that is feasible with the existing available $I_{2,I}$; Otherwise, do not change $F_{4,i,j}^{(a)}$. We call the result after this step $F_{4,i,j}^{(b)}$.
   c) If the maximum transportation capacity to RDC j is not enough for $F_{4,i,j}^{(b)}$, decrease it to the largest integer that is feasible; otherwise, do not change $F_{4,i,j}^{(b)}$. We call the result after this step $F_{4,i,j}^{(c)}$.
   d) $F_{4,i,j}^{(c)}$ is the quantity of SKU i to be shipped to RDC j, which is used in the dynamic material balances for the inventories.

Naturally, solving and continuous optimization problem and rounding the answer for implementation is not always acceptable. We note that this rounding is generally acceptable in this problem because the number of SKU's manufactured and transported is relatively large. The results from numerous case studies reported in this chapter demonstrate the applicability of the approach for this realistic supply chain.

5.5 Case Study Results and Discussion

The simulation case studies were performed on a PC with Intel Core 2 Duo 3.0 GHz, 4GB memory and Windows Vista. The solution for the plant simulation is programmed in MATLAB 7.5, and the controller SOCP optimization problems are solved in GAMS with the interior point (barrier) solver of CPLEX 11. The data in MATLAB and CPLEX are exchanged using the interface software MATGAMS developed by Ferris (2005).

The following Section 5.5.1 discusses the case studies with 1 IP/SKU (material) type and 1 RDC, and Section 5.5.2 discusses a case study involving a more complex system with 2 IP/SKU types and 2 RDCs. Section 5.5.3 discuss the computational complexity of the method with respect to the number of IP/SKU types and RDCs. We start all cases with sufficient inventory so that the inventories remain non-zero (for uncertain demands) until the first shipments from the DC arrive at the RDC. This is to avoid back orders that are merely due to an insufficient initial inventory.

The uncertainties descriptions used in the studies are evaluated using the methods in Section 5.3.3. To evaluate the importance of considering the uncertainty in the supply chain optimizer, both nominal MPC and robust MPC methods are applied to the case study problems.

1) **Nominal MPC**

This method solves the LP problem **NMPC_SCO** (equation (5.22)) at each controller execution period.

2) **Robust MPC**

The method is the one developed in Section 5.3, which solves a series of SOCP problems at each controller execution period with the active set heuristic. The detailed steps of implementing the method is the same as described in Section 3.5 of Chapter 3 (for process control), except that the SOCP subproblems to be solved is as follows (developed in Section 5.3 of this chapter)

$$\min_{\mathbf{t},\mathbf{O}} \quad \tilde{C}_x^T \mathbf{x} + \tilde{C}_u^T \mathbf{u} + \tilde{C}_{IO}^T \mathbf{O} \tag{5.38a}$$

$$\text{s.t.} \left.\begin{array}{ll} E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\theta} + r_{u,l}^{+} & \\ \quad + \Phi^{-1}(\alpha^{1/2}) \| V_{u,l}^{1/2}(\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \le \mathbf{u}_{\max,l} & (5.38b) \\ E(L_{ur,l})\mathbf{t} + E(M_{ur,l})\boldsymbol{\theta} + r_{u,l}^{-} & \\ \quad + \Phi^{-1}(\alpha^{1/2}) \| V_{u,l}^{1/2}(\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \ge \mathbf{u}_{\min,l} & (5.38c) \end{array}\right\} l = 1,\cdots,n_u n$$

$$\begin{array}{ll} E(L_{xr,l})\mathbf{t} + E(M_{xr,l})\boldsymbol{\theta} + r_{x,l}^{-} & \\ \quad + \Phi^{-1}(\alpha^{1/2}) \| V_{x,l}^{1/2}(\mathbf{t}^T, \boldsymbol{\theta}^T, 1)^T \|_2 \ge \mathbf{x}_{\min,l} - \tilde{B}_o \mathbf{O}, & l = 1,\cdots,n_x n \end{array} \tag{5.38d}$$

$$(diag(\mathbf{I}) - \boldsymbol{\delta}) \cdot \mathbf{t} = \mathbf{u}_c \tag{5.38e}$$

$$\mathbf{O} \ge 0 \tag{5.38f}$$

Also notice that the uncertainty in the supply chain system is time-varying so the deviation model approach developed in Chapter 3 is not needed.

5.5.1 Case study with 1 IP/SKU type and 1 RDC

Case studies in this section only consider the 1st IP/SKU and the 1st RDC in the system. The controller and system parameters in all the case studies results are basically the same, expect that the nominal demands used are different and the specific realizations of the uncertainty experienced by the simulations are different, which will be described later for each specific study case.

Here we display the common parameters are used for all the case studies in this section. Table 5-1 shows the system parameters, and Table 5-2 shows the cost information specifically (which was defined arbitrarily for the study). Table 5-3 shows the different periods used in both the nominal and robust MPC controllers. Since the measurements of the inventories are available once a day, the MPC execution period, ΔT_c, is 1 day. According to the different decision implementation periods shown in Table 5-1 and the rules discussed in Section 5.2.1.1, the simulation period of the discrete model of the system used by the controller, ΔT, is selected to be 12 hours. Table 5-4 shows the weights related to the different controlled variables and decision variables in the weighing matrices Q, R of the inner QP problems, which are used to approximate the future LP optimizers. The weights of the different variables are first selected as the squares of the related costs (because we are using quadratic functions to approximate the linear economic functions). The resulting controller could be overly aggressive because the importance of the controller variables is amplified with the use of quadratic function, so the controller is tuned to reduce the aggressiveness. The case study results will demonstrate that using the weights showed in Table 5-4 in the inner problems of the robust MPC gives good supply chain optimization performance.

Table 5-1 Parameters of the system with the 1st IP/SKU and the 1st RDC

Parameter	Value
Nominal SKU manufacturing rate R_s (SKU/hour)	16.7
R_s range with 90% confidence (SKU/ hour)	13.3-22.2
Unit converting coefficients $C_{IP\text{-}SKU}$ (IP/ SKU)	6.0
Nominal SKU transportation time τ (hour)	144
τ range with 90% confidence (hour)	132-156
SKU Shipping Intervals (hour)	12
SKU transportation capacity $F_{4,\max}$ (SKU/12 hours)	40
Demands D range with 90% confidence (SKU/day)	0-38

Table 5-2 Costs defined by the author for the case study

Parameter	Value
IP Inventory cost, C_{I_1} (\$/IP/hour)	0.6
SKU Inventory cost, C_{I_2}, C_{I_3} (\$/SKU/hour)	0.1
Back order cost (penalty) C_o (\$/SKU/hour)	100
IP manufacturing cost C_P (\$/IP)	1
SKU manufacturing cost C_T (\$/SKU)	0.1
SKU Transportation cost C_{F_4} (\$/ SKU)	0.01

Table 5-3 The different periods of the nominal and robust MPC controllers

Parameter	Value
Measurement and MPC execution period, ΔT_c (day)	1
Discrete model time interval, ΔT (day)	0.5
Control and prediction horizon, n (day)	14

Table 5-4 The additional parameters of the robust MPC controller

Parameter	Value
Confidence of each chance constraint, α	90%
Q in inner problem – element for IP Inventory (IP^{-2})	0.072
Q in inner problem – element for SKU Inventory (SKU^{-2})	0.002
R in inner problem – element for IP manufacturing (IP^{-2})	1
R in inner problem – element for SKU manufacturing (SKU^{-2})	0.01
R in inner problem – element for SKU Transportation (SKU^{-2})	0.0144

5.5.1.1 Simulation results of three typical situations

Results for three typical model mismatch situations are presented here:

Case A: The forecast prediction of customer demand, SKU manufacturing rate and SKU transportation time are exactly correct. This is the no model-mismatch case.

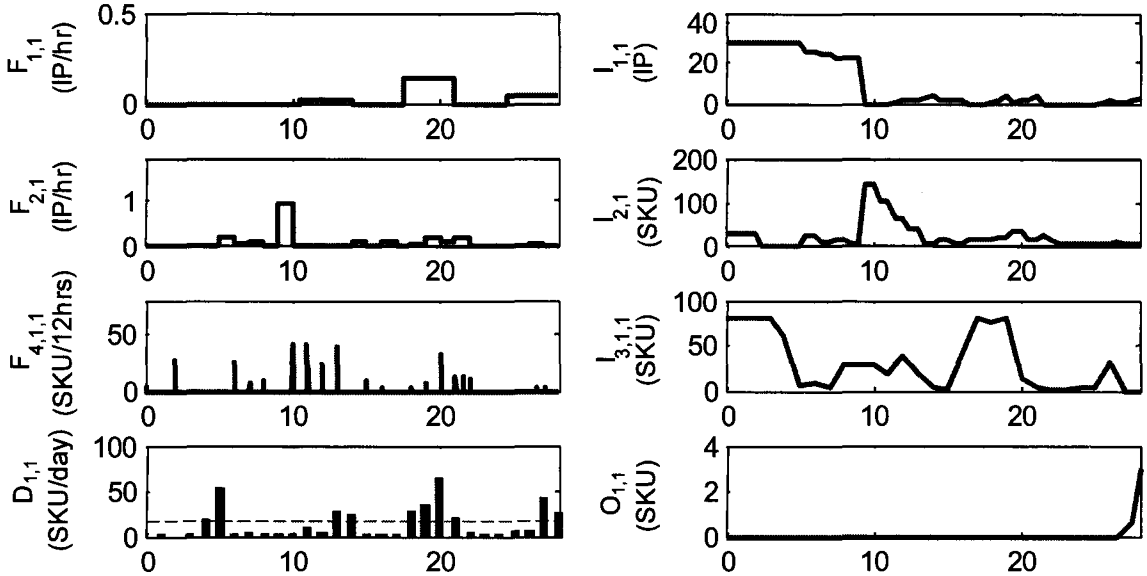
Case B: The nominal SKU manufacturing rates and SKU transportation times differ from their nominal values at each time period in a random manner; the sampled values could be outside of the 90% limits reported in Table 5-1. The forecast prediction of customer demand per day is the expected demand obtained from historical. The simulated plant experiences the actual demands taken from the demands in some successive 28 days in the year of 2006 (which could be outside of the 90% limits used in designing the controller), as well as a group of specific SKU manufacturing rates and the SKU transportation times over the 28 days that are randomly selected according to their assumed distributions.

Case C: Similar to Case B, but the demand forecast is nearly perfect, differing from the actual demand only on the 16^{th} day (the prediction is 30 SKU less then the actual demand).

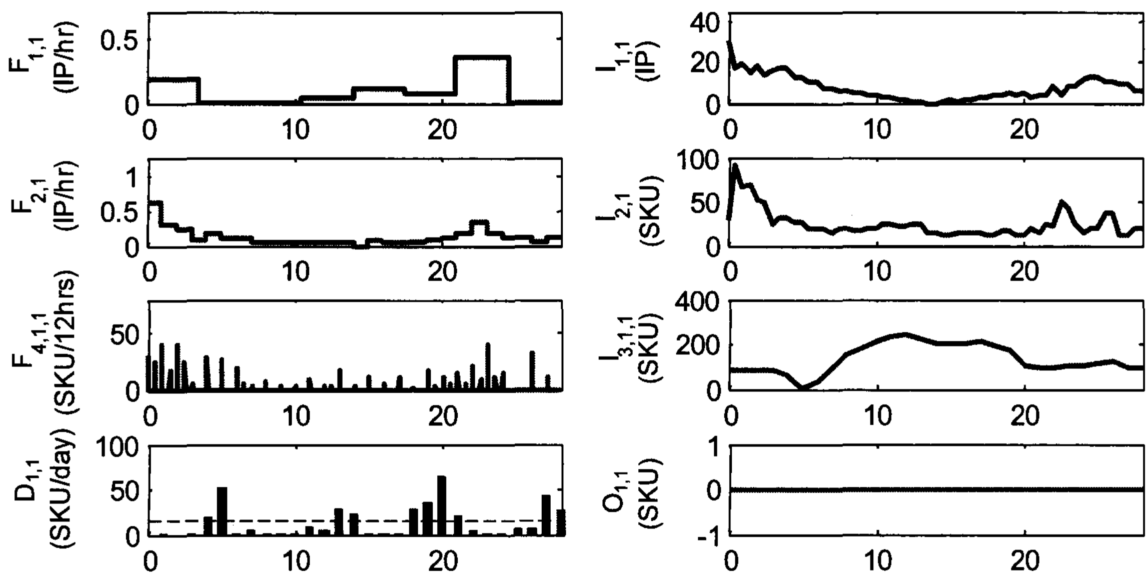
Note that Case A represents an ideal case that is very unlikely to happen in the real world, and Cases B and C are more realistic. Case B corresponds to the situation in which the daily customer demands are random and not known ahead of time, but the expected demands during a period can be estimated using historical data (e.g., retailers). Case C corresponds to the situation in which most of the products are ordered or contracted ahead of time and rush orders are possible but infrequent (e.g., wholesalers). We applied the nominal MPC and the robust MPC to all these threes situations respectively.

Figure 5.6 shows the simulation results with the nominal MPC and robust MPC in Case A. First, we notice that the variable $F_{4,1,1}$ in Figure 5.6, which denotes the SKU quantity shipped from DC to the RDC, is shown in a discrete way because we assume the shipments only occur at particular time points in the supply chain system; but $F_{4,1,1}$ itself is a continuous variable in the MPC calculation, and it is rounded to an integer variable for the implementation to the real plant (using the rule described in Section 5.4). The customer demand $D_{1,1}$ is shown with their daily demand in the figure, and the demand rate is assumed to be constant within each day.

Second, we observe in Figure 5.6 that the nominal MPC performs very well – it controls the inventory near zero while satisfying customer demands essentially all the time. (Note that the small back orders occurring around the 28^{th} day were caused by the modeling approximations we made when building the nominal model used by the controller.) Robust MPC also prevents the back orders, but it keeps a larger inventory



(a) Results with nominal MPC



(b) Results with robust MPC

$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

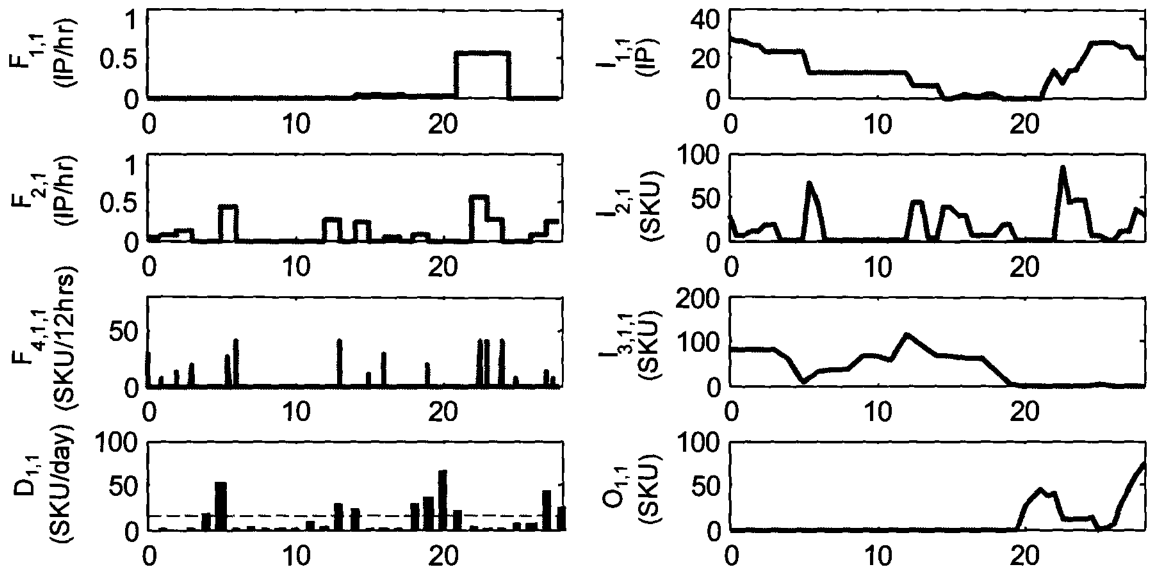
Figure 5.6 Simulation Results without model mismatch – Case A of IP and SKU because the robust MPC tries to avoid back orders not only for the nominal realization of the system, but also for other realizations within the confidence level. The additional inventories to prevent the potential back orders are "safety stock" for uncertainty, which is not required in this *ideal case*.

Finally, the manipulated variables $F_{4,1,1}$ and the $F_{2,1}$ (determined by the manipulated variable $T_{s,1}$) are fluctuating from day to day and the fluctuations under robust MPC are greater than under nominal MPC. Similar results can also be found in other case studies in this chapter. This may be because the robust MPC is more "sensitive" to the time-varying uncertainties. The fluctuations in the manipulated variables could be reduced in either MPC method by adding move suppression terms in the objective function or constraints on the change of the manipulated variables.

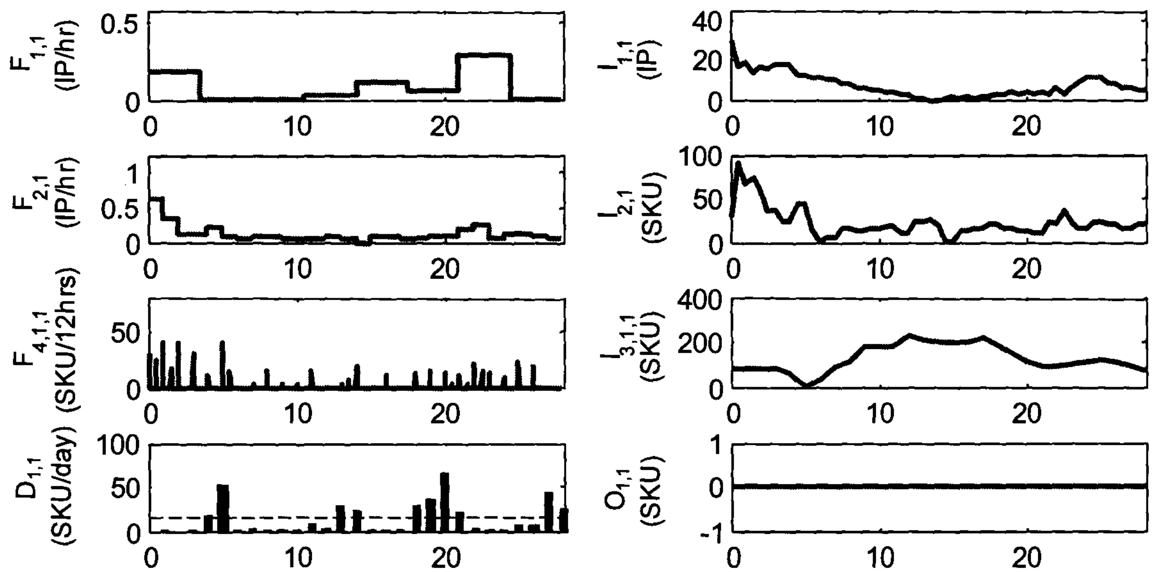
Figure 5.7 and Figure 5.8 present the simulation results in Cases B and C, respectively. In these mismatch cases, the nominal MPC performs unsatisfactory because of the large numbers of back orders occurring after the about 20th day. The nominal MPC reduces inventories and does not consider safety stock, which would enable it to respond well to mismatch. The robust MPC prevents back orders because it keeps a safety stock to satisfy deviations from average performance that are due to model and forecast mismatch. In both Cases B and C, the nominal MPC takes long time to eliminate the back orders, because the transportation time from the DC the RDC is long (5.5-6.5 days), which is the minimum time for the system to respond to the shortage SKU at the RDC.

5.5.1.2 Summary of the results of more situations

The figures in the last subsection represent the behavior of nominal and robust MPC for a specific mismatch realization in three typical situations. The simulation studies for a larger numbers of realizations and the variation combinations of mismatch were performed to better understand the advantages of robust MPC. These simulation results are summarized in Table 5-5. To generate the results for Table 5-5, simulation were performed with nominal and robust MPC for 12 different cases for 28 days, in which the parameters R_s, τ, D can have mismatch from the actual process. (Note the mismatch types in Case 1, 11, 12 are the same to those in Case A, B, C discussed in the previous subsection.) The results in the table report the average behavior for 100



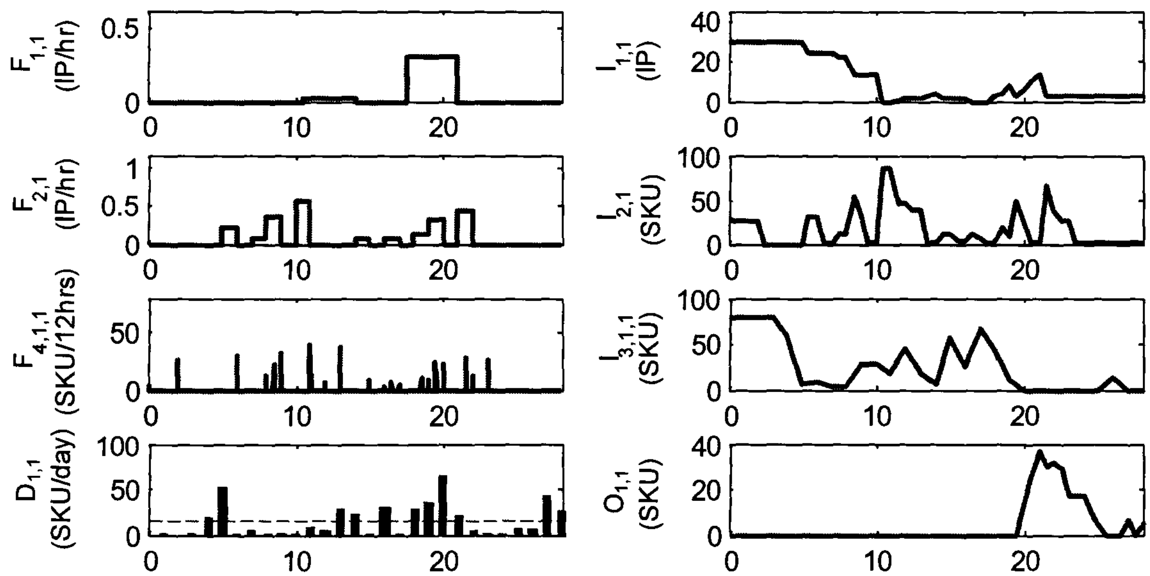
(a) Results with nominal MPC



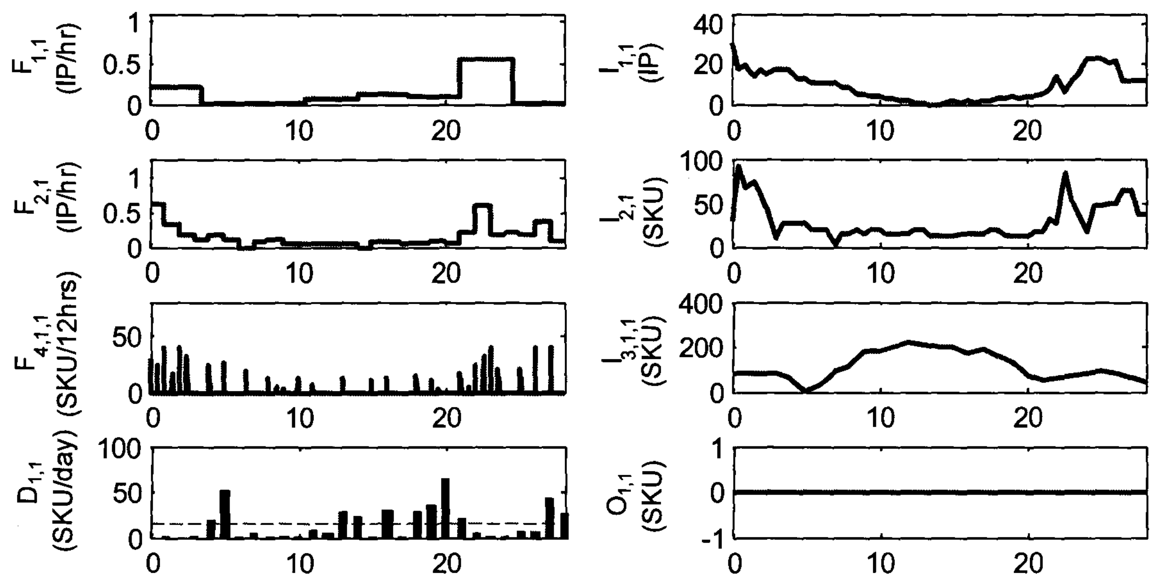
(b) Results with robust MPC

$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

Figure 5.7 Simulation results with model mismatch – Case B



(a) Results with nominal MPC



(b) Results with robust MPC

$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

Figure 5.8 Simulation Results with model mismatch – Case C realizations of the uncertain parameters for each of the 12 cases. The uncertainty realizations were selected randomly from their distributions and therefore, were allowed to exceed the ranges used when designing the optimizer.

The results in Table 5-5 clearly demonstrate the reduced quantity of back orders when the supply chain system is optimized by the robust MPC. Some small backorders occurred with robust MPC because of (a few) realizations with demand outside the design confidence region and the approximation made for the modeling. In contrast, the average back order was nonzero in all the 11 study cases when nominal MPC was used, and the increase in backorders was substantial in most cases. It is also clear that the inventories maintained by the robust MPC are larger than those by the nominal MPC. The extra inventories maintained by the robust MPC provide the safety stocks that significantly reduce back orders.

Table 5-5 also shows that the uncertainty in any single source has effect on the system performance. Uncertainty in the customer demand D has the dominant effect, and this uncertainty must be addressed in the MPC calculation to reduce the back orders. While the effects of the uncertainties in the SKU transportation time τ are less significant, addressing it in the robust MPC calculation is still essential to reduce the back orders. The uncertainty in the manufacturing rate R_s has the least effect on the system performance in Table 5-5, because the manufacturing capacity is large enough in this case study to enable the daily manufacturing quantity to be fulfilled as the uncertain rate varied, that is, sufficient spare capacity existed in most of the days. Finally, the more sources of uncertainty are present, the more back orders are incurred.

From these studies, we conclude that we need to address all sources of uncertainty simultaneously in the robust MPC to reduce the back orders. Also, since R_s, τ appear in the feedback model and D is the disturbance to the system, both the feedback model mismatch and the disturbance uncertainty must be addressed to achieve the robust MPC performance in Table 5-5. Many existing robust MPC methods are only able to address part of the sources of the uncertainty in Table 5-5, e.g., feedback model mismatch (Kothare et al., 1996) or disturbance uncertainty (Goulart et al., 20006), so they cannot achieve the same performance as the method developed in this thesis.

Table 5-5 The simulation results with nominal and robust MPC in the 12 cases

Case Number	R_s	τ	D	Nominal MPC Ave[5](ΣI[6]) (SKU)	Nominal MPC Ave(ΣO[7]) (SKU)	Robust MPC Ave(ΣI) (SKU)	Robust MPC Ave(ΣO) (SKU)
1	C[1]	C	C	6233	4[8]	11733	0
2	C	C	U1[3]	7356	602	10187	43
3	C	C	U2[4]	6593	241	10540	30
4	C	U	C	7722	46	11531	0
5	C	U	U1	7296	794	10279	70
6	C	U	U2	6558	330	10647	56
7	U[2]	C	C	6282	5	11212	0
8	U	C	U1	7300	607	10284	43
9	U	C	U2	6571	248	10454	32
10	U	U	C	7502	49	11408	0
11	U	U	U1	7185	796	10359	74
12	U	U	U2	6560	338	10519	56

Note: [1] C = Certain; [2] U = Uncertain; [3] U1 for uncertain demands denotes the nominal demands are expected demands; [4] U2 for uncertain demands denotes the nominal demands are only incorrect for the 16th day; [5] The average of the quantity in the parentheses for the 100 samples; [6] The sum of the daily inventory level over the simulation horizon; [7] The sum of the daily back order level over the simulation horizon; [8] The small back orders are due to the approximation introduced in the modeling.

We emphasize that the robust MPC obtains the minimum safety stock through optimization according to the known uncertainty characterization. Therefore, the safety stock is the minimum for the closed-loop uncertainty experienced, which is better than setting constant safety stocks based on past experience.

5.5.1.3 Trade-off between back orders and safety stock

Table 5-5 reports small amounts of back order under the robust MPC, which are caused by the realizations outside the confidence region addressed by the robust MPC and modeling approximations. We could increase the confidence level of satisfying uncertain constraints to reduce the potential back orders at the cost of maintaining greater "safety stock". The trade-off in the tuning of the confidence level for robust MPC is shown in Table 5-6. In this table, the results with the robust MPC using different confidence levels are from the simulation of Case 11 in Table 5-5 (where the nominal demands are the expected demands obtained from historical data). The quantities shown in the table are from 100 realizations of the system.

We can see from Table 5-6 that when the confidence level is increased, the average total inventory during the 28 days increases while the average total back orders and the worst case back orders decrease. A new quantity, service level, is also shown in Table 5-6. The service level of a period is defined to be 1 minus the ratio of the total back orders to the total demands during the period (Tersine, 1994). It indicates how well the customers are serviced during the period. It is clear that the higher the confidence level, the better the system can service the customer but the more inventory the system needs to maintain.

If the back order cost (penalty) correctly reflects the importance of service to the customers, we can determine the best tuning of the confidence level quantitatively by comparing the total costs of the inventories and back orders. According to the inventory and back order unit costs shown in Table 5-2 and the inventories and back orders shown in Table 5-6, we can calculate the costs for the 5 different tuning of the confidence levels.

The calculated results are illustrated in Figure 5.9. We can see that the inventory cost increases and the back order cost decreases with the increase of the confidence level, and the lowest total cost is present with the confidence of 90%. We note the steep

Table 5-6 The simulation results of robust MPC with different confidence levels

Confidence level	Ave[1](ΣI[2])	Ave(ΣO[3])	Max[4](ΣO)	Average service level
99%	22819	35	261	93.1%
95%	13244	37	271	92.9%
90%	12439	42	362	92.1%
85%	11319	58	418	89.2%
80%	10416	72	670	86.9%

Note: [1] The average of the quantity in the parentheses for the 100 samples; [2] The sum of the daily inventory over the simulation horizon; [3] The sum of the daily back order over the simulation horizon; [4] The maximum of the quantity in the parentheses for the 100 samples.

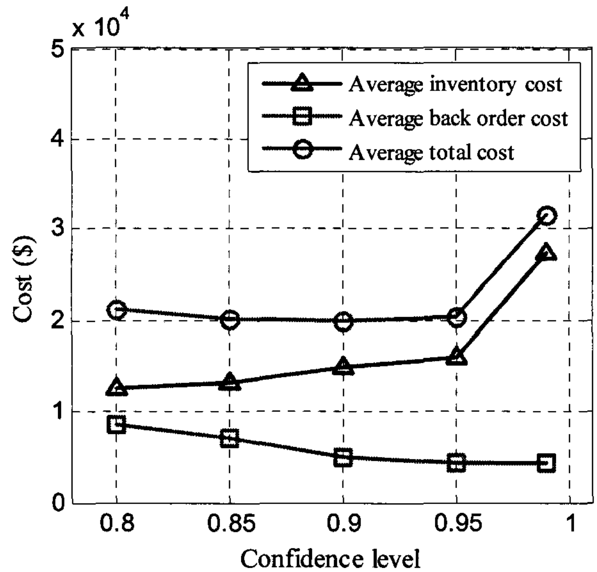


Figure 5.9 Relationship between system cost and the confidence level of robust MPC

increase in total cost when the confidence level is above 95%, which indicates that achieving service levels above 93% will be very expensive for this system because of model uncertainty and demand forecast errors as well as the long transportation times which delay feedback corrections.

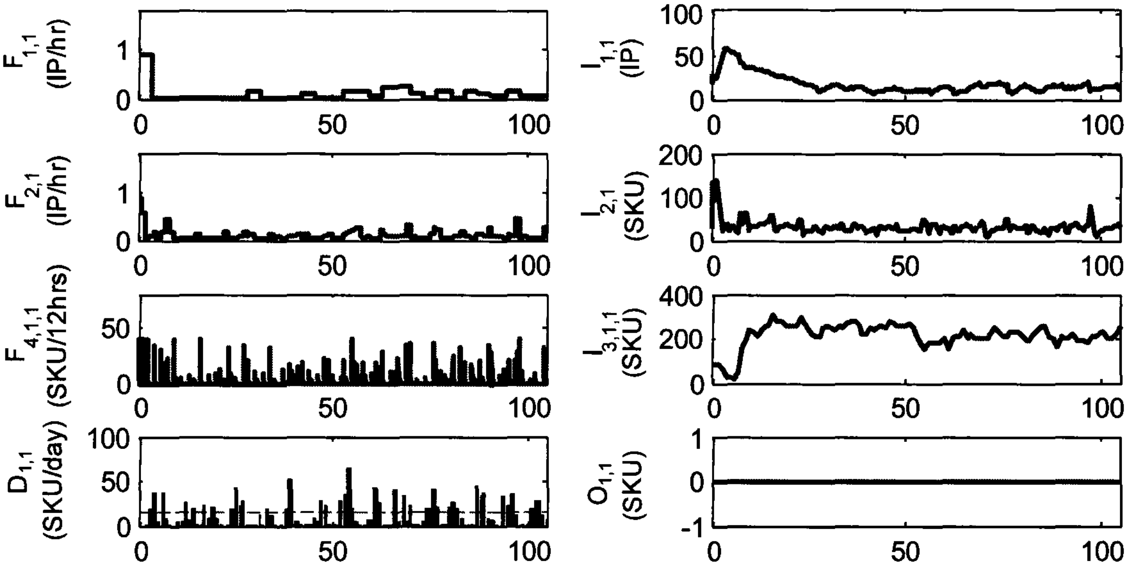
5.5.1.4 The Tuning of the confidence level

A confidence level properly tuned off-line may still not as good as desired in practice, because the current uncertainty of the system may be different from the uncertainty characterized according to historical data. Reasons could be changes in traffic, modifications to process equipment (planned or wear), or changes in customer purchasing patterns. This problem can be addressed by tuning the confidence level automatically online based on real time data. Here, we are not trying to propose a systematic adaptive tuning method, but to demonstrate the advantage of the idea of adaptive tuning through a simple heuristic and a case study.

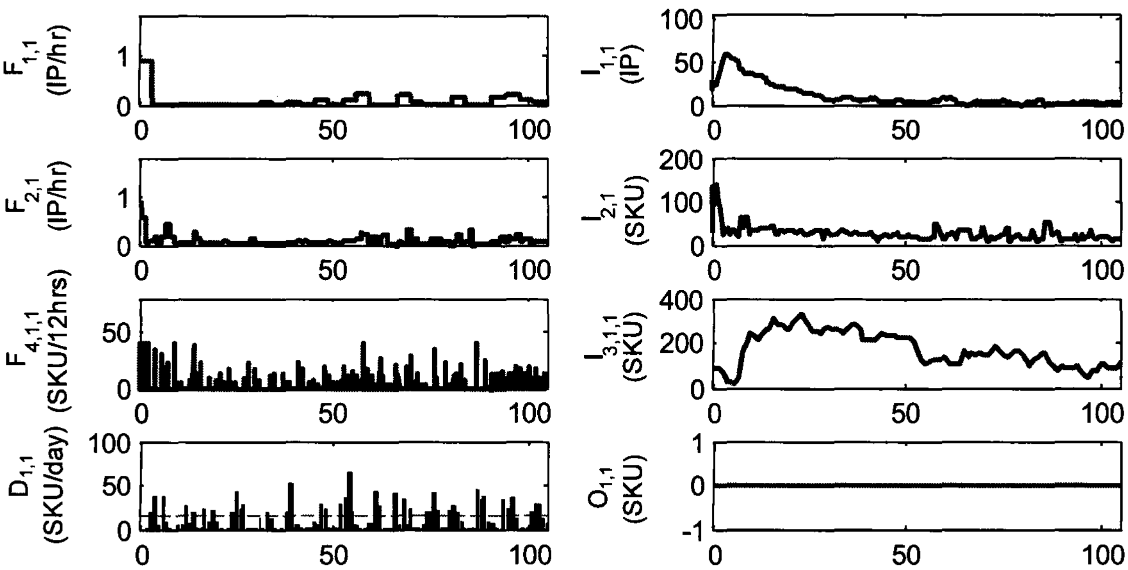
The heuristic adaptive tuning method used here is:

1) When back orders occur, increase the confidence level of the robust MPC, α, by 1%. This is to prevent α to be overly small.
2) When the SKU inventory at the RDC (I_3) is above a particular level for a week, decrease α by 1%. The particular inventory level is an indicator to judge if the safety stock is too large, which can be obtained by the experienced personnel or an inventory management heuristic.

Figure 5.10 compares the robust MPC methods with and without adaptive tuning by the simulation of the closed-loop system for 15 weeks. Here, the nominal demands are the expected demands obtained from historical data (i.e. as in the Case 11 in Table 5-5). In the controller, the nominal predictions of the manufacturing rate and transportation time have mismatch and the forecast of the demand is the average demand. The initial confidence level of the robust MPC is 99%. We set the 90 SKUs to be the threshold inventory level to judge if the SKU inventory at the RDC is overly large. We can find that if keeping the 99% confidence level without using the adaptive tuning, the robust MPC will maintain an excessive SKU inventory ($I_{3,1}$) of over 200 SKUs at the RDC for



(a) Simulation results under robust MPC without adaptive tuning



(b) Simulation results under robust MPC with adaptive tuning

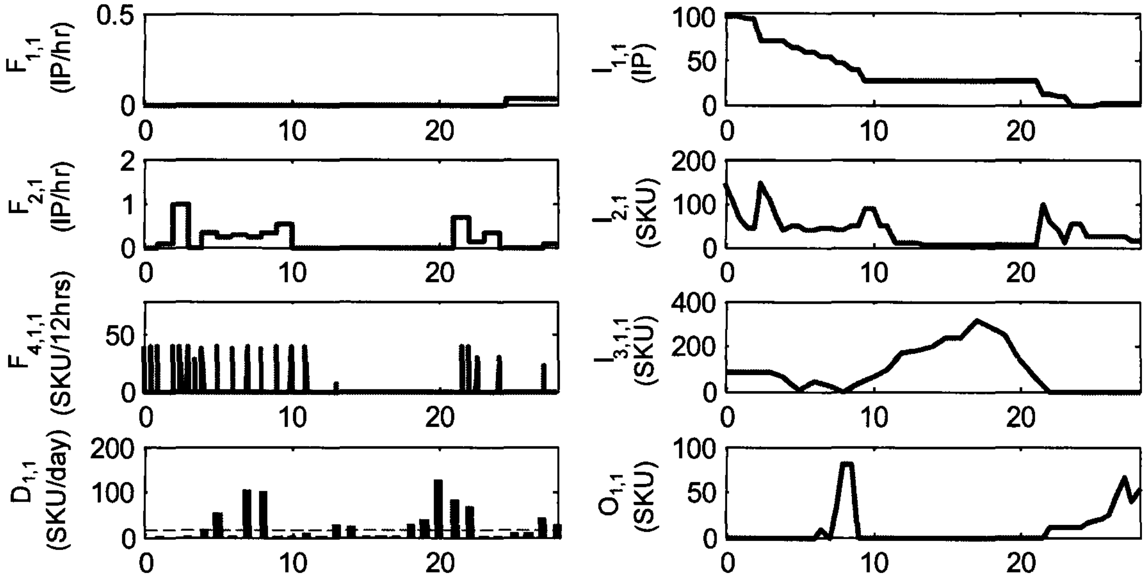
$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

Figure 5.10 Comparison of the robust MPC with and without adaptive tuning the quarter of the year. With adaptive tuning the robust MPC will gradually decrease the confidence level so that $I_{3,1}$ decreased to about 100 SKUs. This study demonstrates the potential for improving supply chain optimization through adaptive updating of uncertainties, and we conclude that this is an opportunity for future investigation.

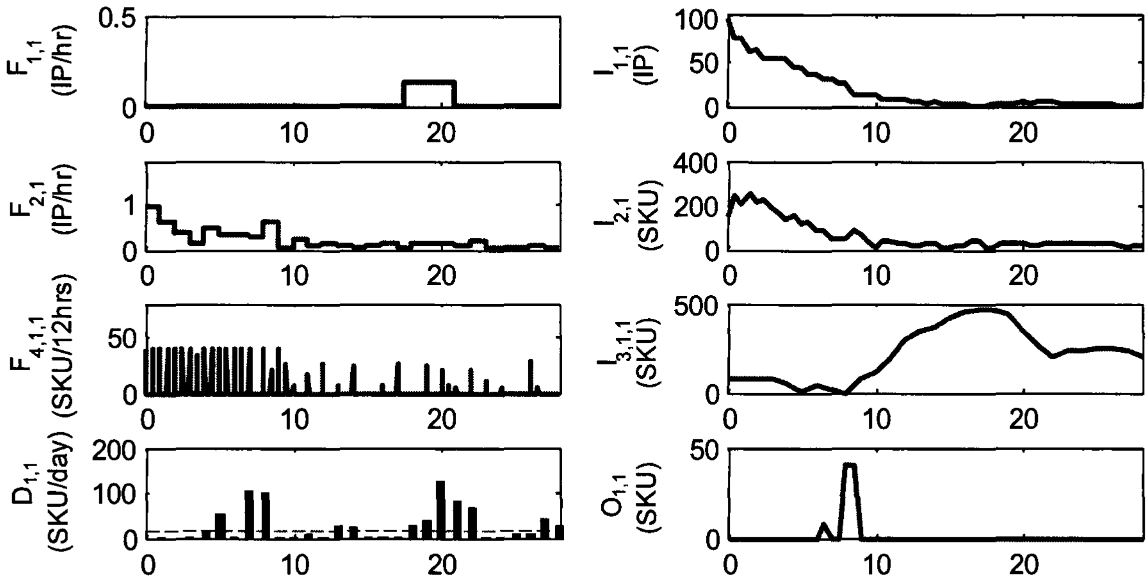
5.5.1.5 Back orders due to limited transportation capacity

Previous cases have demonstrated that we can decrease back orders occurring with robust MPC by increasing the confidence level of the chance constraints. Here, we note that it is not always possible to achieve zero back orders by increasing the confidence level to almost 100%. In some situations, the occurrence of back orders is not due to the tuning of robust MPC, but to the limitations in the behaviour of the real system.

Figure 5.11 shows a situation where large amount of back orders occur both under nominal MPC and under robust MPC. In the controllers, the nominal predictions of the manufacturing rate and transportation time have mismatch and the forecast of the demand is the average demand (as the Case 11 in Table 5-5), but the simulated plant experiences the actual demands taken from some successive 28 days in the year of 2004 (which is within the 90% uncertain demand region used in the controller design) In this case scenario, there are large demands (100 SKUs/day) at the RDC on the 7th and 8th days of the simulation period. The transportation time is 6 days to ship SKUs from DC to the RDC, and this lead time (in supply chain terminology) or dead time (in automatic control terminology) presents a limit to the responsiveness of the feedback system. In this case study, the initial SKU inventory at the RDC roughly equals the demands in the first 6 days, so no infeasibility occurs due to insufficient initial inventory. From the system dynamics, the SKUs shipped in the first two days from the DC to the RDC must be able to satisfy the demands on the 7th and 8th days (minus any residual inventory in the RDC) to ensure that no back orders occur. However, the transportation capacity for two days is limited to a maximum of 160 SKUs (40 SKU/shipment and 1 shipment/12 hours). Because of short-term high demands at the RDC, the existing transportation capacity is not large enough to satisfy all the demands in the 7th and 8th days (totally 200 SKUs). We can see in Figure 5.11 (b) that the robust MPC addresses the potential uncertainty and



(a) Results with nominal MPC



(b) Results with robust MPC

$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

Figure 5.11 Simulation Results with incorrect nominal prediction of parameters - unavoidable back orders due to the limit of transportation capacity orders the system to ship the SKUs to the RDC at the full capacity in the first several days. However, because of capacity limits, back orders still occur during the 7^{th} and 8^{th} days. As the demand decreases to within the capacity of the transportation system, the back orders are quickly eliminated after the 8^{th} day by the SKUs transported on the 3^{rd} and subsequent days.

We can also find from Figure 5.11 (a) that more back orders occur under nominal MPC on the 7^{th} and 8^{th} days as well as on the 21^{st} and subsequent days. Therefore, the robust MPC still outperforms the nominal MPC in this situation.

5.5.2 Case study with 2 IP/SKU types and 2 RDCs

This section shows the case study results when considering two material/product types (the 1^{st} and 2^{nd} IP/SKU) and two regional distribution centers (the 1^{st} and 2^{nd} RDC). Table 5-7, Table 5-8 and Table 5-9 show the supply chain system parameters, the different periods used in the nominal and robust MPC and the additional robust MPC parameters used in this section respectively. Also, we use the costs in Table 5-2 for this section. The uncertainty in the customer demands of both SKU types at both RDCs can be characterized from the historical demand data, and we show the histograms of the demand data in Appendix H. The nominal forecast of the demand is the expected demand obtained from historical data. The nominal forecast values of the manufacturing rate and transportation time do not match the actual process behavior. The simulations were run for 28 days.

Figures 5.12 and 5.13 show the simulation results under nominal and robust MPC, respectively. We can find that if the nominal MPC is employed, back orders of the 1^{st} SKU will occur at both RDCs (especially the 2^{nd} RDC) due to the model mismatch; but no back orders of the 2^{nd} SKU occur because the uncertainty in the demand forecast of the 2^{nd} SKU is much smaller than that of the 1^{st} SKU.

We can also find that if the robust MPC is employed, back orders do not occur for either SKUs at either RDCs because of the explicit handling of uncertainty in robust MPC. Also, the robust MPC maintains much larger inventories of the 1^{st} SKU at the RDCs (about 200 SKUs for each) than those of the 2^{nd} SKU at the RDCs (about 70 and

100 for each), because the nominal prediction and the uncertainty in the 1st SKU demands are much larger than those for the 2nd SKU.

Table 5-7 Parameters of the system with the 1st and 2nd IP/SKU and the 1st and 2nd RDC

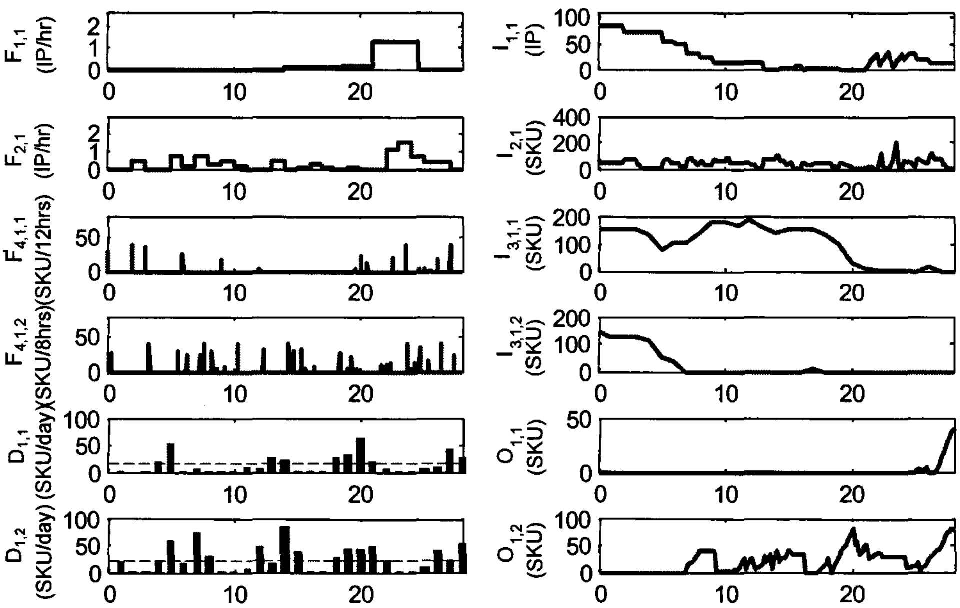
Parameter	Value
Nominal SKU manufacturing rate R_s (SKU/hour)	16.7 (the 1st IP/SKU) 16.7 (the 2nd IP/SKU)
R_s range with 90% confidence (SKU/ hour)	13.3-22.2 (the 1st IP/SKU) 13.3-22.2 (the 2nd IP/SKU)
Unit converting coefficients $C_{IP\text{-}SKU,i}$ (IP/ SKU)	6.0 (the 1st IP/SKU) 6.6 (the 2nd IP/SKU)
Nominal SKU transportation time τ_j (hour)	144 (the 1st RDC) 144 (the 2nd RDC)
τ_j range with 90% confidence (hour)	132-156 (the 1st RDC) 132-156 (the 2nd RDC)
SKU Shipping Intervals (hour)	12 (the 1st RDC) 8 (the 2nd RDC)
SKU transportation capacity $F_{4,\max,j}$ (SKI/shipping interval)	40 (the 1st IP/SKU) 40 (the 2nd IP/SKU)
Customer demands $D_{i,j}$ range with 90% confidence* (SKU/day)	0-38 (the 1st SKU 1, the 1st RDC) 0-55 (the 1st SKU 1, the 2nd RDC) 0-12 (the 2nd SKU 1, the 1st RDC) 0-16 (the 2nd SKU 1, the 2nd RDC)

Table 5-8 The different periods of the nominal and robust MPC controllers

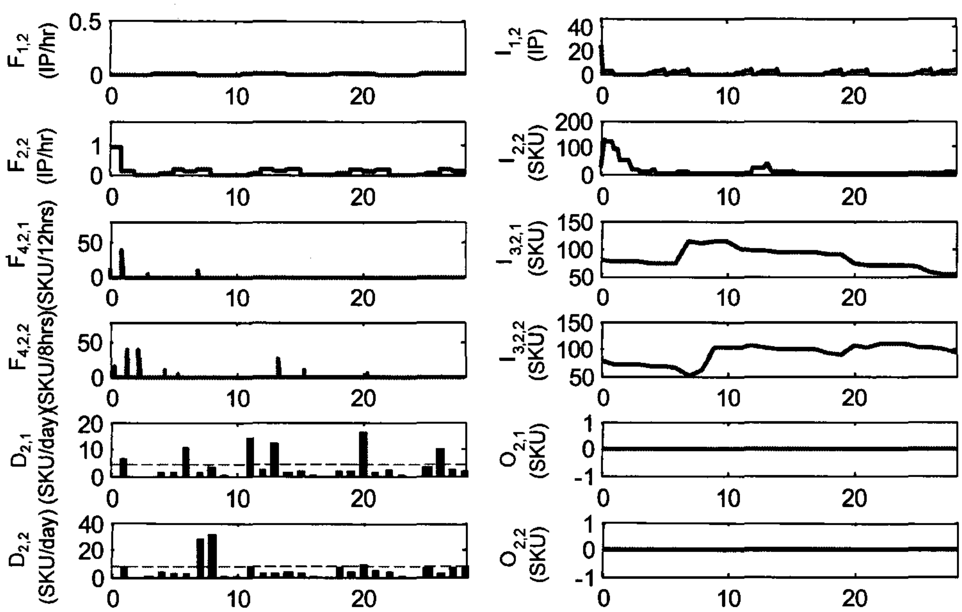
Parameter	Value
Measurement and MPC execution period, ΔT_c (day)	1
Discrete model time interval, ΔT (hour)	4
Control and prediction horizon, n (day)	14

Table 5-9 The additional parameters of the robust MPC controller

Parameter	Value
Confidence of each chance constraint, α	90%
Q in inner problem – element for IP Inventory (IP^{-2})	0.0144
Q in inner problem – element for SKU Inventory (SKU^{-2})	0.0004
R in inner problem – element for IP manufacturing (IP^{-2})	1
R in inner problem – element for SKU manufacturing (SKU^{-2})	0.01
R in inner problem – element for SKU Transportation (SKU^{-2})	0.0016



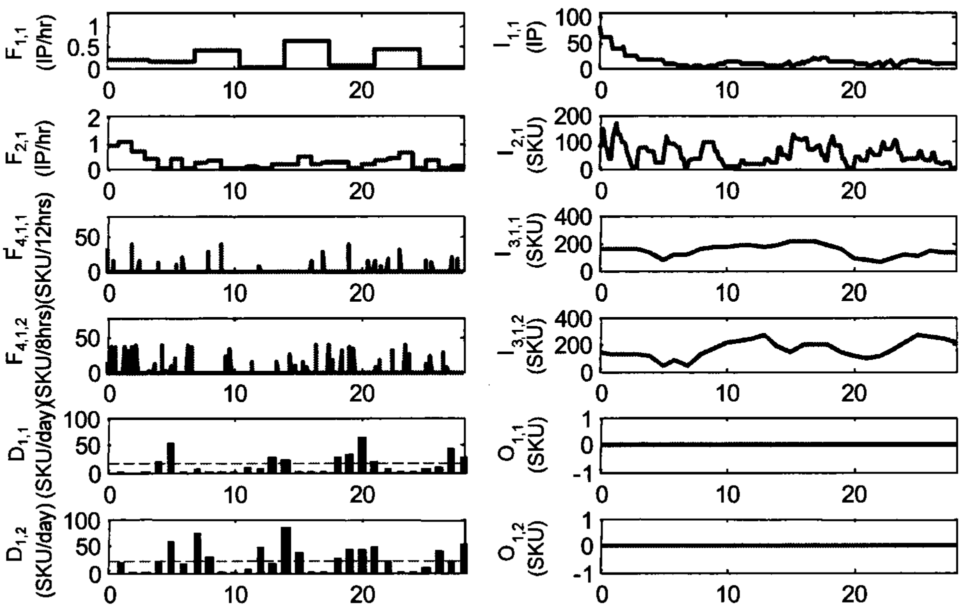
(a) Simulation results for the 1st IP/SKU



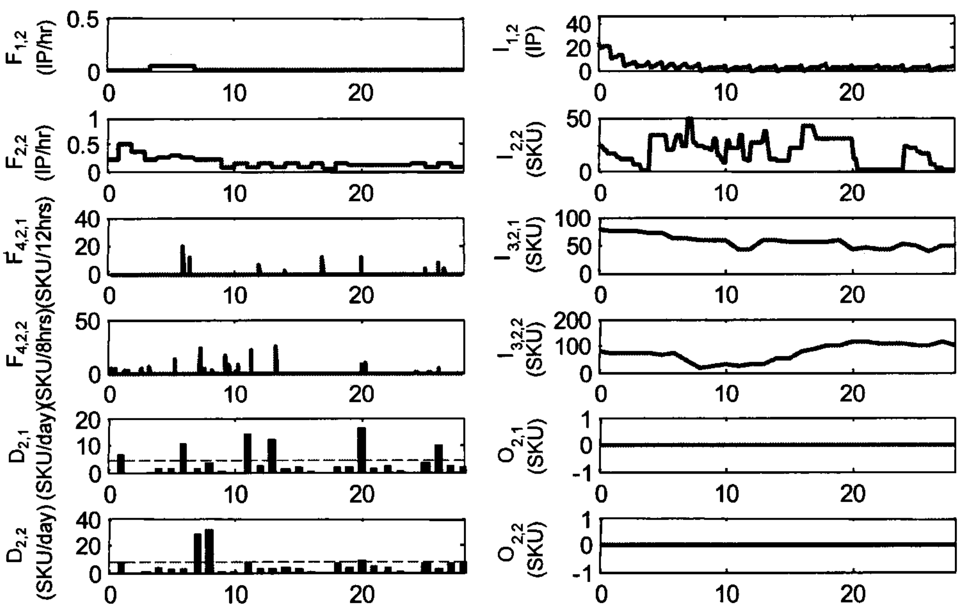
(b) Simulation results for the 2nd IP/SKU

$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

Figure 5.12 Simulation results of 2 IP/SKU types and 2 RDCs – Nominal MPC



(a) Simulation results for the 1st IP/SKU



(b) Simulation results for the 2nd IP/SKU

$F_{1,i}$: Flow of the i^{th} IP from IPM to IPS $F_{2,i}$: Flow of the i^{th} IP from IPS to SKUM $F_{4,i,j}$: Shipping of the i^{th} SKU to the j^{th} RDC
$I_{1,i}$: Inventory of the i^{th} IP at IPS $I_{2,i}$: Inventory of the i^{th} SKU at DC $I_{3,i,j}$: Inventory of the i^{th} SKU at the j^{th} RDC
$D_{i,j}$: Customer demand of the i^{th} SKU to the j^{th} RDC $O_{i,j}$: Back order of the i^{th} SKU at the j^{th} RDC

Figure 5.13 Simulation results of 2 IP/SKU types and 2 RDCs – Robust MPC

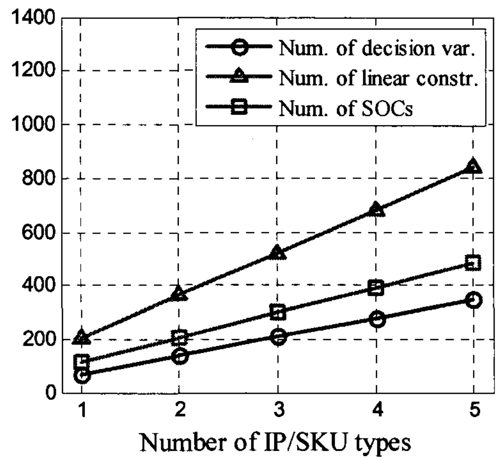
5.5.3 Computational complexity

In Section 5.5.2 we demonstrated that the robust MPC method works well for multiple IP/SKU types and multiple RDCs. This section discusses the computational complexity of the problem with respect to the number of IP/SKU types and RDCs. We define Case B described in Section 5.5.1.1 as the base case for the simulation, and compare the results with different numbers (denoted by $n_{IP/SKU}$) of IP/SKU types and only 1 RDC, or the results with different number of RDCs (denoted by n_{RDC}) and 1 IP/SKU type. Also, only one manufacturing plant (SKUM) and one plant SKU distribution center (DC) exist in all the simulations. We will discuss the both theoretical computational complexity and also give computational results from test problems.

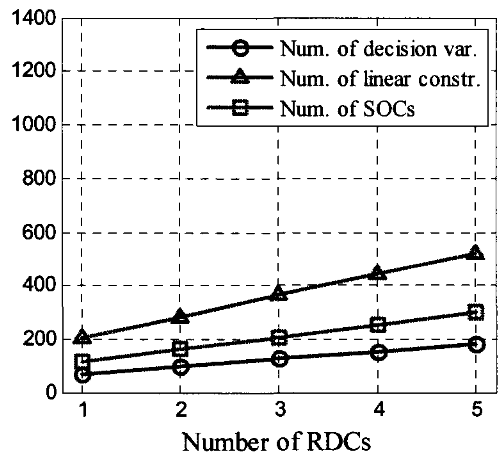
First, with the increase of n_{RDC} and $n_{IP/SKU}$, the scale of the optimization problem (formulation (5.37)) will increase. According to the modeling of the system and the robust MPC formulation shown in Sections 5.2 and 5.3, the number of the decision variables $n_{decision}$, the number of the linear constraints n_{LC} and the number of the second order cones N_{SOC} are all linear with respect to $n_{RDC}n_{IP/SKU}$, i.e. $n_{decision}, n_{LC}, N_{SOC} \sim O(n_{RDC}n_{IP/SKU})$. The following Figure 5.14 summarizes the simulation results that indicate the effects of n_{RDC} and $n_{IP/SKU}$ on $n_{decision}, n_{LC}, N_{SOC}$, which is in accordance with the theoretical analysis.

Second, the larger optimization problem requires the increased computer memory for real-time computation. The memory is basically used to store the matrices used in the calculation, so it is proportional to the elements in the matrices. According to formulation (5.37), the matrix elements are $O((n_{RDC}n_{IP/SKU})^2)$. Figure 5.15 shows the memory required in the online calculation, which is more linear than quadratic with respect to $n_{RDC}n_{IP/SKU}$. This is because many matrices are sparse, and the zero elements do not occupy any space when sparse matrix numerical methods are employed.

Finally, the larger optimization problem takes more time to solve. Note that we need to update the uncertainty characterization according to the saturation of the decision variables in the real-time before we solve the optimization problem. According to the discussion in Section 3.3.2 in Chapter 3 and in Appendix E, the uncertainty update

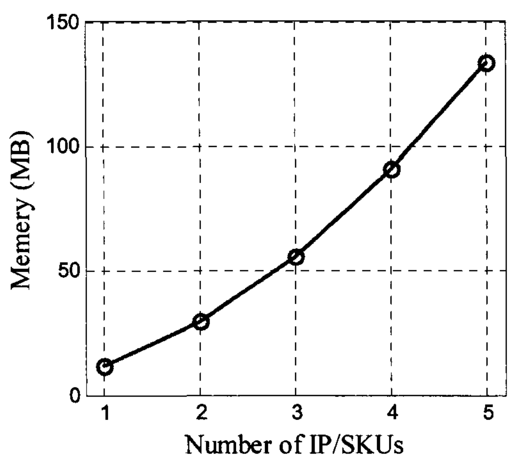


(a) Results with 1 RDC and different numbers of IP/SKU types

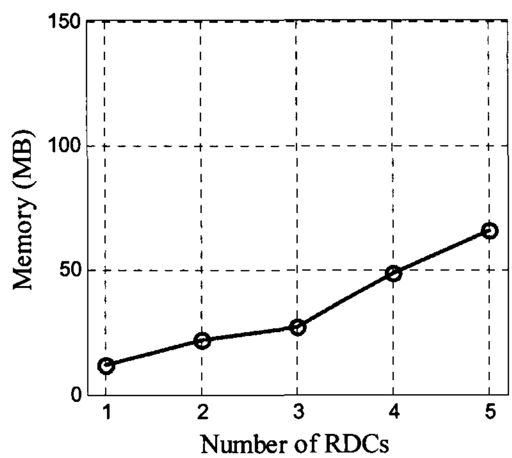


(b) Results with 1 IP/SKU types and different numbers of RDCs

Figure 5.14 Effects of numbers of IP/SKU types and RDCs on problem scale



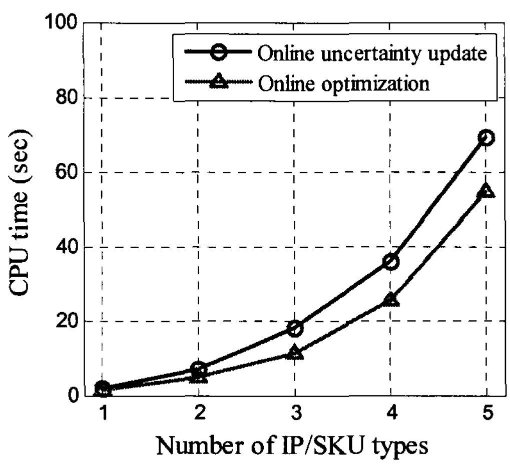
(a) Results with 1 RDC and different numbers of IP/SKU types



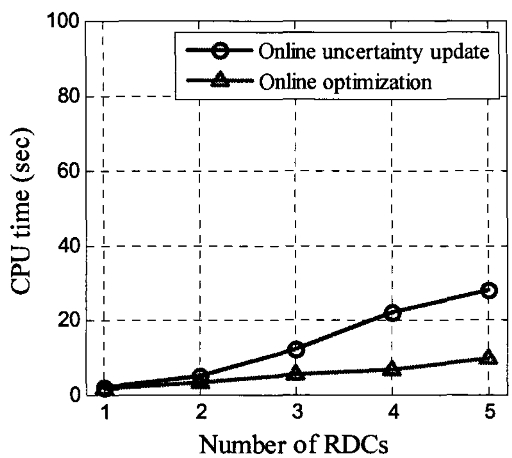
(b) Results with 1 IP/SKU types and different numbers of RDCs

Figure 5.15 Effects of numbers of IP/SKU types and RDCs on memory calculation is cubic with respect to the number of decision variables, i.e. $O(n_{decision}^3)$ and thus $O((n_{RDC}n_{IP/SKU})^3)$. According to Lobo et al. (1998), the time complexity of solving a SOCP problem is bounded from above by $O(N_{SOC}^{1/2}n_{decision}^2\sum_{l=1}^{N_{SOC}} n_{SOC,l})$, where $n_{SOC,l}$ denotes the size of each second order cone, which is proportional to n_{RDC}, $n_{IP/SKU}$ in this problem. So the time complexity of solving the SOCP problem (5.37) is bounded by $O(N_{SOC}^{1/2}n_{decision}^2 N_{SOC}n_{RDC}n_{IP/SKU})$, i.e., $O((n_{RDC}n_{IP/SKU})^{9/2})$.

Figure 5.16 shows the effects of n_{RDC} and $n_{IP/SKU}$ on the CPU time required for the real-time uncertainty update and the optimization. It's clear that both times are polynomial with respect to n_{RDC} and $n_{IP/SKU}$. Note that here we only show the CPU time of these two jobs, but the total time to complete all the calculations required for a controller execution is much more than those shown in Figure 5.16 for the current version of the software. The additional time is due to (a) a large amount of auxiliary operations and calculations to preprocess the data and build the optimization formulation in MATLAB; and (b) the slow procedure to exchange data between MATLAB and the CLPLEX optimizer in GAMS (through reading and writing data from and to the hard



(a) Results with 1 RDC and different numbers of IP/SKU types



(b) Results with 1 IP/SKU types and different numbers of RDCs

Figure 5.16 Effects of numbers of IP/SKU types and RDCs on computing time disk). This additional time can be reduced by changing the software structure to make the data preprocessing and organization more efficient and improving the data exchange between MATLAB and GAMS in computer memory.

5.6 Conclusions

This chapter discusses the application of robust MPC to supply chain optimization through a real industrial multi-echelon supply chain optimization problem. Since the structure of the supply chain system is representative of those in industry, the method developed for this system should also be applicable to many other real problems. The key restriction is the occurrence of only continuous variables for the supply chain, with any discrete decisions made at a lower level in the decision hierarchy. While certainly not completely general, other researchers have found similar formulations appropriate for industrial problems, e.g., Braun et al. (2003), Wang et al. (2007), etc.

In this chapter, we choose the greatest common divisor of the different decision implementation periods and feedback period as the sampling time period, so the supply chain system can be modeled in the form of a canonical discrete time state-space model. The different uncertainties of the system are modeled using uncertain parameters. The structural uncertainty caused by the uncertainty in the SKU transportation time is approximated by a novel disjunctive model formulation with parametric uncertainty.

We adapted the general robust MPC framework developed in Chapter 3 for the supply chain optimization. The resulting bilevel optimization problem is different from the one in Chapter 3, because the inner optimization problems are LPs instead of QPs. We approximate these LPs by QPs with the goal of achieving the targets set by the upper level optimization. With this modification we can apply the active set heuristic developed in Chapter 3 and transform the bilevel problem into single level problem. Also, the non-normally distributed uncertain customer demands are characterized with Monte Carlo sampling, so that it can be handled within the framework of chance-constrained program

The case study results show that the robust MPC can determine the optimal safety stock with the known information on uncertainties, which is a key advantage of the robust MPC over nominal MPC for supply chain optimization. The simulation study also shows the ability of the robust MPC to address both the model mismatch and disturbance uncertainty for this supply chain optimization problem, which is important to reduce the back orders. When hard bounds (that represents the limitation of the real system) are encountered, the robust MPC may not prevent back orders, but it can manage the system to reduce back orders when compared with a nominal MPC.

We also discuss the trade-off between the inventory and back order (or service) levels through tuning the confidence level of robust MPC through and evaluate the effects through simulation studies. The trade-off can be evaluated by comparing the inventory and back order costs with different tunings. The importance of adaptive tuning is discussed, and its advantage is demonstrated in a case study using a prototype adaptive method.

The robust MPC is successfully applied to the system with 1 IP/SKU type and 1 RDC and the system with 2 IP/SKU types and 2 RDCs. We point out that the theoretical computational complexity of the problem is polynomial with respect to the number of IP/SKU types and RDCs, which is validated by simulations of different test problem. This means we can apply this method to larger systems with more IP/SKU types an RDCs, and while computations increase with problem size, the rate of increase is moderate and the robust MPC does not suffer from the curse of dimensionality.

Chapter 6

Summary and Future Work

6.1 Summary

This thesis develops a novel robust MPC method for the control and optimization of dynamic, uncertain systems with feedback, such as process control systems or supply chain systems. The method is designed to optimize an uncertain closed-loop system behavior, not to robustly stabilize it, although the method could be extended to provide robust stability, as discussed later. It offers a general framework that can address different sources of parametric uncertainty with efficient and reliable solution for real-time implementation, and this framework can be tailored for the application to different types of problems.

Chapter 3 develops the general framework of the new robust MPC method based on the conventional nominal MPC formulation with a state-space model. Because the controller influences the prediction of future behavior for uncertain systems, the robust MPC formulation is initially a bilevel stochastic optimization with the inner optimization approximating the future controller behavior. With an industry-proven heuristic and the chance-constrained programming technique, this difficult-to-solve problem is solved (approximately) by solving a limited number of deterministic, convex SOCP problems, which can be solved efficiently and reliably with an optimizer using an interior point method. An enhanced dynamic model with deviation variables is developed to reduce the conservativeness in the prediction of time-invariant uncertainty. An efficient closed-loop uncertainty characterization method is developed so that the extensive calculation can be performed off-line and the on-line calculation is efficient. The uncertainty in the state estimation, if not all the states are measurable, is integrated explicitly in the general framework of the new robust MPC. The case studies of several CSTR control problems demonstrate the new robust MPC method outperforms nominal MPC and outperforms simpler robust MPC formulations that do not include feedback uncertainty descriptions or state estimation errors.

Chapter 4 extends the general robust MPC framework developed in Chapter 3 to include two key features required for process control applications. The first feature is the robust steady-state optimization, which obtains feasible and economically optimal set points while addressing the closed-loop uncertainty, for the trajectory optimization (control) at each control execution period. Deviation variables are again used to enhance the steady-state model of the system for better prediction of time-invariant uncertainty. The steady-state method is originally formulated as a bilevel stochastic optimization problem, which is then approximated by a limited number of deterministic SOCP problems using the similar approach introduced in Chapter 3. The second feature is a quadratic and convex objective function formulation that can include expected performance and variances of the controlled variables in the prediction horizon, so that the robust MPC method can account for different input-output uncertainties when optimizing the (expected) performance. The advantages of the two extensions on handling constraints and achieving robust performance with the presence of uncertainty are demonstrated through case studies of several distillation and CSTR control systems.

Chapter 5 tailors the general robust MPC framework developed in Chapter 3 for a typical industrial supply chain optimization problem. At the beginning, a nominal linear state-space model is developed for the supply chain system with appropriate assumptions, and the uncertainties in the system are modeled as uncertain parameters in the linear model. Next, a bilevel *linear* stochastic optimization formulation is built to optimize the uncertain closed-loop dynamics of the supply chain system. The inner LP problems of the formulation are then approximated by QP problems so that the formulation can be approximated by a limited number of deterministic SOCP problems using the similar approach introduced in Chapter 3 and a tailored chance-constrained programming technique. The case study results show the advantage of the robust MPC over nominal MPC on reducing the back orders as well as the trade-off between the controller tuning and the customer service level. Finally, the theoretical polynomial computational complexity of the robust MPC method is validated by simulation studies, which show that the computations increase with problem size moderately and this robust MPC supply chain method does not suffer from the curse of dimensionality. Note that if we model the supply chain optimization problem in Chapter 5 using multi-stage stochastic programming formulation with recourse, a 14- stage problem must be solved that will be computationally intractable because the scale of the problem is exponential in the number of scenarios.

6.2 Summary of Contributions

The key contributions of this thesis are summarized in the following.

- A general formulation of a new linear robust MPC method that optimizes the uncertain closed-loop system behavior in the prediction horizon and is subject to hard bounds on manipulated variables and soft bounds on controlled variables. The formulation explicitly addresses correlated, time-varying or time-invariant, parametric uncertainty of the plant/model mismatch, measured disturbance plant/model mismatch, unmeasured disturbances and noises. Although existing robust MPC methods can address one or more of these sources of uncertainty, none of them can address all these sources of uncertainty in a unified framework simultaneously with efficient solution.
- Efficient real-time solution of only a few convex SOCP problems, with intensive calculations for uncertainty description performed offline.
- Uncertainty in state estimation is addressed explicitly in the general robust MPC framework (if not all the system states are measurable).

- The first robust steady-state optimization method that explicitly addresses closed-loop uncertainty. This method has efficient and reliable solution for real-time applications.
- The first explanation and application of the deviation variable formulation for robust MPC applied to time-invariant systems. This formulation provides a tight bound on the uncertainty of the future transient behavior.
- A flexible convex and quadratic objective function is formulated to include nominal or expected dynamic performance as well as the output variances of an uncertain system. The inclusion of the objective function does not affect the computational tractability of the method.
- A tailored robust MPC formulation, with modified inner optimization problems and chance-constrained approach, for the operational optimization of a typical industrial supply chain system with uncertainties in manufacturing, transportation time and customer demand. Theoretical analysis and computational studies demonstrate that the computing time and storage space for solution increase moderately with system scale (i.e. number of SKUs and number regional distribution centers) and the method does not suffer from the curse of dimensionality.

6.3 Future Work

Optimization of uncertain systems with feedback is a broad topic with a large number of research opportunities. This section briefly discusses some specific topics for the research to address issues identified but not addressed in the process of completing this thesis.

6.3.1 Addressing integer variables

Integer variables may be required in the model for a process or supply chain system. For example, a process system may contain some parts described by logic, such as on-off decisions associated with equipment start up or shutdown or selection of only one from many manipulated variables. The logic is usually modelled with binary variables in the mathematical formulation of the optimization problem. In a supply chain system, integer variables may also come from the discrete nature of some quantities, such as number of trucks in a distribution network, the number of product packages to be shipped, and so forth. So, these systems must be modelled with both continuous and integer variables, and they are usually called hybrid systems. The application of nominal MPC to hybrid systems is attracting more and more attention for process systems (e.g. Bemporad and Morari, 1999a) and has always been recognized as important for supply chain systems (e.g., Mestan, et al. 2006).

Addressing integer variables in the new robust MPC framework will be very challenging because the inner optimization problems in the original bilevel stochastic optimization formulation are Mixed Integer Quadratic Programming (MIQP) problems instead of QP problems. Therefore, they cannot be transformed equivalently into an optimization formulation as a set of algebraic equations (as QP problems were transformed using their first order KKT conditions). Other approximating approaches or heuristics are needed to transform the bilevel formulation into single level formulation. Alternatively, one could use simpler control laws (e.g. PID control) to approximate the future control actions in the closed-loop prediction, so that the bilevel formulation can be avoided at the beginning; however, this would introduce complementary constraints for saturation effects.

Another challenge brought by integer variables is that the resulting robust MPC formulation is a Mixed Integer Programming (MIP) problem, which does not have polynomial solution times. So, the robust MPC method may not be able to be solved in real-time for a large (or even medium) scale problem. The future research should take advantage of the advances in the MIP research. For more details in MIP solution methods, readers can refer to Biegler and Grossmann (2004), Grossmann and Biegler (2004), Floudas (1995) and for robust MPC for hybrid system, e.g. Mhaskar et al. (2008).

6.3.2 Addressing nonlinear prediction model

As stated in Chapter 1, the robust MPC using linear models can deal with linear systems or nonlinear systems that can be well approximated by linear models with uncertain parameters (i.e., the nonlinearity of the system around the operating point or in the operating region is not substantial). In the engineering applications, however, some systems are highly nonlinear, and the operation of such a system may not be around a specific operating point (e.g. a batch process system). In this case, an explicit use of nonlinear model for the prediction of future dynamic behavior is needed.

The challenges of addressing nonlinear model in the robust MPC are similar to that of addressing integer variables. First, the inner optimization problems of the original bilevel stochastic optimization problem cannot be transformed into their first order KKT conditions. More complete KKT conditions (including the second order sufficient conditions) are needed for a (possibly approximating) transformation (Nocedal and Wright, 1999), which will make the formulation much more complicated. A realistic way to avoid the complicated KKT conditions is to use alternative approximating control laws to model the future controller. Second, the inclusion of nonlinear model may make the robust MPC formulation nonconvex. Although there are successful applications of local optimization method to nonconvex optimization in nonlinear MPC (e.g. Zavala and Biegler, 2009; Zavala et al., 2008), a global optimization (Floudas, 1999) method, which is not polynomial in time, is required to guarantee satisfactory control performance. Third, the use of nonlinear model makes the linear state estimation method, e.g. Kalman Filter, invalid, and requiring a more complicated estimation method, such as Extend Kalman Filter (EKF) (Kwakernaak and Sivan, 1972) or moving horizon estimation (Rawlings and Bakshi, 2006).

The future research has to take advantage of the advances in nonlinear MPC and nonlinear optimization. More details on these topics can be found in Allgower and Zheng (2000) and Biegler and Grossmann (2004).

6.3.3 Incorporating robust stability

The new robust MPC method proposed in this thesis does not guarantee the stability of the closed-loop system with presence of uncertainty, but it could be extended to include additional constraints to guarantee robust stability.

Let's briefly review how stability is guaranteed for nominal MPC first. The basis of achieving nominal stability for MPC is the Lyapunov stability theory (Haddad and

Chellaboina, 2008). According to the author's knowledge, all the existing MPC methods ensure asymptotical stability by forcing a Lyapunov function to monotonically decrease during the transient. For infinite horizon MPC, the dynamic system performance over the horizon (i.e., the squared difference between the controlled and manipulated variables and their set points) is a natural choice of the Lyapunov function (Keerthi and Gilbert, 1988). For finite horizon MPC, additional features need to be added to guarantee stability, such as zero stability constraints (Keerthi and Gilbert, 1988; Mayne and Michalska, 1990), a local stabilizing controller (with the resulting method dual-mode MPC) (Michalska and Mayne, 1993), or adding a terminal cost in the objective function with or without additional terminal constraints (e.g., Scokaert and Rawlings, 1998; Alamir and Bornard, 1995). Some other approaches force a Lyapunov function other than the dynamic performance to decrease in the constraints (e.g., Bemporad, 1998a; Yang and Polak, 1993).

The above idea for nominal stability may be extended to ensure robust stability, i.e., forcing a particular Lyapunov function to decrease during the transient not only for the nominal uncertainty realization, but also for all the other uncertainty realizations of concern. This strategy has been applied in some existing robust MPC methods. For example, Kothare et al. (1996) presented a robust MPC formulation for the system with polytopic or ellipsoidal uncertainty in plant/model mismatch, where the upper bound of a Lyapunov function is guaranteed to decrease with the presence of such uncertainty provided the initial solution is feasible. The real-time optimization problem of this method can be formulated into a (deterministic) Linear Matrix Inequalities (LMI) or called Semi-Definite Programming (SDP) (Ben-Tal and Nemirovski, 2002; Boyd and Vandenberge, 2004) problem. LMI or SDP problems are still convex problems that can be solved efficiently and reliably by inter point method, but they are typically more difficult than SOCP problems (Lobo et al, 1998). We may be able to incorporate this idea into the new robust MPC framework developed in this thesis to achieve robust stability at a particular significance level and for specific uncertainty sources and ranges.

6.3.4 Handling the change in the uncertainty

Since robust MPC addresses the uncertainty explicitly in it calculation, it is important that the uncertainty representation used by robust MPC is in accordance with the uncertainty in the real system. If the uncertainty used by robust MPC is larger than the real uncertainty, robust MPC will give overly conservative control which may lead to poor dynamic performance; if it is smaller, robust MPC will give overly aggressive control which may lead to constraint violations or even instability of the system.

In this thesis, the system uncertainty used in the robust MPC calculation is obtained off-line through historical data. During the real-time applications, the actual (current) uncertainty may be different from the uncertainty characterized off-line, e.g., it may have different mean, variance or even distribution pattern, and the robust MPC must be able to address this change to prevent undesired control performance.

One way to address the uncertainty change is to develop a heuristic to tune the robust MPC parameters (such as weighting matrices, confidence level, etc.) in the real time according to the measured information from a past time horizon. This idea has been demonstrated by a case study in Section 5.5.1.4 of Chapter 5 where a simple heuristic is applied to tune the confidence level of the robust MPC.

The second way is to identify the uncertainty according to recent measured data of the system variables in real-time, so that we can address the change in the uncertainty in the robust MPC formulation explicitly. Although it is different from the traditional system identification that identifies the value of a parameter instead of its uncertainty, the research in this approach should take advantage of the state-of-the-art system identification (e.g. Verhaegen and Verdult, 2007), and time series analysis (Box et al., 2008) methods. To avoid the need of interrupting the process in the real time, the method should be able to identify the uncertainty according to the closed-loop data without additional exciting signals, so the future research should also take advantage of the research work in the closed-loop identification (e.g., Box and MacGregor, 1976; Zhu, 1998; Esmaili et al., 2000). Note that some parameters in the system may be measured directly in the real time, such as production rate or transportation time in a supply chain system, and this may make the characterization of uncertainty easier.

6.3.5 Very Large-scale Systems

Although the solution time for the new robust MPC method is polynomial with respect to the size of the system, it can be prohibitively large for very large-scale systems (e.g. a supply chain system with hundreds of SKUs and stores). In this case, additional numerical techniques need to be incorporated into the robust MPC method to speed up the solution procedure.

First, a large-scale system is usually "sparse" in the sense that each of most inputs to the system only affects a few outputs of the system and most outputs of the system are only affected by a few inputs. The optimization of such system involves calculations with sparse matrices, and the efficiency of these calculations can be dramatically improved with the sparse matrix techniques (Zlatev, 1991).

Second, a large-scale system usually has a specific structure that can be taken advantage of. For example, the system may be composed of different subsystems that are connected by only a few variables. In this case, the Lagrangian decomposition (Guignard and Kim, 1987) method can be applied to decompose the optimization problem approximately into the subproblems for the different subsystems, respectively. The subproblems are easier to solve and can be solved in parallel. Then, the whole problem can be solved by solving these easier subproblems iteratively until the error in the optimal value is within a given tolerance.

Nomenclature

A	Coefficients in the nominal state-space model (3.1)
A_c	Coefficients in the continuous state-space model (B.4)
A_{ext}	Coefficients in the model (B.10) with additional states for time delay
A_r	Coefficients in the uncertain state-space model (3.5)
A_z	Coefficients in the extended state-space model (A.9)
$\widetilde{A}_e$	Coefficients in the state-space model (3.4)
$\widetilde{A}_x$	Coefficients in the state-space model (3.4)
b	Difference between measured and nominally predicted controlled variables
b_{ssr}	Difference between measured and nominally predicted controlled variables at the steady state
B	Coefficients in the nominal state-space model (3.1)
B_c	Coefficients in the continuous state-space model (B.4)
B_d	Coefficients in the nominal state-space model (3.1)
B_{dr}	Coefficients in the uncertain state-space model (3.5)
B_{ext}	Coefficients in the model (B.10) with additional states for time delay
B_o	Coefficients in the bounds (5.15)
$\widetilde{B}_o$	Extended B_o matrix for the whole prediction horizon
B_r	Coefficients in the uncertain state-space model (3.5d)

B_z	Coefficients in the extended process model (A.9)
$\widetilde{B}$	Coefficients in the state-space model (3.4)
$\widetilde{B}_d$	Coefficients in the state-space model (3.4)
c_u	Cost of manipulated variables used in steady-state optimization
c_y	Cost of controlled variables used in steady-state optimization
C	Coefficients in the nominal state-space model (3.1)
$\widetilde{C}$	Coefficients in the state-space model (3.4)
C_c	Coefficients in the continuous state-space model (B.4)
C_{ext}	Coefficients in the model (B.10) with additional states for time delay
C_r	Coefficients in the uncertain state-space model (3.5)
C_x	Cost of state variables
C_z	Coefficients in the extended process model (A.10)
C_u	Cost of the manipulated variables
C_x	Cost of the state variables
C_{I_1}	Cost of I_1
C_{I_2}	Cost of I_2
C_{I_3}	Cost of I_3
C_{IP-SKU}	The constant to convert the product unit into the intermediate product unit
C_{IO}	Cost of the inventories and backorders
C_{F_4}	Cost of transportation

C_O	Penalty (cost) on back orders
C_P	Cost of intermediate product manufacturing
C_S	Cost of the products in the transportation
C_{T_s}	Cost of the product manufacturing
d_m	Measured or predicted disturbances
$\mathbf{d}_m$	d_m in the future p time steps
d_{mss}	Predicted steady-state disturbances
$\tilde{\mathbf{d}}_m$	d_m in the future $n+p$-1 time steps
d_r	Real values of measured or predicted disturbances
$\delta\mathbf{d}_r$	Differences between the measured or predicted disturbances and their real values over the prediction horizon
d_s	Virtual steady-state measured disturbances used in the deviation model
D	Customer demands to the regional distribution centers
e	Unmeasured disturbances
$\hat{e}$	Estimated unmeasured disturbances from the output measurements using nominal model
$\hat{e}_r$	Estimated unmeasured disturbances from the output measurements using the uncertain model representing the real plant
e_s	Virtual steady-state feedback used in the deviation model
e_{ss}	Unmeasured disturbances at steady state
F_1	Flow of the intermediate products entering the plant storage
$F_{1,\max}$	Upper bound on F_1
$F_{1,\min}$	Lower bound on F_1

F_2	Flow of the intermediate products to be processed in the manufacturing plant
F_3	Flow of the products entering the plant distribution center
F_4	Shipment of the products to the regional distribution centers
$F_{4,\max}$	Transportation capacity for the regional distribution centers
F_5	Flow of the products arriving at the regional distribution centers
F_6	Flow of the sent to the customers
G_d	The coefficients in equation (D.12)
G_t	The coefficients in equation (D.12)
G_{udr}	Coefficients in the uncertain closed-loop model (4.16) for steady state
G_{ur}	Coefficients in the uncertain closed-loop model (4.16) for steady state
G_w	Coefficients of unmeasured disturbances in the model (A.11)
G_{wx}	Coefficients of unmeasured disturbances in the model (A.11) for states
G_{we}	Coefficients of unmeasured disturbances in the model (A.11) for feedback
G_{ydr}	Coefficients in the uncertain closed-loop model (4.16) for steady state
G_{yr}	Coefficients in the uncertain closed-loop model (4.16) for steady state
$G_{\zeta\zeta}$	Coefficients in the state-estimation model (3.50)
$G_{\zeta\mu}$	Coefficients in the state-estimation model (3.50)
G_{ξ}	The coefficients in equation (D.12)
$G_{\xi d}$	The coefficients in equation (D.13)
$G_{\xi t}$	The coefficients in equation (D.13)

$G_{\xi\xi}$	The coefficients in equation (D.13)
$G_{\xi\omega}$	The coefficients in equation (D.13)
G_{ω}	The coefficients in equation (D.12)
I	Identity matrix
I_d	Pick-up matrix in equation (D.7)
I_{d2}	Pick-up matrix in equation (D.16)
I_{pd}	Pick-up matrix in equation (D.1)
I_{pu}	Pick-up matrix in equation (3.8)
I_{δ}	Matrix to specify saturation of manipulated variables
$\mathbf{I}_{\delta}$	Extended I_{δ} for the future n time steps
$I_{\delta,\, ss}$	Matrix to specify saturation of manipulated variables in steady state model
$I_{\Delta 1}$	Matrix in the objective function (3.4)
$I_{\Delta 2}$	Matrix in the objective function (3.4)
I_1	Inventory of intermediate products at plant
I_2	Inventory of products at plant distribution center
I_3	Inventory of products at regional distribution center
K_d	Coefficients in the nominal steady state model (4.5)
K_{dr}	Coefficients in the uncertain steady state model (4.3)
K_e	Coefficients in the control law (3.12)

K_r	Coefficients in the uncertain steady state model (4.3)
K_u	Coefficients in the control law (3.12)
K_x	Coefficients in the control law (3.12)
K_{CA0}	Uncertain gain in models (4.64) or (4.67) for CSTR control system 3 or 4
L_e	Linear steady-state Kalman gain for feedback vector e
L_{ur}	Uncertain coefficients in the closed-loop model (3.14)
L_x	Linear steady-state Kalman gain for states vector x
L_{yr}	Uncertain coefficients in the closed-loop model (3.14)
L_z	Linear steady-state Kalman gain for the extended state vector z
M_{ur}	Uncertain coefficients in the closed-loop model (3.14)
M_{yr}	Uncertain coefficients in the closed-loop model (3.14)
n	Control horizon
n_d	Number of measured or predicted disturbances
$n_{decision}$	Number of decision variables of an optimization problem
n_e	Number of feedback variables
n_s	Number of samples for Monte Carlo sampling
n_u	Number of manipulated variables
n_w	Number of unmeasured disturbances (not including the measurement noises on controlled variables)
n_x	Number of state variables

n_y	Number of controlled variables
$n_{IP/SKU}$	Number of intermediate product and final product types
n_{LC}	Number of linear constraints of an optimization problem
n_{RDC}	Number of regional distribution centers
n_{SOC}	Size of a second order cone
N_u	Coefficients in the closed-loop model (3.14)
N_y	Coefficients in the closed-loop model (3.14)
N_{SOC}	Number of second order cones in an optimization problem
O	Back orders at the regional distribution centers
$\mathbf{O}$	O over the prediction horizon
p	Prediction horizon
p_{obs}	Observer horizon
p_-	The backward horizon for state-estimation under uncertainty
P	Manufacturing rate of the intermediate products
q	Diagonal elements in the weighting matrix Q
Q	Weighting matrix for controlled variables
$\tilde{Q}$	Extended weighting matrix for controlled variables in the prediction horizon
Q_{u1}	Uncertain coefficients in equation (D.14)
Q_{u2}	Uncertain coefficients in equation (D.14)

Q_{u3}	Uncertain coefficients in equation (D.14)
Q_w	Covariance matrix of unmeasured disturbances w_z
Q_{x1}	Uncertain coefficients in equation (D.15)
Q_{x2}	Uncertain coefficients in equation (D.15)
Q_{x3}	Uncertain coefficients in equation (D.15)
r	Diagonal elements in the weighing matrix R
r_u^+	The threshold parameter to transform chance constraints – for u upper bound
r_u^-	The threshold parameter to transform chance constraints – for u lower bound
r_x^+	The threshold parameter to transform chance constraints – for x upper bound
r_x^-	The threshold parameter to transform chance constraints – for x lower bound
R	Weighting matrix for manipulated variables
$\widetilde{R}$	Extended weighting matrix for manipulated variables in the control horizon
R_s	Product manufacturing rate
R_v	Covariance matrix of measurement noises of the controlled variables v
s	Slack variables for controlled variables in the soft constraints
$\mathbf{s}$	s in the future p time steps
s_u	Slack variables for difference between references and targets of manipulated variables, as used in formulation (4.5)
s_y	Slack variables for difference between references and targets of controlled variables, as used in formulation (4.5)
S	The additional state variables used to model the transportation time delay
t	Degrees of freedom of the optimization problem in robust MPC

$\mathbf{t}$	t in the future n time steps
t_{ss}	Degrees of freedom of the optimization problem in robust steady-state optimization
ΔT	Simulation period of the prediction model used in MPC
ΔT_c	Controller execution period
ΔT_P	Decision-making period for manufacturing rate of the intermediate products
T_s	Machine running time for the products
$T_{s,\max}$	Maximum bound on T_s
u	Nominal manipulated variables
$\mathbf{u}$	u in the future n time steps
Δu	Difference between the nominal manipulated variables at two successive time steps
$\Delta\mathbf{u}$	Δu in the future n time steps
u_c	The active bounds on manipulated variables
$\mathbf{u}_c$	u_c in the future n time steps
$u_{\max}$	Upper bounds of manipulated variables
$\mathbf{u}_{\max}$	$u_{\max}$ in the future n time steps
$u_{\min}$	Lower bounds of manipulated variables
$\mathbf{u}_{\min}$	$u_{\min}$ in the future n time steps
u_r	Real manipulated variables
$\mathbf{u}_r$	u_r in the future n time steps
Δu_r	Difference between the real manipulated variables at two successive time steps

$\Delta\mathbf{u}_r$	Δu_r in the future n time steps
u_s	Virtual steady-state manipulated variables used in the deviation model
u_{sp}	Targets of manipulated variables obtained by steady-state optimization
$\mathbf{u}_{sp}$	u_{sp} in the future n time steps
$\hat{u}_{sp}$	Targets of the manipulated variables that will be obtained by steady-state optimization in the future
u_{ss}	Nominal manipulated variables at steady state
u_{ssr}	Real manipulated variables at steady state
$u^{(ref)}$	References of manipulated variables obtained by upper-lever nonlinear real-time optimization or from experience
v	Noises on the manipulated variables
V_u	Covariance matrix in the SOCP formulation (3.43)
V_{uss}	Covariance matrix in the SOCP formulation (4.20)
V_y	Covariance matrix in the SOCP formulation (3.43)
V_{yss}	Covariance matrix in the SOCP formulation (4.20)
Vol	Volume (m^3)
w	Diagonal elements in weighing matrix W
w_d	Noises on the measurements of disturbances
w_k	Unmeasured disturbances
w_s	Diagonal elements in weighing matrix W_s
w_z	Unmeasured disturbances on the state variables

W	Move suppression matrix
$\widetilde{W}$	Extended move suppression matrix for the future n time steps
W_s	Penalty matrix for the slack variables of the controlled variables
$\widetilde{W}_s$	Extended penalty matrix for the slack variables of the controlled variables in the future p time steps
x	Nominal state variables
$\mathbf{x}$	x in the future p time steps
$\hat{x}$	Estimated state variables from the output measurements and nominal plant model
$\hat{x}_r$	Estimated state variables from the output measurements and plant model with uncertain parameters
$\widetilde{x}$	Extended state vector that contains additional states for time delay
x_r	Real state variables
$\mathbf{x}_r$	x_r in the future p time steps
$\hat{x}_r$	Estimated state variables from the output measurements using the uncertain model representing the real plant
x_s	Virtual steady-state state variables used in the deviation model
x_{ss}	Nominal state variables at steady state
x_{ssr}	Real state variables at steady state
y	Nominal controlled variables
$\mathbf{y}$	y in the future p time steps
$y_{\max}$	Upper bounds on controlled variables
$\mathbf{y}_{\max}$	$y_{\max}$ in the future p time steps

$y_{\min}$	Lower bounds on controlled variables
$\mathbf{y}_{\min}$	$y_{\min}$ in the future p time steps
y_r	Real controlled variables
$\mathbf{y}_r$	y_r in the future p time steps
$y_{r,m}$	Measured controlled variables
y_s	Virtual steady-state controlled variables used in the deviation model
y_{sp}	Targets of controlled variables obtained by steady-state optimization
$\mathbf{y}_{sp}$	y_{sp} in the future p time steps
$\hat{y}_{sp}$	Targets of controlled variables that will be obtained by steady-state optimization in the future
y_{ss}	Nominal controlled variables at steady state
y_{ssr}	Real controlled variables at steady state
$y^{(ref)}$	References of controlled variables obtained by upper-lever nonlinear real-time optimization or from experience
z	Extended state variables that include x and e
$\hat{z}$	Estimated extended state variables that include $\hat{x}$ and $\hat{e}$
$\hat{z}_r$	Estimated extended state variables that include $\hat{x}_r$ and $\hat{e}_r$
Z_0	Feed composition of light key of the binary distillation column

Greek Letters

α	Constraint-wise confidence level for trajectory optimization
α_{ss}	Constraint-wise confidence level for steady-state optimization
β	Ratios for the virtual routes to model the transportation time uncertainty
γ_u	Weight on the input variance in objective function (4.49)
γ_y	Weight on the output variances in objective function (4.49)
$\gamma_{\Delta u}$	Weight on the variances of input changes in objective function (4.49)
τ	Number of delayed time steps, or transportation time from the plant distribution center to the regional centers in the supply chain system
τ_n	Nominal transportation time
$\tau_{\max}$	Maximum transportation time
$\tau_{\min}$	Minimum transportation time
ζ	Uncertain estimated state in equation (3.50)
θ	Time delay (continuous variable)
$\boldsymbol{\theta}$	Vector in the closed-loop model (3.14), containing the effects from "the past"
λ^+	Lagrange multipliers of the upper bounds on manipulated variables in optimization problem, as in equations (4.7)
$\boldsymbol{\lambda}^+$	λ^+ in the future n time steps, as in equation (3.6)
λ^-	Lagrange multipliers of the lower bounds on manipulated variables in optimization problem, as in equation (4.7)
$\boldsymbol{\lambda}^-$	λ^- in the future n time steps, , as in equation (3.6)
μ	The quantities used to estimate the uncertain states in equation (3.50).

ξ	Vector defined in equation (D.12)
$\boldsymbol{\pi}$	Extended version of $\boldsymbol{\theta}$ with the closed-loop model integrating state-estimation uncertainty (as defined by equation (3.55)
Φ_r	The set containing indices of uncertain realizations
ω	Vector containing all the unmeasured stochastic disturbances

Indexing Subscripts

i	Index for an arbitrary manipulated variable for the discussion in Section 3.3.2; Or index for intermediate or final products for supply chain modeling
j	Index for an arbitrary time step for the discussion in Section 3.3.2; Or index for regional distribution centers in the supply chain system
k	Time step index
$k/k-1$	Index to indicate the estimated value for time step k according to the information at time step $k-1$.
l	Index for the rows for a matrix or the elements of a vector
o_1	Index for the elements in the sub-matrices discussed in Section 3.3.2
o_2	Index for the elements in the sub-matrices discussed in Section 3.3.2

Indexing Superscripts

k_1	Index for the sub-matrices in the closed-loop model coefficient matrices discussed in Section 3.3.2
k_2	Index for the sub-matrices in the closed-loop model coefficient matrices discussed in Section 3.3.2
$(\cdot)'$	The prime symbol indicates the variable in the parentheses is a deviation variable used to handle time-invariant uncertainty in robust MPC

List of References

Alamir, M., and G. Bornard. "Stability of a truncated infinite constrained receding horizon scheme: The general discrete nonlinear case." *Automatica*, 31 (1995): 1353-1356.

Allgower, F., T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. "Nonlinear Predictive Control and Moving Horizon Estimation - An Introductory Overview." Chapter in *Advances in Control (Highlights of ECC 99)*. Springer, 1999.

Allgower, F., and A. Zheng. *Nonlinear Model Predictive Control.* Birkhauser, 2000.

Badgwell, T. A. "Robust model predictive control of stable linear systems", *International Journal of Control*, 68 (1997): 797-818.

Badgwell, T. A., and S. J. Qin. "Nonlinear Predictive Control." Chapter in *Review of Nonlinear Model Predictive Control Applications.* IEE Control Engineering Series, 2001.

Balakrishnan, N., and A. P. Basu. *The Exponential Distribution: Theory, Methods, and Applications.* New York: Gordon and Breach, 1996.

Bellman, R. E. *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

Bemis, B. M., L. J. Bain, and J. J. Higgins. "Estimation and hypothesis testing for the parameters of a bivariate exponential distribution." *Journal of the American Statistical Association*, 67 (1972): 927-929.

Bemporad, A. "A predictive controller with artificial Lyapunov function for linear system with input/state constraints." *Automatica*, 34 (1998a): 1255-1260.

Bemporad, A. "Reducing conservativeness in predictive control of constrained systems with disturbance." *Proceedings of the IEEE Conference on Decision and Control* (1998b): 1384-1391.

Bemporad, A, W. P. M. H. Heemels, and B. De Schutter. "On hybrid systems and closed-loop MPC systems." *IEEE Transaction on Automatic Control*, 47 (2002): 863-869.

Bemporad, A. and M. Morari. "Control of systems integrating logic, dynamics and constraints. " *Automatica*, 35 (1999a): 407-427.

Bemporad, A., and M. Morari. "Robust model predictive control: a survey." *Lecture Notes in Control and Information Sciences*, 245 (1999b): 207-226.

Ben-Tal, A., and A. Nemirovski. "Robust optimization – methodology and applications." *Mathematical Programming*, 92 (2002): 453-480.

Ben-Tal, A., and A. Nemirovski. "Robust solutions of uncertain linear programs." *OR Letters,* 25 (1999): 1-13.

Bertsekas, D. P. *Dynamic Programming and Optimal Control.* 2nd edition. Belmont, MA: Athena Scientific, 2000.

Bertsekas, D. P., and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. New York: Athena Scientific, 1998.

Biegler, L. T., and I. E. Grassmann. "Retrospective on optimization." *Computers & Chemical Engineering*, 28 (2004): 1169-1192.

Braun, M. W., D. E. Rivera, M. E. Flores, W. M. Carlyle, and K. G. Kempf. "A model predictive control framework for robust management of multi-product, multi-echelon demand networks." *Annual Reviews in Control*, 27 (2003): 229-245.

Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control.* Hoboken, NJ: John Wiley, 2008.

Box, G. E. P., and J. F. MacGregor. "Parameter estimation with closed-loop operating data." *Technometrics*, 18 (1976), 366-375.

Boyd, S., and L. Vandenberge. *Convex Optimization.* Cambridge, UK: Cambridge University Press, 2004.

Business Dictionary. http://www.businessdictionary.com/definition/backorder.html, retrieved 2009.

Camacho, E. F., and C. Bordons. "Nonlinear model predictive control: An introductory review." *Lecture Notes in Control and Information Sciences*, 358 (2007): 1-16.

Camacho, E. F., and C. Bordons. *Model Predictive Control.* New York: Springer, 1999.

Charnes, A., and W. W. Cooper (1958). "Chance-constrained programming." *The Fifth International Meeting, The Institute of Management Sciences*, Philadelphia.

Chen, H., and F. Allgower. "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability." *Automatica*, 34 (1998): 1205-1217.

Chopra, S., and P. Meindl. *Supply Chain Management: Strategy, Planning, and Operation.* Upper Saddle River, NJ: Prentice-Hall, 2001.

Clark, P. A., and A. W. Westerberg. "Optimization for design problems having more than one objective." *Computers and Chemical Engineering*, 7 (1983): 259–278.

Colson, B., P. Marcotte, and G. Savard. "An overview of bilevel optimization." *Annals of Operations Research*, 153 (2007): 235-256.

Davison, E. J., and H. W. Smith. "Pole assignment in linear time-invariant multivariable systems with constant disturbances." *Automatica*, 7 (1971): 489-498.

Darlington, J., C. C. Pantelides, B. Rustem, and B. A. Tanyi. "Decreasing the sensitivity of open-loop optimal solutions in decision making under uncertainty." *European Journal of Operational Research,* 121 (2000): 343-362.

Darlington, J., C. C. Pantelides, B. Rustem, and B. A. Tanyi. "An algorithm for constrained nonlinear optimization under uncertainty." *Automatica,* 35 (1999):

217-218.

Dougherty, D., and D. Cooper. "A Practical Multiple Model Adaptive Strategy for Multivariable Model Predictive Control." *Control Engineering Practice*, 11 (2003): 649-664.

Dua, V., N. A. Bozinis, and E. N. Pistikopoulos. "A multiparametric programming approach for mixed-integer quadratic engineering problems." *Computers & Chemical Engineering*, 26 (2002): 715-733.

Esmaili, A., J. F. MacGregor, and P. A. Taylor. "Direct and Two-Step Methods for Closed-Loop Identification: A Comparison of Asymptotic and Finite Data Set Performance." *Journal of Process Control*, 10 (2000), 525-527.

Floudas, C. A. *Deterministic Global Optimization: Theory, Methods and Applications*. Norwell, MA: Kluwer Academic Publishers, 1999.

Floudas, C. A. *Nonlinear and Mixed-Integer Optimization: Fundamentals and applications*. New York: Oxford University Press, 1995.

Forrester, J. W. *Industrial Dynamics*. Cambridge, MA: MIT press, 1961.

Forrester, J. W. "Industrial dynamics: a major breakthrough for decision makers." *Harvard Business Review*, 36 (1958), 37–66.

Friedman, Y., and G. V. Reklaitis. "Flexible solutions to linear programs under uncertainty: inequality constraints," *AIChE Journal*, 21 (1975a): 77-83.

Friedman, Y., and G. V. Reklaitis. "Flexible solutions to linear programs under uncertainty: equality constraints." *AIChE Journal*, 21 (1975b), 83-90.

Fukushima, H., T. H. Kim, and T. Sugie. "Adaptive Model Predictive Control for a Class of Constrained Linear Systems Based on the Comparison Model." *Automatica*, 43 (2007): 301-308.

Gilbert, E. G., and K. T. Tan. "Linear systems with state and control constraints: the theory and applications of maximal output admissible sets." *IEEE Transaction* *on Automatic Control*, 36 (1991): 1008-1020.

Golub, G. H., and C. F. Van Loan. *Matrix computations.* 3rd edition. Baltimore, MD: The Johns Hopkins University Press, 1996.

Goulart, P. J., E. C. Kerrigan, and J. M. Maciejowski. "Optimization over state feedback policies for robust control with constraints." *Automatica*, 42 (2006): 523-533.

Grancharova, A., and T. Johansen "Computation, approximation and stability of explicit feedback min–max nonlinear model predictive control." *Automatica,* 45(2009):1134–1143.

Grassmann, I. E., and L. T. Biegler. "Future perspective on optimization." *Computers & Chemical Engineering*, 28 (2004): 1193-1218.

Grossmann, I. E., and A. W. Westerberg. "Research challenges in process systems engineering." *AIChE Journal*, 46 (2000): 1700-1703.

Guignard, M., and S. Kim. "Lagrangian decomposition: A model yielding stronger Lagrangian bounds." *Mathematical Programming,* 39 (1987): 215-228.

Haddad, W. M., and V. Chellaboina. *Nonlinear Dynamic Systems and Control: A Lyapunov-Based Approach*. Princeton, NJ: Princeton University Press, 2008

IBM. "Computer science reaches historic break through." *on IBM official website at* http://www.ibm.com/ibm/ideasfromibm/us/roadrunner/20080609/index.shtml, retrieved 2008.

ILOG Inc. "ILOG CPLEX®: The world's most trusted mathematical programming optimizers." *On ILOG website at* http://www.ilog.com/products/cplex/index.cfm, retrieved 2008.

Kaisare, N. S., J. M. Lee, and J. H. Lee. "Simulation based strategy for nonlinear optimal control: application to a microbial cell reactor." *International Journal of Robust Nonlinear Control*, 13 (2003): 347-363.

Kalman, R. E. "A new approach to linear filtering and prediction problems." *Journal of*

Basic Engineering, 82 (1960): 35–45.

Kassmann, D. E., T. A. Badgwell, and R. B. Hawkins. "A robust steady-state target calculation for model predictive control." *AIChE Journal*, 46 (2000): 1007-1024.

Keerthi, S., and E. Gilbert. "Optimal infinite-horizon feedback laws for general class of constrained discrete-time systems," *Journal of Optimization Theory and Applications*, 57 (1988): 265-293.

Kleywegt, A. J., and A. Shapiro. "Stochastic Optimization." In *the Handbook of Industrial Engineering*. 3rd Edition ed. Gavriel Salvendy, New York: John Wiley (2001), 2625 – 2650.

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

Kouvaritakis, B., Rossiter, A., and Schuurmans, J. "Efficient robust predictive control." *IEEE Transactions on Automatic Control*, 45 (2000): 1545-1549.

Kwakernaak, H., and R. Sivan. *Linear optimal control systems*. USA: Wiley-interscience, 1972.

Lambert, D. M., M. C. Cooper, and J. D. Pagh. "Supply chain management: Implementation issues and research opportunities." *International Journal of Logistics Management*, 9 (1998): 1-19.

Lee, J. M., and J. H. Lee. "Neuro-dynamic programming method for MPC." *DYCOPS6*, Jejudo-Island, 2001.

Lee, J. M., and J. H. Lee. "Approximate dynamic programming strategies and their applicability for process control: A review and future directions." *International Journal of Process Control, Automation and Systems*, 2 (2004), 263-278.

Lee, J. M., J. H. Lee, and J. Pekny. "Model predictive control of uncertain systems via neuro-dynamic programming." *AIChE Annual Meeting*, Los Angeles, 2000.

Lee, H., V. Padmanabhan, and S. Whang. "The Bullwhip Effect in Supply Chains." *Sloan*

Management Review, 39 (1997): 93-102.

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

Li, P., H. Arellano-Garcia, and G. Wozny. "Chance constrained programming approach to process optimization under uncertainty." *Computers & Chemical Engineering*, 32 (2000): 25-45.

Li, P., M. Wendt, and G. Wozny. "A probabilistically constrained model predictive control." *Automatica*, 38 (2002): 1171-1176.

Lin, P., D. S. Wong, S. Jang, S. Shieh and J. Chu. "Controller design and reduction of bullwhip for a model supply chain system using z-transform analysis." *Journal of process control*, 14 (2004): 487-499.

Lin, X., S. L. Janak, and A. Floudas. "A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty." *Computers & Chemical Engineering*, 28 (2004): 1069-1085.

Lobo, M. S., L. Vandenberghe, S. Boyd, and H. Lebret. "Applications of second-order cone programming." *Linear Algebra & Its Applications*, 284 (1998): 193-228.

Luenberger, D. G. "An introduction to observers." *IEEE Transactions on Automatic Control*, 16 (1971): 596-602.

Luo, Z. Q., J. S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge, UK: Cambridge University Press, 1996.

Lustig, I. J., and J. F. Puget. "Program does not equal program: constraint programming and its relationship to mathematical programming." *Interfaces,* 31 (6), 29-53.

Luyben, W. *Process Modeling, Simulation and Control for Chemical Engineers.* 2nd Edition. McGraw-Hill, 1989.

Maciejowski, J. M. *Predictive Control: With Constraints*. Harlow, UK: Prentice Hall, 2002.

Mangoubi, R. S. *Robust Estimation and Failure Detection: A Concise treatment*. Berlin, Germany: Springer – Verlag, 1998.

Marlin, T. E. *Process Control: Designing Processes and Control Systems for Dynamic Performance.* 2nd edition. New York: McGraw-Hill, 2000.

Marlin, T. E. *Process Control: Designing Processes and Control Systems for Dynamic Performance*. New York: McGraw-Hill, 1995.

Marshall, A. W., and I. Olkin. "A multivariate exponential distribution." *Journal of the American Statistical Association*, 62 (1967): 30-44.

Mayne, D. Q., and H. Michalska. "Receding horizon control of nonlinear systems." *IEEE Transactions on Automatic Control*, 35 (1990): 814-824.

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

Mestan, E., M. Turkay, and Y. Arkun. "Optimization of operations in supply chain systems using hybrid systems approach and model predictive control." *Industrial and Engineering Chemistry Research*, 45 (2006): 6493-6503.

Mhaskar, P., N. H. El-Farra, and P. D. Christofides. "Robust Predictive Control of Hybrid Systems: Satisfying Uncertain Schedules Subject to State and Control Constraints," *International Journal of Adaptive Control and Signal Processing*, 22 (2008): 161-179.

Michalska, H., and D. Q. Mayne. "Robust receding horizon control of constrained nonlinear systems." *IEEE Transactions on Automatic Control*, 38 (1993): 1623-1633.

Muske K. R., and T. A. Badgwell. "Disturbance modeling for offset-free linear model predictive control." *Journal of Process Control*, 12 (2002): 617-632.

Muske, K. R., and J. B. Rawlings. "Model predictive control with linear models." *AIChE Journal*, 39 (1993): 262-287.

Nahmias, S. *Production and Operations Analysis.* 4th edition. New York, McGraw-Hill, 2001.

Nise, N. S. *Control Systems Engineering.* 5th edition. John Wiley & Sons, 2008.

Nocedal, J., and S. J. Wright. *Numerical Optimization.* New York: Springer, 1999.

Perea, E., I. Grossmann, E. Ydstie, and T. Tahmassebi. "Dynamic modeling and classical control theory for supply chain management." *Computers and Chemical Engineering*, 24 (2000): 1143-1149.

Perea-Lopez, E., I. Grossmann, and E. Ydstie. "Dynamic modeling and decentralized control of supply chains." *Industrial and engineering Chemistry Research*, 40 (2001): 3369-3383.

Perea-Lopez, E, E. Ydstie, and I. Grossmann. "A model predictive control strategy for supply chain management." *Computers and Chemical Engineering*, 27 (2003): 1201-1218.

Pinedo, M. L. *Planning and Scheduling in Manufacturing and Services.* Springer, 2005.

Prett, D. M., and R. D. Gillette. "Optimization and constrained multivariable control of a catalytic cracking unit." *AIChE National Meeting*, Houston, TX, 1979.

Proschan, F., and P. Sullo. "Estimating the parameters of a multivariate exponential distribution." *Journal of the American Statistical Association*, 71 (1976): 465-472.

Qin, S. J., and T. A. Badgwell. "A survey of industrial model predictive control technology." *Control Engineering Practice*, 11 (2003): 733-764.

Rahideh, A., M. H. Shaheed, and H. J. C. Huijberts, "Stable adaptive model predictive control for nonlinear systems. " *American Control Conference*. Seattle, Washington, USA, 2008.

Rawlings, J. B., and B. R. Bakshi. "Particle filtering and moving horizon estimation." *Computers and Chemical Engineering*, 30 (2006): 1529–1541.

Sakizlis, V., N. V. Dua, J. D. Perkins, and E. N. Pistikopoulos. "Robust model-based tracking control using parametric programming." *Computers & Chemical Engineering*, 28 (2004): 195-207.

Sakizlis, V., N. M. P. Kakalis, V. Dua, J. D. Perkins, and E. N. Pistikopoulos. "Design of robust model-based controllers via parametric programming." *Automatica*, 40 (2004): 189-201.

Sanchez, J. M. M., and J. Rodellar. *Adaptive Predictive Control: From the Concepts to Plant Optimization*. New Jersey: Prentice Hall, 1996.

Sarimveis, H., P. Patrinos, C. D. Tarantilis, and C. T. Kiranoudis. "Dynamic modeling and control of supply chain systems: A review." *Computers & Operations Research*, 35 (2008): 3530-3561.

Schwarm, A. T., and M. Nikolaou. "Chance-constrained model predictive control." *AIChE Journal*, 45 (1999): 1743-1752.

Scokaert, P. O. M., D. Q. Mayne, and J. B. Rawlings. "Suboptimal model predictive control (feasibility implies stability)." *IEEE Transactions on Automatic Control*, 44 (1999): 648-654.

Scokaert, P. O. M., and J. B. Rawlings. "Constrained linear quadratic regulation." *IEEE Transactions on Automatic Control*, 43 (1998): 1163-1169.

Seferlis, P., and N. F. Giannelos. "A two-layered optimization-based control strategy for multi-echelon supply chain networks." *Computers & Chemical Engineering*, 28 (2004): 799-809.

Sen, S., and J. L. Higle. "An introductory tutorial on stochastic linear programming models." *Interfaces*, 29 (1999): 33-61.

Shapiro, J. F. "Challenges of strategic supply chain planning and modeling," *Computers and Chemical Engineering*, 28 (2003): 855-861.

Simchi-Levi, D., P. Kaminski, and E. Simchi-Levi. *Designing and Managing the Supply*

Chain: Concepts, Strategies and Case Studies. New York: McGraw-Hill/Irwin, 1999.

Slotine, J. E., and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Prentice Hall, 1991.

Sterman J. D. *Business dynamics: systems thinking and modeling for a complex world.* New York, McGraw-Hill, 2000.

Sturm, J. F. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones." *Optimization Methods and Software*, 11-12 (1999): 625-653.

Tersine, R. J. *Principles of Inventory and Materials Management.* 4th edition. New Jersey: PRT Prentice Hall, 1994.

Towill, D. R. "Industrial dynamics modeling of supply chains." *Logistics Information Management*, 4 (1996): 43-56.

Towill, D. R. "Dynamic analysis of an inventory and order based production control system," *International Journal of Production Research*, 6 (1982): 671-687.

Tzafestas, S., G. Kapsiotis, and E. Kyriannakis. "Model-based predictive control for generalized production planning problems." *Computers in Industry*, 34 (1997): 201-210.

Van Hessem, D., and O. Bosgra. "Stochastic closed-loop model predictive control of continuous nonlinear chemical processes." *Journal of Process Control*, 16 (2006): 225-241.

Verhaegen M., and V. Verdult. *Filtering and System Identification: A Least Square Approach*. Cambridge, UK: Cambridge University Press, 2007.

Wächter, A., and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." *Mathematical Programming*, 106 (2006): 25-57.

Wang, Y. J., and J. B. Rawlings. "A new robust model predictive control method I: theory and computation." *Computers and Chemical Engineering*, 14 (2004): 231-247.

Wang, W., D. E. Rivera, and K. G. Kempf. "Model predictive control strategies for supply chain management in semiconductor manufacturing." *International Journal of Production Economics*, 107 (2007): 56-77.

Warren, A. *Sequential Decision-Making Under Uncertainty*. Ph.D. thesis, McMaster University, 2004.

Weisstein, E. W. "Spectral Norm, from Mathworld-A wolfram Web Resource." http://mathworld.wolfram.com/SpectralNorm.html, retrieved 2009.

Wold, S., K. Esbensen, and P. Geladi. "Principal component analysis." *Chemometrics and Intelligent Laboratory Systems*, 2 (1987): 37-52.

Xie, L., Y. C. Soh, and C. E. de Souza. "Robust Kalman Filtering for Uncertain Discrete-Time Systems." IEEE Transactions on Automatic Control, 39 (1994): 1310-1314.

Yang, T. H., and E. Polak. "Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty." *International Journal of Control*, 58 (1993): 875-903.

Zavala, V. M., and L. T. Biegler. "The advanced step NMPC controller: optimality, stability and robustness." *Automatica*, 45 (2009): 86-93.

Zavala, V. M., C. D. Laird, and L. T. Biegler. "Fast implementations and rigorous models: can both be accommodated in NMPC?" *International Journal of Robust Nonlinear Control*, 18 (2008): 800-815.

Zhu, Y. C. "Multivariable process identification for MPC: the asymptotic method and its applications." *Journal of Process Control*, 8 (1998): 101-115.

Zlatev, Z. *Computational Methods for General Sparse Matrices*. Dordrecht: Kluwer Academic Publishers, 1991

Appendix A

The Output Feedback and Linear State Estimation in Nominal MPC

The feedback scheme, or how to utilize the output measurements in the controller calculation, is key for offset-free control if unmeasured disturbances enter the process or model mismatch is present. In the context of the nominal MPC using state-space model, the feedback scheme needs to address two issues:

1) A prior structure of the unmeasured disturbances that is used to update the model according to output measurements;
2) An observer to estimate the states and unmeasured disturbances (if they are not measurable).

Once the states and the unmeasured disturbances are estimated, the open-loop optimal control problem will be resolved.

A.1 Output Feedback With Unmeasured Disturbances Model

Muske K. R. and T. A. Badgwell (2002) gave a general model structure for unmeasured disturbances in state-space model:

$$x_{k+1} = Ax_k + Bu_k + G_d e_k \tag{A.1}$$

$$y_{k+1} = Cx_{k+1} + G_p p_{k+1} \tag{A.2}$$

$$e_{k+1} = e_k \tag{A.3}$$

$$p_{k+1} = p_k \tag{A.4}$$

where e denotes the unmeasured state disturbances and p denotes the unmeasured output disturbances. Equations (A.3-A.4) mean the state and output disturbances are assumed to be constant in the prediction horizon.

G_d and G_p are design parameters of the unmeasured disturbance model. According to the theory presented in Muske and Badgwell (2002), for the purpose of offset-free control, the unmeasured disturbance model should be designed such that augmented system (A.1-A.4) is detectable.

There are two typical designs of unmeasured disturbance model:

1) When $G_p = I$ and $G_d = 0$, the disturbance model assumes that a step disturbance is added on the controlled variables and remains constant throughout the prediction horizon. This famous "DMC feedback model" (Qin and Badgwell, 2003; Maciejowski, 2002) has been recognized to be weak for characterizing non-stationary disturbances on the outputs and open-loop unstable processes. Muske and Badgwell (2002) pointed out that such augmented system is not detectable when the process has integrating modes.
2) When $G_p = 0$, the disturbance model assumes that a step disturbance is added on the state variables through a G_d channel and remains constant throughout the prediction horizon. Here G_d is the design parameter of the unmeasured disturbance model. According to Muske and Badgwell (2002), if the outputs of the system are linearly independent, there exists a G_d such that the system (A.1-A.4) is detectable. Davison and Smith (1971) presented this as a standard technique to remove steady-state offset for the Linear Quadratic Regulator (LQR). A special case of this design is to choose $G_d = B$, which means the disturbance model assumes that a step disturbance is added

on the manipulated variables and remains constant throughout the prediction horizon. This feedback scheme is good for both open-loop stable and unstable processes (Muske and Rawlings, 1993).

In this thesis we adopt the framework of the second design, i.e. we write the process and the unmeasured disturbance model as

$$x_{k+1} = Ax_k + Bu_k + B_e e_k \tag{A.5}$$

$$y_{k+1} = Cx_{k+1} \tag{A.6}$$

$$e_{k+1} = e_k \tag{A.7}$$

where B_e denotes the channel through which the unmeasured disturbances affect the states. For example, if we assume the unmeasured disturbances are errors on manipulated variables and they influence the states through the input channel, then $B_e = B$. With this type of design, we can achieve off-set free for all the case studies in this thesis, i.e. for each of the case studies, we can find a parameter B_e such that the system (A.5-A.7) is detectable. Although not having been done in this thesis, the new robust MPC method developed in this thesis can be easily extended to include the more general unmeasured disturbance model (A.1-A.4).

In the case that measurement or prediction on some disturbances is available and we need to use this information in the controller calculation, equation (A.5) can be written as

$$x_{k+1} = Ax_k + Bu_k + B_e e_k + B_d d_{m,k} \tag{A.8}$$

where d_m denotes the measured or predicted disturbances. The model (A.6-A.8) can be rewritten into the following form with the augmented state vector z,

$$z_{k+1} = A_z z_k + B_z \begin{bmatrix} u_k \\ d_{m,k} \end{bmatrix} \tag{A.9}$$

$$y_{k+1} = C_z z_{k+1} \tag{A.10}$$

where $z = \begin{bmatrix} x \\ e \end{bmatrix}$, $A_z = \begin{bmatrix} A & B_e \\ 0 & I \end{bmatrix}$, $B_z = \begin{bmatrix} B & B_d \\ 0 & 0 \end{bmatrix}$, $C_z = \begin{bmatrix} C & 0 \end{bmatrix}$.

A.2 Linear State Estimation with Kalman Filter

The Kalman filter (Kalman, 1960) is an optimal observer that minimizes the effect of the given noise on state estimation (in the least square sense). Assume there are zero-mean, normally distributed independent noise vectors w_z, v_d, v in the following system,

$$z_{p,k+1} = A_z z_{p,k} + B_z \begin{bmatrix} u_k \\ d_{m,k} - v_{d,k} \end{bmatrix} + G_w w_{z,k} \tag{A.11}$$

$$y_{m,k+1} = C_z z_{p,k+1} + v_k \tag{A.12}$$

where z_p denotes the real (extended) states and y_m denotes the measured controlled variables, $G_w = \begin{pmatrix} G_{wx} \\ G_{we} \end{pmatrix}$ denotes the coefficients of w_z, G_{wx} corresponds to states x and G_{we} corresponds to feedback e. Let $w = B_z \begin{pmatrix} 0 \\ -v_d \end{pmatrix} + G_w w_z$, then w obeys zero-mean multivariate normal distribution. Thus equation (A.11) can be written as,

$$z_{k+1} = A_z z_k + B_z \begin{bmatrix} u_k \\ d_k \end{bmatrix} + w_k \tag{A.13}$$

A standard steady-state Kalman filter algorithm (Maciejowski, 2002; Muske and Rawlings, 1993) for system (A.12-A.13) can be expressed as.

$$\hat{z}_{k/k} = \hat{z}_{k|k-1} + L_z(y_k - C_z \hat{z}_{k|k-1}) \tag{A.14}$$

where $\hat{z}_{k/k-1}$ denotes the estimate of the augmented state vector at time step k given the output measurement at time step k-1 and $\hat{z}_{k/k}$ denotes the update of the estimate given the output measurement at time step k. L_z denotes the steady-state Kalman filter gain that can be computed by solving the following algebraic Riccati equation,

$$P = A_z[P - PC_z^{\ T}(C_z P C_z^{\ T} + R_v)^{-1} C_z P] A_z^{\ T} + Q_w \tag{A.15}$$

$$L_z = PC_z^{\ T}(C_z P C_z^{\ T} + R_v)^{-1} \tag{A.16}$$

This observer optimally reconstructs the states from the output measurement given the noise w, v obey zero mean normal distribution and have the covariance Q_w, R_v respectively. If $R_v > 0$, (A_z, C_z) is detectable and $(A_z, G_w Q_w^{1/2})$ is stabilizable, this filter is nominally stable (Maciejowski, 2002; Muske and Rawlings, 1993), i.e., the estimate of the states by the observer will converge to the real states nominally.

We define p_{obs} the time steps for the convergence of the estimate of the states to the real states, i.e., with any given initial estimate error $\delta z_0 = z_0 - \hat{z}_{0/0}$, the estimate error after p_{obs} time steps, $\delta z_{p_{obs}} = z_{p_{obs}} - \hat{z}_{p_{obs}/p_{obs}}$, is negligible. Note that according to (A.13-A.14), we have the following recursive relationship

$$\delta z_k = (A_z - L_z C_z A_z)\delta z_{k-1} \tag{A.17}$$

which means,

$$\begin{aligned}\delta z_{p_{obs}} &= (A_z - L_z C_z A_z)^{p_{obs}} \delta z_0 \\ &= K_{z,obs}\delta z_0\end{aligned} \tag{A.18}$$

Therefore, we can select p_{obs} such that the norm of $K_{z,obs}$ is less than some threshold K_{max}. In this thesis, we set K_{max}=0.05 and use the following spectral norm for $K_{z,obs}$ (Weisstein, 2009),

$$\begin{aligned}\| K_{z,obs} \|_2 &= (\text{maximum eigenvalue of } K_{z,obs}^T K_{z,obs})^{1/2} \\ &= \max_{\|\delta z_0\|_2 \neq 0} \frac{\| K_{z,obs}\delta z_0 \|}{\| \delta z_0 \|_2}\end{aligned} \tag{A.19}$$

So the state estimation error $\| \delta z_{p_{obs}} \|_2 \le K_{\max} = 0.05 \| \delta z_0 \|_2$ for any initial state estimation error δz_0.

The observer algorithm shown by equation (A.14) for the augmented state vector can be partitioned for the states and the unmeasured disturbances of the system respectively,

$$\hat{x}_{k/k} = \hat{x}_{k|k-1} + L_x(y_k - C\hat{x}_{k|k-1}) \tag{A.20}$$

$$\hat{e}_{k/k} = \hat{e}_{k/k-1} + L_e(y_k - C\hat{x}_{k|k-1}) \tag{A.21}$$

where $\begin{bmatrix} L_x \\ L_e \end{bmatrix} = L_z$. For simplicity of the notation, we write $\hat{x}_{k/k}$ as $\hat{x}_k$ and $\hat{e}_{k/k}$ as $\hat{e}_k$ in this thesis.

Appendix B

Integrating Time-Delay in the State-Space Model by Introducing Additional States

B.1 Delay Between Manipulated and State Variables

Let's see the following state-space model without time-delay,

$$x_{k+1} = Ax_k + Bu_k \tag{B.1}$$

where $x_{k+1} = \left(x_{1,k+1}, x_{2,k+1}, \cdots, x_{n_x,k+1}\right)^T \in R^{n_x}$ contains the state variables at time step k+1 and $u_k = \left(u_{1,k}, u_{2,k}, \cdots, u_{n_u,k}\right)^T \in R^{n_u}$ contains the manipulated variables at time step k. Here we are not considering the measured disturbances $d_{m,k}$ explicitly because they can be handled as manipulated variables for the time-delays.

Assume there is a delay of τ time steps between the state variable $x_{i,k+1}$ and the manipulated variable $u_{j,k}$, then define the additional state vector $(x_{delay}^{(i,j)})_k = \left(x_{delay,1,k}^{(i,j)}, \cdots, x_{delay,\tau,k}^{(i,j)}\right)^T \in R^{\tau\times 1}$ where $x_{delay,m,k}^{(i,j)}$ denotes the jth manipulated variable implemented at time step k-m that will affect the ith state variable at the time step k-m+τ +1. Thus,

$$(x_{delay,1}^{(i,j)})_{k+1} = u_{j,k}$$

$$(x_{delay,2}^{(i,j)})_{k+1} = (x_{delay,1}^{(i,j)})_k$$

$$\vdots$$

$$(x_{delay,\tau}^{(i,j)})_{k+1} = (x_{delay,\tau-1}^{(i,j)})_k$$

$$x_{i,k+1} = \sum_{l=1}^{n_x} A_{i,l} x_{i,k} + B_{i,j}(x_{delay,\tau}^{(i,j)})_k + \sum_{jj=1}^{j-1} B_{i,jj} u_{jj,k} + \sum_{jj=j+1}^{n_u} B_{i,jj} u_{jj,k}$$

$$x_{ii,k+1} = \sum_{l=1}^{n_x} A_{ii,l} x_{ii,k} + \sum_{l=1}^{n_u} B_{ii,l} u_{l,k} \qquad ii = 1,\cdots,i-1,\, i+1,\cdots n_x$$

where $A_{i,l}$ denotes the element of A at the ith row and lth column, $B_{i,j}$ denotes the element of B at the ith row and jth columen. The above equations can be summarized in the following canonical state-space model

$$\begin{pmatrix} x_{delay,1}^{(i,j)} \\ x_{delay,2}^{(i,j)} \\ \vdots \\ x_{delay,\tau}^{(i,j)} \\ x \end{pmatrix}_{k+1} = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \vdots & \vdots & \vdots \\ \vdots & \ddots & 0 & \vdots & \vdots \\ \vdots & \cdots & 1 & 0 & 0 \\ 0 & \cdots & 0 & B_{i,j}^{+} & A \end{bmatrix} \begin{pmatrix} x_{delay,1}^{(i,j)} \\ x_{delay,2}^{(i,j)} \\ \vdots \\ x_{delay,\tau}^{(i,j)} \\ x \end{pmatrix}_{k} + \begin{pmatrix} I_{1,j}^{+} \\ B_{i,j}^{-} \end{pmatrix} u_k \tag{B.2}$$

where $B_{i,j}^{+} \in R^{n_x}$ with the ith element being $B_{i,j}$ and other elements being 0, $B_{i,j}^{-} \in R^{n_x \times n_u}$ with the ith element being 0 and other elements equivalent to the corresponding elements in B, $I_{1,j}^{+} \in R^{\tau \times n_u}$ with the jth element of the first row being 1 and other elements being 0.

More delays between state variables and manipulated variables can be integrated one by one into the state-space model in the same way.

B.2 Delay Between Manipulated and Controlled Variables

Here we consider the Single Input Single Output (SISO) system first. Assume the following general input-output model in s-domain and there is no time-delay between the controlled (output) variable y and the manipulated (input) variable u:

$$y(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1 s + a_0} u(s) \tag{B.3}$$

The above model can be transformed into state-space model in time domain (Nise, 2008):

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{B.4}$$

$$y(t) = C_c x(t) \tag{B.5}$$

where

$$A_c = \begin{bmatrix} -a_{n-1} & -a_{n-2} & -a_{n-3} & \cdots & -a_0 \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad C_c = \begin{bmatrix} b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix}.$$

With a proper sampling time Δt, we can discretize the above state-space model into

$$x_{k+1} = Ax_k + Bu_k \tag{B.6}$$

$$y_{k+1} = Cx_{k+1} \tag{B.7}$$

Details of the transformation can be found in Nise (2008). Now let's consider the time delay θ (which becomes $\theta/\Delta t=\tau$ steps with the sampling time Δt) between y and u. define the additional state vector $(x_{delay}^{(i,j)})_k=\left(x_{delay,1,k}^{(i,j)},\cdots,x_{delay,\tau,k}^{(i,j)}\right)^T\in R^{\tau\times 1}$ where $x_{delay,m,k}^{(i,j)}$ denotes the *j*th manipulated variable implemented at time step *k-m* that will affect the *i*th state variable at the time step $k\text{-}m+\tau+1$. Thus,

$$(x_{delay,1}^{(i,j)})_{k+1}=u_{j,k}$$

$$(x_{delay,2}^{(i,j)})_{k+1}=(x_{delay,1}^{(i,j)})_k$$

$$\vdots$$

$$(x_{delay,\tau}^{(i,j)})_{k+1}=(x_{delay,\tau-1}^{(i,j)})_k$$

$$x_{k+1}=Ax_k+B_{i,j}(x_{delay,\tau}^{(i,j)})_k$$

$$y_{k+1}=Cx_{k+1}$$

The above equations can be rewritten into the following vector form:

$$\begin{pmatrix} x_{delay,1}^{(i,j)} \\ x_{delay,2}^{(i,j)} \\ \vdots \\ x_{delay,\tau}^{(i,j)} \\ x \end{pmatrix}_{k+1}=\begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \vdots & \vdots & \vdots \\ \vdots & \ddots & 0 & \vdots & \vdots \\ \vdots & \cdots & 1 & 0 & 0 \\ 0 & \cdots & 0 & B & A \end{bmatrix}\begin{pmatrix} x_{delay,1}^{(i,j)} \\ x_{delay,2}^{(i,j)} \\ \vdots \\ x_{delay,\tau}^{(i,j)} \\ x \end{pmatrix}_k+\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}u_k \tag{B.8}$$

$$y_{i,k+1}=\begin{pmatrix} 0 & \cdots & 0 & C \end{pmatrix}\begin{pmatrix} x_{delay,1}^{(i,j)} \\ x_{delay,2}^{(i,j)} \\ \vdots \\ x_{delay,\tau}^{(i,j)} \\ x \end{pmatrix}_{k+1} \tag{B.9}$$

or simply

$$\tilde{x}_{k+1} = A_{ext}\tilde{x}_k + B_{ext}u_k \tag{B.10}$$

$$y_{k+1} = C_{ext}\tilde{x}_{k+1} \tag{B.11}$$

where $\tilde{x}_{k+1} = \left(x^{(i,j)}_{delay,1,k+1}, \cdots, x^{(i,j)}_{delay,\tau,k+1}, x_{k+1}\right)^T$ denotes the extended state vector for the system. Note that if the delay step $\tau = 0$, the extended state-space model (B.10-B.11) will degrade into the original model without time delay (B.6-B.7).

Now Let's consider the Multi-Input Multi-Output (MIMO) system. Assume the delay between the *i*th controlled variable y_i and the *j*th manipulated variable u_j is $\tau_{i,j}$ time steps. Then according to the above derivation for SIMO system, the relationship between y_i and u_j can be written as

$$\tilde{x}^{(i,j)}_{k+1} = A^{(i,j)}_{ext}\tilde{x}^{(i,j)}_k + B^{(i,j)}_{ext}u_{j,k} \tag{B.12}$$

$$y_{i,k+1} = C^{(i,j)}_{ext}\tilde{x}^{(i,j)}_{k+1} \tag{B.13}$$

Repeat doing the same thing for all the controlled variable-manipulated variable pairs, then the whole MIMO system can be written as

$$
\begin{pmatrix} \tilde{x}^{(1,1)} \\ \vdots \\ \tilde{x}^{(n_y,1)} \\ \vdots \\ \tilde{x}^{(1,n_u)} \\ \vdots \\ \tilde{x}^{(n_y,n_u)} \end{pmatrix}_{k+1} = \begin{pmatrix} A_{ext}^{(1,1)} & & & & & & \\ & \ddots & & & & & \\ & & A_{ext}^{(n_y,1)} & & & & \\ & & & \ddots & & & \\ & & & & A_{ext}^{(1,n_u)} & & \\ & & & & & \ddots & \\ & & & & & & A_{ext}^{(n\ ,n_u)} \end{pmatrix} \cdot \begin{pmatrix} \tilde{x}^{(1,1)} \\ \vdots \\ \tilde{x}^{(n_y,1)} \\ \vdots \\ \tilde{x}^{(1,n_u)} \\ \vdots \\ \tilde{x}^{(n_y,n_u)} \end{pmatrix}_{k}
+ \begin{pmatrix} B_{ext}^{(1,1)} & & & \\ \vdots & & & \\ B_{ext}^{(n_y,1)} & & & \\ & \vdots & & \\ & & \ddots & \\ & & & B_{ext}^{(1,n_u)} \\ & & & \\ & & & B_{ext}^{(n_y,n_u)} \end{pmatrix} \cdot u_k \tag{B.14}
$$

$$
y_{k+1} = \begin{pmatrix} C_{ext}^{(1,1)} & \ddots & C_{ext}^{(n_y,1)} & \cdots & C_{ext}^{(1,n_u)} & \ddots & C_{ext}^{(n_y,n_u)} \end{pmatrix} \begin{pmatrix} \tilde{x}^{(1,1)} \\ \vdots \\ \tilde{x}^{(n_y,1)} \\ \vdots \\ \tilde{x}^{(1,n_u)} \\ \vdots \\ \tilde{x}^{(n_y,n_u)} \end{pmatrix}_{k+1} \tag{B.15}
$$

Appendix C

Additional Discussion on the Prediction of Time-Invariant Uncertainty

Appendix C gives additional discussion on the prediction of time-invariant uncertainty, which complements the discussion in Section 3.2.5 in Chapter 3. In Section C.1 we discuss the reasons to express the variables as deviations from their "virtual" initial steady states values. In Section C.2 we discuss how to address the time delays when determining the virtual steady state.

This appendix deals exclusively with the uncertainty in the feedback model and does not address the uncertainty due to unmeasured disturbances. Before giving the details of the formulations and results, we need to emphasize the proper predictions required by the prediction model in the robust MPC framework. First, formulation should correctly predict the uncertainty bounds around the nominal model due to changes in the manipulated variables in response to an input change such as a set point change. Second, the formulation should predict zero uncertainty when the system is at steady state and no input change is introduced. Three formulations are presented here, only one of which satisfies both of these requirements.

C.1 Dynamic model formulations with uncertainty predictions

If the uncertain parameters of the plant do not change over time or they change slowly so that we can assume they are invariant in the prediction horizon, the formulation can be considered to be time-invariant. The uncertain process model (3.5b-3.5d) can be simplified by setting the unmeasured disturbances to zero, which will enable us to see clearly the effect of feedback model uncertainty. The resulting model is given in the following.

Prediction Model C.1:

$$x_{r,k+1} = A_r x_{r,k} + B_r u_{r,k} + B_{er} e_{r,k} + B_{dr} d_{m,k} \tag{C.1}$$

$$e_{r,k+1} = e_{r,k} \tag{C.2}$$

$$y_{r,k+1} = C_r x_{r,k+1} \tag{C.3}$$

$$k = 0, \cdots, p-1$$

where the parameters A_r, B_r, B_{er}, B_{dr}, C_r are uncertain but time-invariant. Now we will use an example case study to demonstrate the drawback of using the equations (C.1-C.3) for prediction. The case study is for the CSTR control system 1 described in Section 3.5.1. The plant behavior in this study, i.e., the realization from the infinite set of possible plant realizations in the uncertainty description, is the same as the nominal model. The simulation results of this system with robust MPC for a single step set point change are shown in Figure C.1.

Now, we will examine the closed-loop uncertainty predicted by the robust MPC controller using the model (C.1-C.3) at the two different conditions: (a) the time step when the set point change just occurs and (b) an initial steady state without input change. Figure C.2 shows the uncertainty predictions of u and y. We can find from Figure C.2 (a) that both u and y are predicted to be uncertain (i.e. different for different plant realizations) in the short-term future, y is predicted to have zero offset by the end of the horizon (because of the integral mode), and u is predicted to be uncertain at the end of the horizon (because of the gain uncertainty). These uncertainty results meet our expectations. (Recall that unmeasured disturbances are not included in this discussion, so that the controlled and manipulated variables are constant at the end of the transient.)

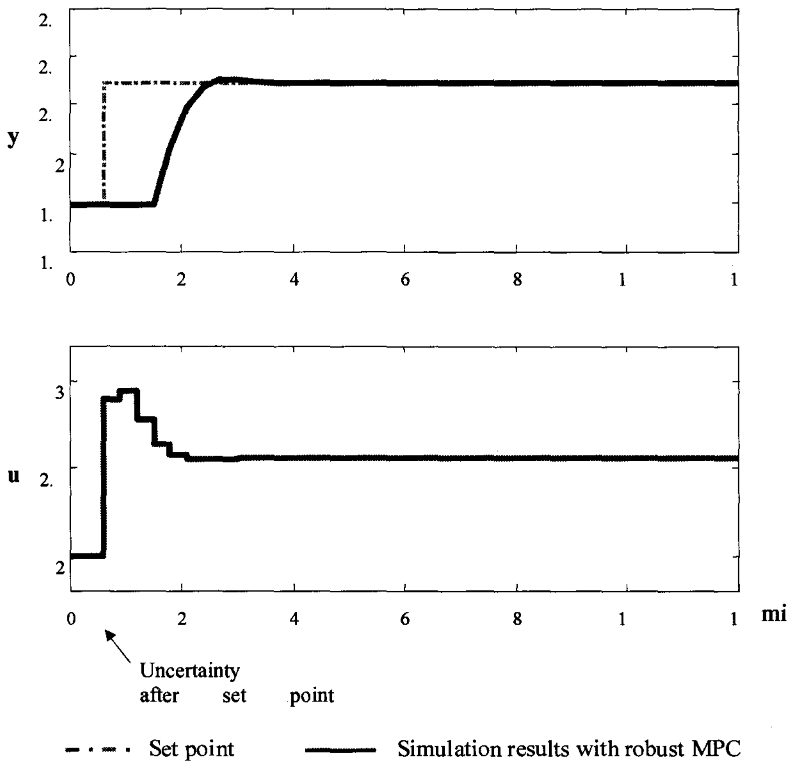


Figure C.1 The simulation results of the example case with robust MPC

However, Figure C.2 (b) shows that, at steady state, both u and y are predicted to be uncertain in the future prediction horizon. This prediction is too conservative because we know the real time-invariant plant is correctly depicted by the nominal model and the feedback at steady state, and there should not be any uncertainty in the future. This conservative prediction could lead to conservative control, e.g., maintaining an excessive safety margin from controlled variable constraints. The reason for this result is that the uncertainty is expressed as a multiple of the controller degrees of freedom, "t", and since this is non-zero, the uncertainty will be non-zero.

One way to avoid this conservativeness is to express variables as deviation variables, so that they will be zero at the steady state about which the deviation is calculated. A second formulation in the equations (C.1-C.3) expresses variables as deviations from their *final steady states*, as shown in the following.

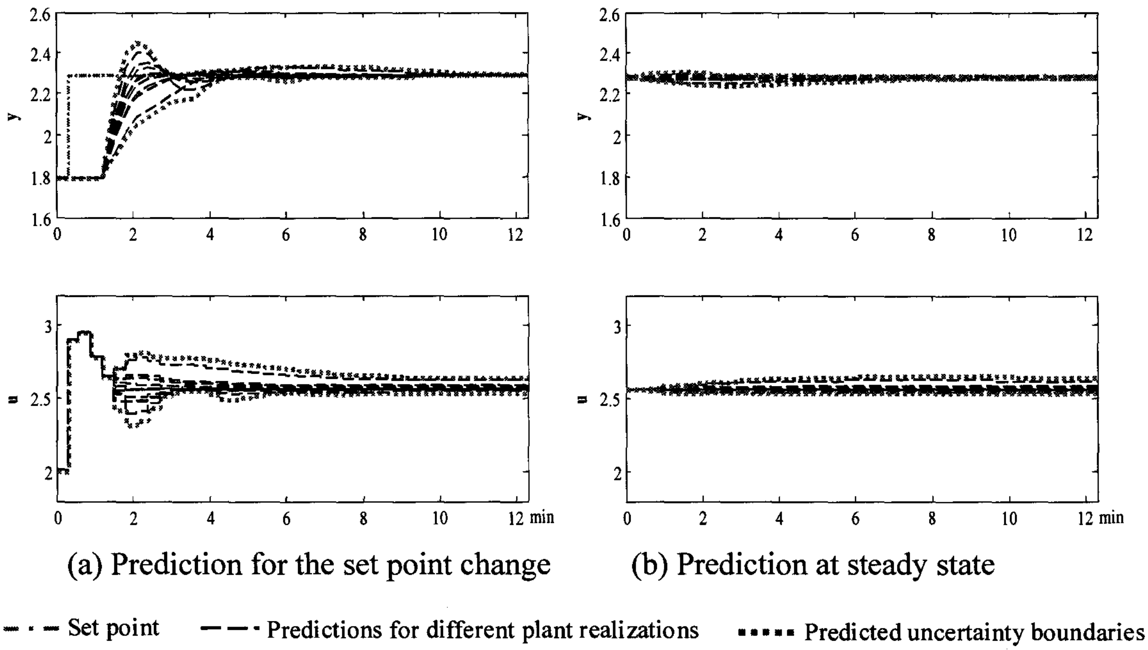


Figure C.2 The uncertainty in prediction models using Prediction Model C.1

Prediction Model C.2:

$$x_{r,k+1} - x_{ss} = A_r(x_{r,k} - x_{ss}) + B_r(u_{r,k} - u_{ss}) + B_{er}(e_{r,k} - e_{ss}) + B_{dr}(d_{m,k} - d_{ss}) \tag{C.4}$$

$$e_{r,k+1} = e_{r,k} \tag{C.5}$$

$$y_{r,k+1} - y_{ss} = C_r(x_{r,k+1} - x_{ss}) \tag{C.6}$$

$$k = 0, \cdots, p-1$$

where x_{ss}, y_{ss}, u_{ss}, d_{ss}, e_{ss} are the values of the process variables at the final steady state, which can be obtained using the steady-state optimization method discussed in Chapter 4. This deviation method was suggested by some researchers for set point tracking by robust MPC, e.g. Kothare et al. (1996) and Wang and Rawlings (2004).

Now, we evaluate the prediction model with the deviation variables (C.4-C.6) with the same example case study by examining the uncertainty predictions of the u and y at the set point change and at steady state, respectively. The predictions are shown in Figure C.3. We can find from Figure C.2 (b) that, at the final steady state, both u and y are predicted to be certain, which is the prediction that we expect.

However, Figure C.3 (a) shows that, u is predicted to be certain at the final steady state of the controller prediction horizon. This prediction is incorrect, because y is driven to the set point at the steady state for every realization of the plant, so that u must achieve different steady-state values for different realizations of the plant (with different process gains). Therefore, Prediction Model C.2 is incorrect for Figure C.3a although it is correct for Figure C.2b.

Next we evaluate another deviation model where the deviation variables are not from the final steady state but a "virtual" steady state that would be determined by the

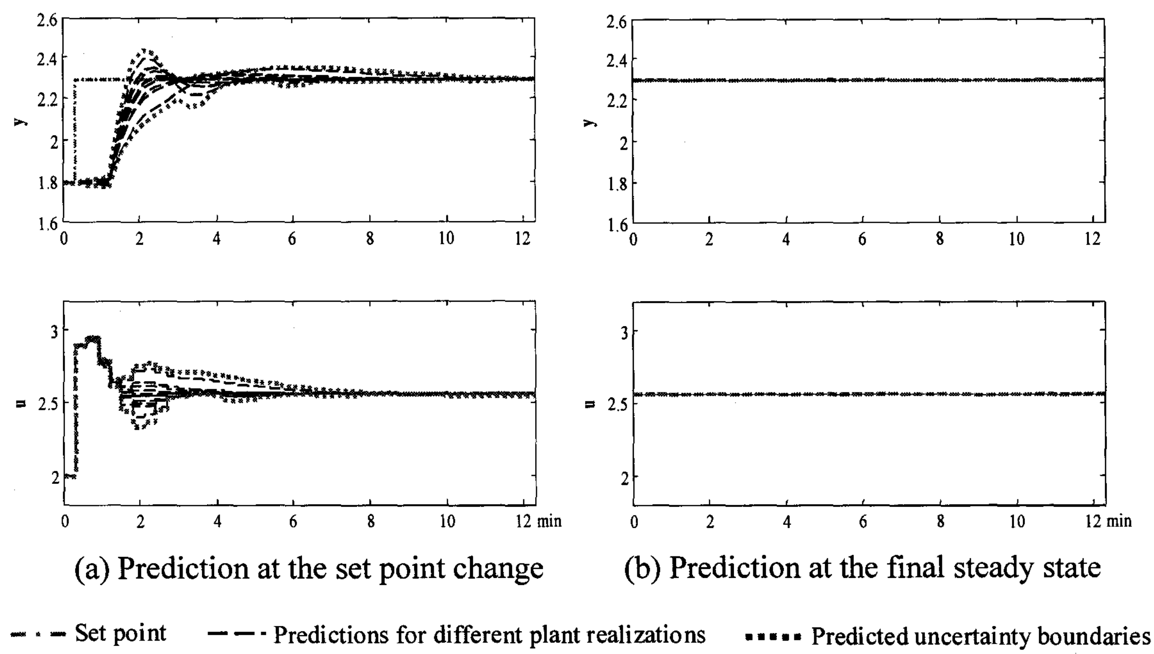


Figure C.3 Uncertainty in prediction models using Prediction Model C.2

most current manipulated variables u_{-1} and the measured disturbances $d_{m,-1}$. This virtual steady state is the steady state that would have been achieved if the controller had been turned off and the processed had been allowed to settle to steady state. Thus, the deviation from this virtual steady state measures the effects of the controller executions from the current situation to the final steady state. A similar idea has been successfully applied to robust steady-state optimization by Kassmann et al. (2000) without explanation. We denote the variables at the virtual steady state as x_s, y_s, u_s, d_s, e_s, where u_s, d_s, e_s are known or are estimated at each controller execution as follows.

$$u_s = u_{-1} \tag{C.7}$$

$$d_s = d_{m,-1} \tag{C.8}$$

$$e_s = \hat{e}_0 \tag{C.9}$$

Then, x_s, y_s can be obtained using the nominal model.

$$x_s = Ax_s + Bu_s + B_e e_s + B_d d_s \tag{C.10}$$

$$y_s = Cx_s \tag{C.11}$$

Then we can express the controller prediction model (C.1-C.3) as the deviation from the virtual steady state as shown in the following.

Prediction Model C.3:

$$x_{r,k+1} - x_s = A_r(x_{r,k} - x_s) + B_r(u_{r,k} - u_s) + B_{dr}(d_{m,k} - d_s) \tag{C.12}$$

$$y_{r,k+1} - y_s = C_r(x_{r,k+1} - x_s) \tag{C.13}$$

$$k = 0, \cdots, p-1$$

The advantage of the Prediction Model C.3 can be illustrated by Figure C.4, which shows the uncertainty predictions of u and y for the example case study for the set point change and at steady state. We find that for the set point change in Figure C.4a, both u and y are different for different plant realizations throughout the transient, with zero uncertainty for y and non-zero uncertainty for u at the end of the transient. Also, for the prediction starting at a steady state in Figure 4b, both u and y are predicted to be certain throughout the transient. Both of these results are qualitatively correct.

Prediction Model C.3 uses deviation variables, so when the deviation variables are zero (at the virtual steady-state), the calculated uncertainty as a multiple of the variables becomes zero. Also, since the deviation is from a "virtual" steady state, they are non-zero when responding to a non-stationary disturbance; then, the predicted uncertainty is non-zero. Therefore, we conclude that Prediction Model C.3 is better than Prediction Models C.1 and C.2, and we have used Prediction Model C.3 for the robust MPC controller of the time-invariant uncertain closed-loop systems in this research.

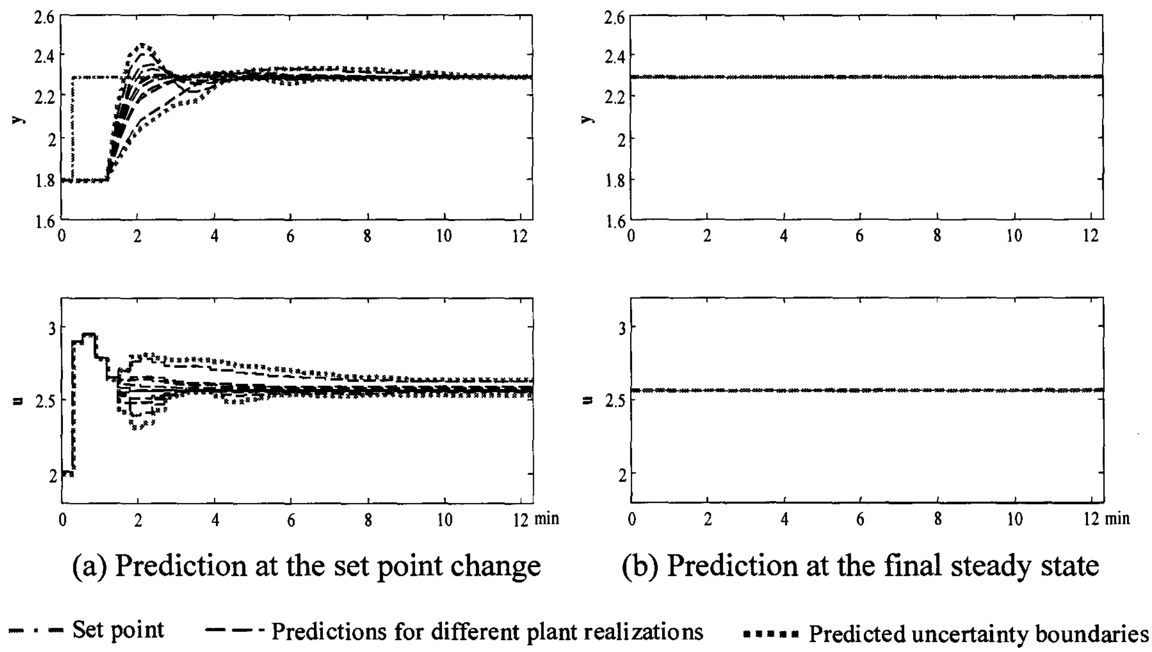


Figure C.4 Uncertainty in prediction models using Prediction Model C.3

C.2 Choosing Virtual Steady States With Time Delays

Note that we pair the current feedback $\hat{e}_0$ with the "most current" manipulated variables u_{-1} for the virtual steady state, because $\hat{e}_0$ reflects the mismatch due to u_{-1}. However, if there is time delay between the controlled and manipulated variables, $\hat{e}_0$ will not includes the effects of u_{-1}, but the effects of u in an earlier time step. We have chosen to select the value of u for the virtual steady state according to the time delays. For a particular manipulated variable, we choose its value $\theta_{\max}$ time steps before the current time, where $\theta_{\max}$ denotes the maximum time delay between this manipulated variable and different controlled variables. This is because the effects of the manipulated variable at that time on all the controlled variables are included in the current feedback. We can also choose the value of a measured disturbance in the similar way if there are time delays between the controlled variables and the measured disturbances.

Appendix D

Derivation of the Closed-loop Model (3.14b-3.14c) in Formulation RMPC-CLT

The closed-loop model (3.14b-3.14c) is derived from the three parts of the closed-loop model in formulation RMPC-CLT.

<u>(1) The controller model:</u>

According to Section 3.2.3, the controller model at the kth time step can be approximated by

$$u_{r,k} = I_{\delta,k}(K_x \hat{x}_{r,k} + K_e \hat{e}_{r,k} + K_u u_{r,k-1} + K_d I_{pd,k} \tilde{\mathbf{d}}_m) + t_k \tag{D.1}$$

where $I_{\delta,k}$ is the diagonal matrix with the diagonal vector δ_k specifying the active bounds on manipulated variables at the kth time step, $\tilde{\mathbf{d}}_m = \left(d_{m,0}^T, \cdots, d_{m,n+p-2}^T\right)^T$ and matrix $I_{pd,k}$ picks the measured or predicted disturbances used by the controller at the *kth* time step from $\tilde{\mathbf{d}}_m$, i.e., $I_{pd,k}\tilde{\mathbf{d}}_m = \left(d_{m,k}^T, \cdots, d_{m,k+p-1}^T\right)^T$. The control law (D.1) can be simplified as

$$u_{r,k} = K_{x,k} \hat{x}_{r,k} + K_{e,k} \hat{e}_{r,k} + K_{u,k} u_{r,k-1} + K_{d,k} \tilde{\mathbf{d}}_m + K_{t,k} t_k \tag{D.2}$$

(2) The uncertain process model:

The model at the *k*th time step is same as Equation (3.5c-3.5f):

$$x_{r,k+1} = A_{r,k+1}x_{r,k} + B_{r,k+1}u_{r,k} + B_{er,k+1}e_{r,k} + B_{dr,k+1}d_{m,k} + G_{wx}w_k \tag{D.3}$$

$$e_{r,k+1} = e_{r,k} + G_{we}w_k \tag{D.4}$$

$$y_{r,k+1} = C_{r,k+1}x_{r,k+1} \tag{D.5}$$

$$y_{r,m,k+1} = C_{r,k+1}x_{r,k+1} + v_{k+1} \tag{D.6}$$

Define the pick-up matrix $I_{d,k}$ such that $d_{m,k} = I_{d,k}\widetilde{\mathbf{d}}_m$, then Equation (D.3) becomes

$$x_{r,k+1} = A_{r,k+1}x_{r,k} + B_{r,k+1}u_{r,k} + B_{er,k+1}e_{r,k} + B_{dr,k+1}I_{d,k}\widetilde{\mathbf{d}}_m + G_{wx}w_k \tag{D.7}$$

(3) The state-estimation:

According to the discussion in Appendix A, the state-estimation at the (*k*+1)th time step is

$$\hat{x}_{r,k+1} = \hat{x}_{r,k+1/k} + L_x(y_{r,m,k+1} - C\hat{x}_{r,k+1/k}) \tag{D.8}$$

$$\hat{e}_{r,k+1} = \hat{e}_{r,k+1/k} + L_e(y_{r,m,k+1} - C\hat{x}_{r,k+1/k}) \tag{D.9}$$

Also we have

$$\begin{aligned}\hat{x}_{r,k+1/k} &= A\hat{x}_{r,k} + Bu_{r,k} + B_e\hat{e}_{r,k} + B_d d_{m,k} \\ &= A\hat{x}_{r,k} + Bu_{r,k} + B_e\hat{e}_{r,k} + B_d I_{d,k}\tilde{\mathbf{d}}_m\end{aligned} \tag{D.10}$$

$$\hat{e}_{r,k+1,k} = \hat{e}_{r,k} \tag{D.11}$$

Now define the extended vector $\xi_k = \left(u_{r,k-1}{}^T, x_{r,k}{}^T, e_{r,k}{}^T, \hat{x}_{r,k}{}^T, \hat{e}_{r,k}{}^T\right)^T$ and $\omega_k = \left(w_k{}^T, v_{k+1}{}^T\right)^T$, then we can summarize Equations (D.2) and (D.4-D.11) as

$$\xi_{k+1} = G_{\xi,k}\xi_k + G_{t,k}t_k + G_{d,k}\mathbf{d}_{m,k} + G_{\omega,k}\omega_k \ , \qquad k = 0,\cdots,n-1 \tag{D.12}$$

where

$$G_{\xi,k} = \begin{bmatrix} K_{u,k} & 0 & 0 & K_{x,k} & K_{e,k} \\ B_{r,k+1}K_{u,k} & A_{r,k+1} & B_{er,k+1} & B_{r,k+1}K_{x,k} & B_{r,k+1}K_{e,k} \\ 0 & 0 & I & 0 & 0 \\ \begin{matrix}(B-L_xCB)K_{u,k} \\ +L_xC_{r,k+1}B_{r,k+1}K_{u,k}\end{matrix} & L_xC_{r,k+1}A_{r,k+1} & L_xC_{r,k+1}B_{er,k+1} & \begin{matrix}A-L_xCA+(B-L_xCB)K_{x,k} \\ +L_xC_{r,k+1}B_{r,k+1}K_{x,k}\end{matrix} & \begin{matrix}(B-L_xCB)K_{e,k} \\ +B_e-L_xCB_e \\ +L_xC_{r,k+1}B_{r,k+1}K_{e,k}\end{matrix} \\ \begin{matrix}L_eC_{r,k+1}B_{r,k+1}K_{u,k} \\ -L_eCBK_{u,k}\end{matrix} & L_eC_{r,k+1}A_{r,k+1} & L_eC_{r,k+1}B_{er,k+1} & \begin{matrix}-L_eCA-L_eCBK_{x,k} \\ +L_eC_{r,k+1}B_{r,k+1}K_{x,k}\end{matrix} & \begin{matrix}I-L_eCBK_{e,k} \\ -L_eCB_e \\ +L_eC_{r,k+1}B_{r,k+1}K_{e,k}\end{matrix} \end{bmatrix}$$

$$G_{t,k} = \begin{pmatrix} K_{t,k} \\ B_{r,k+1}K_{t,k} \\ 0 \\ (B-L_xCB)K_{t,k} \\ +L_xC_{r,k+1}B_{r,k+1}K_{t,k} \\ L_eC_{r,k+1}B_{r,k+1}K_{t,k} \\ -L_eCBK_{t,k} \end{pmatrix}, \quad G_{d,k} = \begin{pmatrix} K_{d,k} \\ B_{dr,k+1}I_{d,k} + B_{r,k+1}K_{d,k} \\ 0 \\ (B_d - L_xCB_d)I_{d,k} + (B-L_xCB)K_{d,k} \\ +L_xC_{r,k+1}(B_{dr,k+1}I_{d,k} + B_{r,k+1}K_{d,k}) \\ -L_eCB_dI_{d,k} - L_eCBK_{d,k} \\ +L_eC_{r,k+1}(B_{dr,k+1}I_{d,k} + B_{r,k+1}K_{d,k}) \end{pmatrix},$$

$$G_{\omega,k} = \begin{pmatrix} 0 & 0 \\ G_{wx} & 0 \\ G_{we} & 0 \\ L_xC_{r,k+1}G_{wx} & L_x \\ L_eC_{r,k+1}G_{wx} & L_e \end{pmatrix}.$$

The recursive formula (D.12) can be rewritten into the following explicit formulation:

$$\begin{aligned}\xi_{k+1} &= (\prod_{i=0}^{k} G_{\xi,k-i})\xi_0 + \sum_{i=0}^{k}\left((\prod_{j=0}^{k-1-i} G_{\xi,k-j})(G_{t,i}t_i + G_{d,i}\tilde{\mathbf{d}}_m + G_{\omega,i}\omega_i) \right) \\ &= G_{\xi t,k}\begin{pmatrix} t_0 \\ \vdots \\ t_k \end{pmatrix} + G_{\xi\xi,k}\xi_0 + G_{\xi d,k}\tilde{\mathbf{d}}_m + G_{\xi\omega,k}\begin{pmatrix} \omega_0 \\ \vdots \\ \omega_k \end{pmatrix}\end{aligned} \tag{D.13}$$

$$k = 0,\ldots,n-1$$

Note that robust MPC prediction is based on that the initial estimation is correct, i.e. $x_{r,0} = \hat{x}_0$ and $e_{r,0} = \hat{e}_0$, therefore the formulation (D.13) can be written in the following form,

$$u_{r,k} = O_{u1,k}\begin{pmatrix} t_0 \\ \vdots \\ t_{n-1} \end{pmatrix} + O_{u2,k}\begin{pmatrix} u_{-1} \\ \tilde{\mathbf{d}}_m \\ \hat{x}_0 \\ \hat{e}_0 \end{pmatrix} + O_{u3,k}\begin{pmatrix} \omega_0 \\ \vdots \\ \omega_k \end{pmatrix} \qquad k = 0,\ldots,n-1 \tag{D.14}$$

,

$$x_{r,k+1} = O_{x1,k}\begin{pmatrix} t_0 \\ \vdots \\ t_{n-1} \end{pmatrix} + O_{x2,k}\begin{pmatrix} u_{-1} \\ \tilde{\mathbf{d}}_m \\ \hat{x}_0 \\ \hat{e}_0 \end{pmatrix} + O_{x3,k}\begin{pmatrix} \omega_0 \\ \vdots \\ \omega_k \end{pmatrix} \qquad k = 0,\ldots,n-1 \tag{D.15}$$

,

At the *n*th time step and thereafter, the manipulated variables will not change, i.e., $u_j = u_{n-1}$ for $j = n,\ldots p$. Therefore,

$$\begin{pmatrix} x_{r,n+1} \\ x_{r,n+2} \\ \vdots \\ x_{r,p} \end{pmatrix} = \begin{pmatrix} A_{r,n} \\ A_{r,n+1}A_{r,n} \\ \vdots \\ \prod_{i=1}^{k} A_{r,p-i} \end{pmatrix} x_{r,n} + \begin{pmatrix} B_{r,n} \\ A_{r,n+1}B_{r,n} + B_{r,n+1} \\ \vdots \\ \sum_{i=0}^{p-n-1} ((\prod_{j=1}^{p-n-i-1} A_{r,p-j})B_{r,n+i}) \end{pmatrix} u_{r,n-1}$$

$$+ \begin{pmatrix} B_{er,n} & \cdots & \cdots \\ \vdots & \vdots & \vdots \\ (\prod_{j=1}^{p-n-1} A_{r,p-j})B_{er,n} & \cdots & B_{er,p-1} \end{pmatrix} \left(\begin{pmatrix} \hat{e}_0 \\ \hat{e}_0 \\ \vdots \\ \hat{e}_0 \end{pmatrix} + \begin{pmatrix} G_{we}\sum_{j=0}^{n-1} w_j \\ G_{we}\sum_{j=0}^{n-2} w_j \\ \vdots \\ G_{we}\sum_{j=0}^{p-2} w_j \end{pmatrix} \right) \tag{D.16}$$

$$+ \begin{pmatrix} B_{dr,n} & \cdots & \cdots \\ \vdots & \vdots & \vdots \\ (\prod_{j=1}^{p-n-1} A_{r,p-j})B_{dr,n} & \cdots & B_{dr,p-1} \end{pmatrix} \begin{pmatrix} d_{m,n} \\ d_{m,n+1} \\ \vdots \\ d_{m,p-1} \end{pmatrix} + \begin{pmatrix} G_{wx}w_n \\ G_{wx}w_{n+1} \\ \vdots \\ G_{wx}w_{p-1} \end{pmatrix}$$

$$= A^* x_{r,n} + B^* u_{r,n-1} + B_e^* \hat{e}_0 + B_d^* I_{d2} \widetilde{\mathbf{d}}_m + G_\omega \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_{p-1} \end{pmatrix}$$

where I_{d2} is the matrix such that $I_{d2}\widetilde{\mathbf{d}}_m = \left(d_{m,n}^T, \cdots, d_{m,p-1}^T\right)^T$. According to Equations (D.14-D.15), the above formulation (D.16) can be written into,

$$\begin{pmatrix} x_{r,n+1} \\ x_{r,n+2} \\ \vdots \\ x_{r,p} \end{pmatrix} = (A^* O_{x1,n-1} + B^* O_{u1,n-1}) \begin{pmatrix} t_0 \\ \vdots \\ t_{n-1} \end{pmatrix} + (A^* O_{x2,n-1} + B^* O_{u2,n-1}) \begin{pmatrix} u_{-1} \\ \widetilde{\mathbf{d}}_m \\ \hat{x}_0 \\ \hat{e}_0 \end{pmatrix}$$

$$+ B_e^* \hat{e}_0 + B_d^* I_{d2} \widetilde{\mathbf{d}}_m + (A^* O_{x3,n-1} + B^* O_{u3,n-1} + G_w) \begin{pmatrix} \omega_0 \\ \vdots \\ \omega_{p-1} \end{pmatrix} \tag{D.17}$$

The above Equation (D.17) and Equations (D.14-D.15) can be summarized as

$$\mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_{ur}\boldsymbol{\omega} \tag{D.16}$$

$$\mathbf{x}_r = L_{xr}\mathbf{t} + M_{xr}\boldsymbol{\theta} + N_{xr}\boldsymbol{\omega} \tag{D.17}$$

where $\mathbf{u}_r = \begin{pmatrix} u_{r,0} \\ \vdots \\ u_{r,n-1} \end{pmatrix}$, $\mathbf{x}_r = \begin{pmatrix} x_{r,1} \\ \vdots \\ x_{r,p} \end{pmatrix}$, $\mathbf{t} = \begin{pmatrix} t_0 \\ \vdots \\ t_{n-1} \end{pmatrix}$, $\boldsymbol{\pi} = \begin{pmatrix} u_{-1} \\ \tilde{\mathbf{d}}_m \\ \hat{x}_0 \\ \hat{e}_0 \end{pmatrix}$, $\boldsymbol{\omega} = \begin{pmatrix} \omega_0 \\ \vdots \\ \omega_{p-1} \end{pmatrix}$, and since $\mathbf{y}_r = \tilde{C}_r\mathbf{x}_r$, then

$$\mathbf{y}_r = L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_{yr}\boldsymbol{\omega} \tag{D.18}$$

The closed-loop model parameters L_{yr}, M_{yr}, N_{yr}, L_{ur}, M_{ur}, N_{ur} are different for different realizations of the process and their corresponding nominal values are denoted by L_y, M_y, N_y, L_u, M_u, N_u. Now define

$$N_{yr} = N_r + \Delta N_{yr}$$

where N_r is nominal value of N_{yr}, then

$$N_{yr}\boldsymbol{\omega} = (N_y + \Delta N_{yr})\boldsymbol{\omega} = N_y\boldsymbol{\omega} + \Delta N_{yr}\boldsymbol{\omega} \approx N_y\boldsymbol{\omega}$$

Similarly, we have

$$N_{ur}\boldsymbol{\omega} \approx N_u\boldsymbol{\omega}$$

so from the Equations (D.16) and (D.18) we obtain the closed-loop model

$$\mathbf{u}_r = L_{ur}\mathbf{t} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \tag{D.19}$$

$$\mathbf{y}_r = L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_y\boldsymbol{\omega} \tag{D.20}$$

Appendix E

Closed-Loop Model Update Rules and Computational Complexity of the Efficient Uncertainty Characterization Method

E.1 The Closed-Loop Model Update Rules

Let's see how to update the closed-loop model of the manipulated variables when the *i*th manipulated input at time step *j*, $u_{r,i,j}$, becomes saturated. Fist we write out more details of the closed-loop model shown in Section 3.3.2 as

$$\begin{pmatrix} u_{r,0} \\ \vdots \\ u_{r,n-1} \end{pmatrix} = L_{ur} \begin{pmatrix} t_0 \\ \vdots \\ t_{n-1} \end{pmatrix} + M_{ur}\boldsymbol{\theta} + N_u\boldsymbol{\omega} \tag{E.1}$$

where $u_{r,k} = (u_{r,1,k},\ldots,u_{r,n_u,k})^T \in R^{n_u}$ denotes the manipulated variables at the future *k*th time step, $t_k = (t_{1,k},\ldots,t_{n_u,k})^T \in R^{n_u}$ denotes the degrees of freedom for the control law at the future *k*th time step, n_u denotes the number of manipulated variables. Also we can use n_θ, n_ω to denote the numbers of elements in $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ respectively. The coefficient matrices L_{ur}, M_{ur}, N_u are composed of sub-matrices defined as follows:

$$L_{ur} = \begin{bmatrix} L_{ur}^{(0,0)} & & & & \\ L_{ur}^{(1,0)} & L_{ur}^{(1,1)} & & & \\ L_{ur}^{(2,0)} & L_{ur}^{(2,1)} & L_{ur}^{(2,2)} & & \\ \vdots & & & \ddots & \\ L_{ur}^{(m-1,0)} & L_{ur}^{(n-1,1)} & \cdots & L_{ur}^{(n-1,n-2)} & L_{ur}^{(n-1,n-1)} \end{bmatrix} \in R^{(n\cdot n_u)\times(n\cdot n_u)}$$, where the sub-matrices

$$L_{ur}^{(k_1,k_2)} = \begin{pmatrix} L_{ur,1}^{(k_1,k_2)} \\ \vdots \\ L_{ur,n_u}^{(k_1,k_2)} \end{pmatrix} \in R^{n_u\times n_u}$$ and again the sub-matrices

$L_{ur,\ o_1}^{(k_1,k_2)} = \left(L_{ur,o_1,1}^{(k_1,k_2)},\dots,L_{ur,o_1,n_u}^{(k_1,k_2)}\right) \in R^{1\times n_u}$. The superscripts $k_1 = 0,\dots,n-1$, $k_2 = 0,\dots,k_1$ and the subscripts $o_1 = 1,\dots,n_u$. Note that when $k_1 = k_2$, $L_{ur}^{(k_1,k_2)}$ is an identity matrix.

Also,

$$M_{ur} = \begin{pmatrix} M_{ur}^{(0)} \\ \vdots \\ M_{ur}^{(n-1)} \end{pmatrix} \in R^{(n\cdot n_u)\times n_\theta} \text{ and } M_{ur}^{(k_1)} = \begin{pmatrix} M_{ur,1}^{(k_1)} \\ \vdots \\ M_{ur,n_u}^{(k_1)} \end{pmatrix} \in R^{n_u\times n_\theta},\ N_{ur} = \begin{pmatrix} N_{ur}^{(0)} \\ \vdots \\ N_{ur}^{(n-1)} \end{pmatrix} \in R^{(n\cdot n_u)\times n_\omega} \text{ and}$$

$$N_{ur}^{(k_1)} = \begin{pmatrix} N_{ur,1}^{(k_1)} \\ \vdots \\ N_{ur,n_u}^{(k_1)} \end{pmatrix} \in R^{n_u\times n_\omega}, \text{ where, } k_1 = 0,\dots,n-1.$$

According to the closed-loop (E.1), when $u_{r,j,i}$ does not saturate, its model can be expanded as

$$\begin{aligned} u_{r,i,j} &= \left(L_{ur,i}^{(j,0)},\ \cdots,\ L_{ur,i}^{(j,j-1)}\right)\begin{pmatrix} t_0 \\ \vdots \\ t_{j-1} \end{pmatrix} + t_{i,j} + M_{ur,i}^{(j)}\boldsymbol{\theta} + N_{u,i}^{(j)}\boldsymbol{\omega} \\ &= t_{i,j} + \Delta u_{r,i,j} \end{aligned} \tag{E.2}$$

And when $u_{r,i,j}$ saturates, it will be forced to be the corresponding degrees of the freedom of the controller (according to the heuristic discussed in Section 3.2.4), $t_{i,j}$, i.e.,

$$u_{r,i,j} = t_{i,j}, \qquad (t_{i,j} \text{ will be forced to a bound in optimization}) \tag{E.3}$$

Note that the difference between the $u_{r,i,j}$ values with and without saturation is denoted by $\Delta u_{r,i,j}$ according to the above equations (E.2) and (E.3). With the presence of the $u_{r,i,j}$ saturation, the models for the manipulated variables after the *j*th time step will change; We assume this change can be quantified as:

$$u_{r,o_1,k_1}{}^* = u_{r,o_1,k_1} - C^{(k_1,j)}_{ur,o_1,i} \Delta u_{r,i,j}, \qquad (k_1 = j+1,\ldots,n-1,\ o_1 = 1,\ldots,n_u) \tag{E.4}$$

where u_{r,o_1,k_1} denotes the value of the manipulated variable with the closed-loop model before model update and $u_{r,o_1,k_1}{}^*$ denotes the value after model update, $C^{(k_1,j)}_{ur,o_1,i}$ denotes the closed-loop effect of $u_{r,i,j}$ on u_{r,o_1,k_1}. According to equation (E.4), the coefficients of the closed-loop model of u_{r,o_1,k_1} can be updated as,

$$L^{(k_1,k_2)}_{ur,o_1,o_2}{}^* = L^{(k_1,k_2)}_{ur,o_1,o_2} - C^{(k_1,j)}_{ur,o_1,i} L^{(j,k_2)}_{ur,i,o_1} \tag{E.5}$$

$$\left(M^{(k_1)}_{ur,o_1},\quad N^{(k_1)}_{u,o_1}\right)^* = \left(M^{(k_1)}_{ur,o_1},\quad N^{(k_1)}_{u,o_1}\right) - C^{(k_1,j)}_{ur,o_1,i}\left(M^{(j)}_{ur,i},\quad N^{(j)}_{u,i}\right) \tag{E.6}$$

$$k_1 = j+1,\ldots,n-1,\ k_2 = 0,\ldots,j-1,\ o_1 = 1,\ldots,n_u,\ o_2 = 1,\ldots,n_u$$

Now the question is:

What is $C^{(k_1,j)}_{ur,o_1,i}$?

Next, we will prove

$C^{(k_1,j)}_{ur,o_1,i}$, *the closed-loop effect of* $u_{r,i,j}$ *on* u_{r,o_1,k_1}, *is just* $L^{(k_1,j)}_{ur,o_1,i}$.

According to equation (E.1),

$$u_{r,o_1,k_1} = \left(L_{ur,o_1}^{(k_1,0)}, \quad \cdots, \quad L_{ur,o_1}^{(k_1,k_1)}\right)\begin{pmatrix} t_0 \\ \vdots \\ t_{k_1} \end{pmatrix} + M_{ur,o_1}^{(k_1)}\boldsymbol{\theta} + N_{u,o_1}^{(k_1)}\boldsymbol{\omega} \tag{E.7}$$

We can also write the closed-loop model of u_{r,o_1,k_1} as if the (j+1)th time step is the initial time step,

$$u_{r,k_1,o_1} = L_{ur,o_1}^{(k_1)}\,|_{j+1} \cdot \begin{pmatrix} t_{j+1} \\ \vdots \\ t_{k_1} \end{pmatrix} + M_{ur,o_1}^{(k_1)}\,|_{j+1} \cdot \boldsymbol{\theta}\,|_{j+1} + N_{u,o_1}^{(k_1)}\,|_{j+1} \cdot \boldsymbol{\omega} \tag{E.8}$$

where $|_{j+1}$ means that the value of the parameter or variable is based on taking the ($j+1$)th time step as the initial time step. Since $\boldsymbol{\theta}\,|_{j+1}$ is the linear function of $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, t_0, ..., t_{j-1} and $u_{r,j}$ (note the input vector $u_{r,j} = \left(u_{r,1,j}, \cdots, u_{r,n_u,j}\right)^T \in R^{n_u}$), the above equation (E.8) can be written as

$$u_{r,k_1,o_1} = L_1^*\begin{pmatrix} t_0 \\ \vdots \\ t_{j-1} \end{pmatrix} + L_2^*\begin{pmatrix} t_{j+1} \\ \vdots \\ t_{k_1} \end{pmatrix} + Cu_{r,j} + M^*\boldsymbol{\theta} + N^*\boldsymbol{\omega} \tag{E.9}$$

where C denotes the closed-loop effect of $u_{r,j}$ on u_{r,o_1,k_1}. As we know, each element in $u_{r,j}$ is linear function of $\boldsymbol{\theta}$, $\boldsymbol{\omega}$, t_0, ..., t_j, so equation (E.9) can also be written as

$$u_{r,o_1,k_1} = L_1^{**}\begin{pmatrix} t_0 \\ \vdots \\ t_{j-1} \end{pmatrix} + L_2^{**}\begin{pmatrix} t_{j+1} \\ \vdots \\ t_{k_1} \end{pmatrix} + Ct_j + M^{**}\boldsymbol{\theta} + N^{**}\boldsymbol{\omega} \tag{E.10}$$

Since equations (E.7) and (E.8) are identical for any parameter values, equations (E.7) and (E.10) are identical for any parameter values too. Therefore,

$$C = L_{ur,o_1}^{(k_1,j)}$$

which means the closed-loop effect of $u_{r,j}$ on u_{r,o_1,k_1} is $L_{ur,o_1}^{(k_1,j)}$; so the closed-loop effect of $u_{r,i,j}$ on u_{r,o_1,k_1} is $L_{ur,o_1,i}^{(k_1,j)}$.

According to all the above discussion, the coefficients of the closed-loop model of the manipulated variables after the *j*th time step can be updated as

$$L_{ur,o_1,o_2}^{(k_1,k_2)} * = L_{ur,o_1,o_2}^{(k_1,k_2)} - L_{ur,o_1,i}^{(k_1,j)} L_{ur,i,o_1}^{(j,k_2)} \tag{E.11}$$

$$\left(M_{ur,o_1}^{(k_1)},\quad N_{u,o_1}^{(k_1)}\right)* = \left(M_{ur,o_1}^{(k_1)},\quad N_{u,o_1}^{(k_1)}\right) - L_{ur,o_1,i}^{(k_1,j)}\left(M_{ur,i}^{(j)},\quad N_{u,i}^{(j)}\right) \tag{E.12}$$

$$k_1 = j+1,\ldots,n-1,\quad k_2 = 0,\ldots,j-1,\quad o_1 = 1,\ldots,n_u,\quad o_2 = 1,\ldots,n_u$$

Similarly, the closed-loop model of the controlled variables can be written as

$$\mathbf{y}_r = L_{yr}\mathbf{t} + M_{yr}\boldsymbol{\theta} + N_y\boldsymbol{\omega} \tag{E.13}$$

where

$$L_{yr} = \begin{bmatrix} L_{yr}^{(1,0)} & & & \\ L_{yr}^{(2,0)} & L_{yr}^{(2,1)} & & \\ \vdots & \vdots & \ddots & \\ L_{yr}^{(n,0)} & L_{yr}^{(n,1)} & \cdots & L_{yr}^{(n,n-1)} \\ \vdots & \vdots & \vdots & \vdots \\ L_{yr}^{(p,0)} & L_{yr}^{(p,1)} & \cdots & L_{yr}^{(p,n-1)} \end{bmatrix} \in R^{(p \cdot n_y)\times(n \cdot n_u)}$$, where the sub-matrices

$L_{yr}^{(k_1,k_2)} = \begin{pmatrix} L_{yr,1}^{(k_1,k_2)} \\ \vdots \\ L_{yr,n_y}^{(k_1,k_2)} \end{pmatrix} \in R^{n_y \times n_u}$ and again the sub-matrices $L_{ur,\ o_1}^{(k_1,k_2)} = \left(L_{ur,o_1,1}^{(k_1,k_2)},\ldots,L_{ur,o_1,n_u}^{(k_1,k_2)}\right) \in R^{1\times n_u}$.

The superscripts $k_1 = 1,\ldots,p$, $k_2 = 0,\ldots,\min(k_1 - 1, n-1)$ and the subscripts $o_1 = 1,\ldots,n_y$., n_y denotes the number of the controlled variables. Also,

$M_{ur} = \begin{pmatrix} M_{ur}^{(1)} \\ \vdots \\ M_{ur}^{(p)} \end{pmatrix} \in R^{(p \cdot n_y)\times n_\theta}$ and $M_{ur}^{(k_1)} = \begin{pmatrix} M_{ur,1}^{(k_1)} \\ \vdots \\ M_{ur,n_y}^{(k_1)} \end{pmatrix} \in R^{n_y \times n_\theta}$, $N_{ur} = \begin{pmatrix} N_{ur}^{(1)} \\ \vdots \\ N_{ur}^{(p)} \end{pmatrix} \in R^{(p \cdot n_y)\times n_\varpi}$ and

$N_{ur}^{(k_1)} = \begin{pmatrix} N_{ur,1}^{(k_1)} \\ \vdots \\ N_{ur,n_y}^{(k_1)} \end{pmatrix} \in R^{n_y \times n_\varpi}$, where, $k_1 = 1,\ldots,p$.

If an unsaturated manipulated variable $u_{rj,i}$ becomes saturated, the model coefficients in equation (E.13) can be updated as

$$L_{yr,o_1,o_2}^{(k_1,k_2)}* = L_{yr,o_1,o_2}^{(k_1,k_2)} - L_{yr,o_1,i}^{(k_1,j)} L_{yr,i,o_1}^{(j,k_2)} \tag{E.14}$$

$$\left(M_{yr,o_1}^{(k_1)},\quad N_{y,o_1}^{(k_1)}\right)* = \left(M_{yr,o_1}^{(k_1)},\quad N_{y,o_1}^{(k_1)}\right) - L_{yr,o_1,i}^{(k_1,j)}\left(M_{yr,i}^{(j)},\quad N_{y,i}^{(j)}\right) \tag{E.15}$$

$$k_1 = j+1,\ldots,p,\ k_2 = 0,\ldots,j,\ o_1 = 1,\ldots,n_y,\ o_2 = 1,\ldots,n_u$$

These results are obtained in the similar approach in which the update rules (E.11) and (E.12) are obtained.

E.2 Complexity of On-line Uncertainty Characterization

E.2.1 The time complexity (number of scalar calculation needed)

To update the model coefficients of the manipulated variables by equations (E.11-E.12) for the case that $u_{r,i,j}$ becomes saturated, we need the number of scalar calculations

$$2(n-1-j)\cdot n_u \cdot (n_u \cdot j + n_{MN})$$

where n_{MN} denotes the number of elements of each row of the augmented matrix [M_u, N_u] (or the total number of elements in vectors $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, i.e. $n_{MN} = n_\theta + n_\omega$). To update the model coefficients of the controlled variables by equations (E.14-E.15) for the same case, we need the number of scalar calculations

$$2(p-j+1)\cdot n_y \cdot (n_u \cdot j + n_{MN})$$

So the total calculation associated to coefficient update is

$$2n_s\left(\sum_{[j,i]\in\Omega}(n_u \cdot j + n_{MN})[(n-1-j)\cdot n_u + (p-j+1)\cdot n_y]\right)$$

where n_s denotes the total number samples, Ω denotes the set containing the indices of the additional saturated inputs. In the worst case, we will find all the future inputs should saturate, then the total number of scalar calculation will be

$$
\begin{aligned}
&2n_s\left(\sum_{i=1}^{n_u}\sum_{j=0}^{n-1}(n_u\cdot j+n_{MN})[(n-1-j)\cdot n_u+(p-j+1)\cdot n_y]\right)\\
&=2n_sn_u(\sum_{j=0}^{n-1}(n_{MN}[(n-1)n_u+(p+1)n_y]-n_u(n_u+n_y)j^2\\
&\quad+[(n-1)n_u{}^2+(p+1)n_yn_u-n_{MN}(n_u+n_y)]j))\\
&=2n_sn_u(n\cdot n_{MN}[(n-1)n_u+(p+1)n_y]-n_u(n_u+n_y)\frac{n(n-1)(2n-1)}{6}\\
&\quad+[(n-1)n_u{}^2+(p+1)n_yn_u-n_{MN}(n_u+n_y)]\frac{n(n-1)}{2})\\
&\sim O(n_s(n_un)^3)
\end{aligned}
\tag{E.16}
$$

Here we are assuming the number of manipulated variables times the number of time steps in the control horizon dominates the scale of the system.

E.2.2 The space complexity (number of scalars to be stored in memory)

The online calculation is to update the coefficients of the closed-loop models for different samples, L_{ur}, M_{ur}, N_u, L_{yr}, M_{yr}, N_y. So the number of scalars in memory during the calculation is at least the number of elements in these matrices.

For one sample, the number of elements in L_{ur}, M_{ur}, N_u is at most

$$
\sum_{j=0}^{n-1}n_u^2(j+1)+n_u\cdot n\cdot n_{MN}=\frac{n(n+1)n_u^2}{2}+n_un_{MN}n
$$

and the number of elements in L_{yr}, M_{yr}, N_y is at most

$$
\begin{aligned}
&\sum_{j=0}^{n-1}n_yn_u(j+1)+(p-n)\cdot n_y\cdot n_u\cdot n+n_y\cdot p\cdot n_{MN}\\
&=\frac{n(n+1)n_yn_u}{2}+n(p-n)n_yn_u+n_yn_{MN}p
\end{aligned}
$$

So the total number of scalars is at most

$$n_s\left(\frac{n(n+1)n_u^2}{2}+n_u n_{MN} n+\frac{n(n+1)n_y n_u}{2}+n(p-n)n_y n_u+n_y n_{MN} p\right)$$
$$\sim O(n_s(n_u n)^2) \tag{E.17}$$

Here we are assuming the number of manipulated variables times the number of time steps in the control horizon dominates the scale of the system.

E3 Complexity of Off-line Uncertainty Characterization

E.3.1 The time complexity (number of scalar calculation needed)

The off-line calculation is to obtain the closed-model using the procedure shown in Appendix D. It's not difficult to find that the calculation described by equation (D.13) dominates the total off-line calculation. Let's repeat equation (D.13) here for convenience of discussion:

$$\begin{aligned}\xi_{k+1} &= (\prod_{i=0}^{k} G_{\xi,k-i})\xi_0 + \sum_{i=0}^{k}\left((\prod_{j=0}^{k-1-i} G_{\xi,k-j})(G_{t,i}t_i + G_{d,i}\tilde{\mathbf{d}}_m + G_{\omega,i}\omega_i)\right)\\ &\le \sum_{i=0}^{k}\left((\prod_{j=0}^{k-1-i} G_{\xi,k-j})(G_{t,i}t_i + G_{d,i}\tilde{\mathbf{d}}_m + G_{\omega,i}\omega_i + G_{\xi,k-i}\xi_0)\right)\end{aligned} \tag{E.18}$$

$$k = 0,\ldots,n-1$$

We can see that the matrices $(G_{t,i}, G_{d,i}, G_{\omega,i}, G_{\xi,k-i})$ has the scale of

$$O(n_u+n_y)\times O(n_u+n_{MN})$$

so the number of scalar calculations in equation (E.18) for each time step k is bounded by

$$O((n_u + n_y)^2 (n_u + n_{MN}))\frac{k(k+1)}{2}$$

and the total number of scalar calculations is

$$\sum_{k=0}^{n-1}\left(O((n_u + n_y)^2 (n_u + n_{MN}))\frac{k(k+1)}{2} \right)$$
$$\sim O((n_u n)^3)$$

Since we need to obtain the closed-loop model for each of the realizations, so the total scalar calculations involved in the off-line calculation is at most $\sim O(n_s (n_u n)^3)$.

E.3.2 The space complexity (number of scalars to be stored in memory)

The number of scalars in memory during the off-line calculation is basically the number of elements in the matrices L_{ur}, M_{ur}, N_u, L_{yr}, M_{yr}, N_y in the closed-loop model, so the space complexity of off-line calculation is the same as that of on-line calculation, i.e., $\sim O(n_s (n_u n)^2)$.

Appendix F

Details of the CSTR Processes

We first introduce the general model and the linearization of it for a typical CSTR process, and then give details of the parameters and operating points used in the CSTR control case studies in the thesis.

Figure F.1 illustrates a typical Continuous Stirred Tank Reactor (CSTR) system. We assume the tank is full of well-mixed liquid and physical properties are constant. In the reactor there is an irreversible, elementary reaction A->B, which is first-order with Arrhenius temperature dependence. The heat loss of the system is negligible.

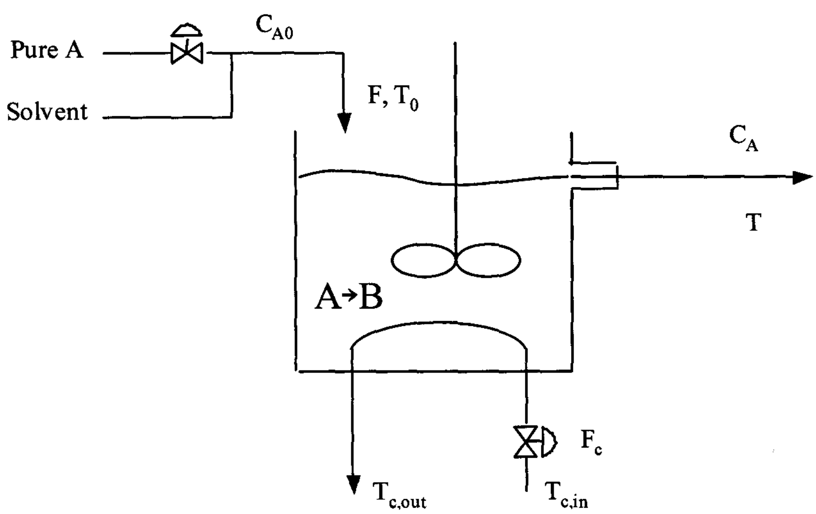


Figure F.1 The diagram of a typical CSTR system

The nonlinear model of the system is composed of the mass balance and the energy balance equations as follows:

Mass balance:

$$Vol\frac{dC_A}{dt} = F(C_{A0} - C_A) - Vol \cdot k_{A0}e^{-E/RT}C_A \tag{F.1}$$

where C_{A0} is the inlet flow concentration of A, C_A is the outlet flow concentration of A. F is the inlet and outlet flow rate, *Vol* is the volume of the reactor, T is the temperature in the reactor, $K_{A0}e^{-E/RT}$ gives the reaction rate K_A that obeys first-order Arrhenius equation.

Energy balance:

$$Vol \cdot \rho C_p \frac{dT}{dt} = \rho C_p F(T_0 - T) - \frac{aF_c^{b+1}}{F_c + \frac{aF_c^b}{2\rho_c C_{pc}}}(T - T_{c,in}) - \Delta H_{rxn} Vol \cdot k_{A0}e^{-E/RT}C_A \tag{F.2}$$

where ρ and C_p are the density and specific heat capacity of the mixture in reactor, ρ_c and C_{pc} is the specific heat capacity of the coolant, *a*, *b* denote the coefficients of the heat transfer during the cooling procedure, T_0 denotes the temperature of the inlet flow, $T_{c,in}$ denotes the temperature of the inlet cooling flow, ΔH_{rxn} denotes the enthalpy of the reaction.

The nonlinear model (F.1)-(F.2) and can be linearized around a steady-state into:

$$\frac{dC_A}{dt} = a_{11}C'_A + a_{12}T' + a_{13}C'_{A0} + a_{14}F'_c \tag{F.3}$$

$$\frac{dC_A}{dt} = a_{21}C'_A + a_{22}T' + a_{23}C'_{A0} + a_{24}F'_c \tag{F.4}$$

where

$$a_{11} = -\frac{F}{Vol} - k_{A0}e^{-E/RT_s}$$

$$a_{12} = -\frac{E}{RT_s^2}k_{A0}e^{-E/RT_s}C_{As}$$

$$a_{13} = \frac{F}{Vol}$$

$$a_{14} = 0$$

$$a_{21} = \frac{-\Delta H_{rxn}k_{A0}e^{-E/RT_s}}{\rho C_p}$$

$$a_{22} = -\frac{F}{Vol} - \frac{a(F_c)_s^{b+1}/[(F_c)_s + a(F_c)_s^b/2\rho_c C_{pc}]}{Vol \cdot \rho C_p} - \Delta H_{rxn}\frac{\frac{E}{RT_s^2}}{\rho C_p}k_{A0}e^{-E/RT_s}C_{As}$$

$$a_{23} = 0$$

$$a_{24} = \frac{-ab(F_{cs})^b\left(F_{cs} + \frac{a}{b}\frac{F_{cs}^b}{2\rho_{pc}C_{pc}}\right)[T_s - (T_{cin})_s]}{Vol \cdot \rho C_p\left(F_{cs} + \frac{aF_{cs}^b}{2\rho_c C_{pc}}\right)^2}$$

and the subscript "s" denotes the steady-state value of the variable, the prime symbol (') denotes the deviation variable that deviates the original variable from its steady-state value.

The process in the CSTR control system 1 and 2 in Chapter 3 and the CSTR control system 3 in Chapter 4 is an exothermic CSTR process (i.e. $\Delta H_{rxn}<0$). This process is discussed and its parameter values used in the thesis are shown on page 897-908 of Marlin (2000). The parameters are shown in Table F-1.

The process in the CSTR control system 4 in Chapter 4 is the CSTR process with zero heat of reaction (i.e. $\Delta H_{rxn}=0$). This process is discussed and its parameter values used in the thesis are shown on page 438-439 of Marlin (2000). The parameters are shown in the following Table F-2.

Table F-1 The parameters of the exothermic CSTR process

Parameter	Value	Unit
F	1	m^3/min
Vol	1	m^3
C_{A0}	2.0	$kmole/m^3$
T_0	343	K
C_p	1	cal/(g•K)
ρ	10^6	g/m^3
k_{A0}	10^{10}	min^{-1}
E/R	8330.1	K
ΔH_{rxn}	-1.3×10^8	cal/kmole
T_{cin}	310	K
C_{pc}	1	cal/(g•K)
ρ_c	10^6	g/m^3
a	1.678×10^6	cal/(min•K)
b	0.5	-
T_s	330.9	K
C_{As}	1.79	$kmole/m^3$
F_{cs}	15	m^3/min

Table F-2 The parameters of the CSTR process with zero heat of reaction

Parameter	Value	Unit
F	0.085	m^3/min
Vol	2.1	m^3
C_{A0}	0.965	kmole/m^3
T_0	423.15	K
C_p	1	cal/(g•K)
ρ	10^6	g/m^3
k_{A0}	5.62×10^7	min^{-1}
E/R	1804.1	K
ΔH_{rxn}	0	cal/kmole
T_{cin}	298.15	K
C_{pc}	1	cal/(g•K)
ρ_c	10^6	g/m^3
a	1.41×10^5	cal/(min•K)
b	0.5	-
T_s	358.55	K
C_{As}	0.465	mole/m^3
F_{cs}	0.5	m^3/min

Appendix G

Model Details of the Binary Distillation Column

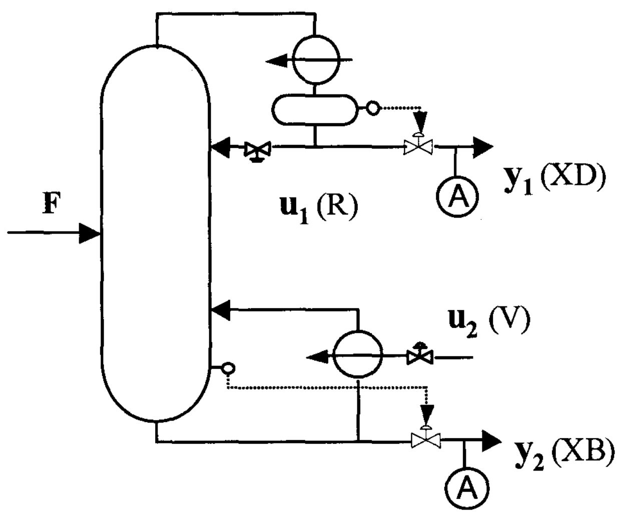


Figure G.1 The binary distillation column

Figure G.1 shows the binary distillation column described in Chapter 4. The model formulation of this distillation column is from Marlin (1995) and parameters from Luyben (1989). The controlled variables are the distillate composition (light key) XD (y_1) and the bottoms composition (light key) XB (y_2). The manipulated variables are the reflux rate (u_1) and boil up rate (u_2). The designed parameter and the initial condition is described in the following Table G-1.

The linearized model around the initial condition shown in Table G-1 can be obtained through step change test and the statistical model identification method introduced in Marlin (2000). For the linearized models used in Chapter 4, the step changes of R0, V0 and F0 of 10 mole/min respectively are performed for the step change test.

Table G-1 The parameters of the binary distillation column in Figure G-1

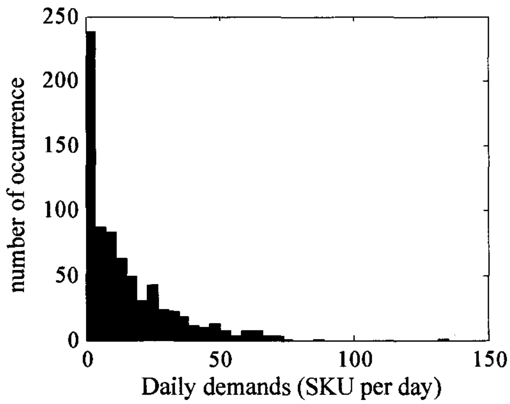
Parameter	Value	Unit
Relative volatility (α)	2.0	-
Number of trays (NT)	25	-
Feed tray location (NF)	12	-
Analyzer dead time (AT)	10	min
Feed rate (F0)	8.7713	kmole/min
Feed light key (z0)	0.5	mole frac
Feed liquid frac (q0)	1	mole frac
Reflux rate (R0)	8.47511	kmole/min
Boilup rate (V0)	13.0022	kmole//min
Distillate rate (D0)	4.52712	kmole//min
Bottoms rate (B0)	4.24418	kmole//min
Distillate light key (XD)	0.95	mole frac
Bottoms light key (XB)	0.02	mole frac
Distillate drum hold up (MD0)	87.713	kmole
Column base hold up (MB0)	87.713	kmole

Appendix H

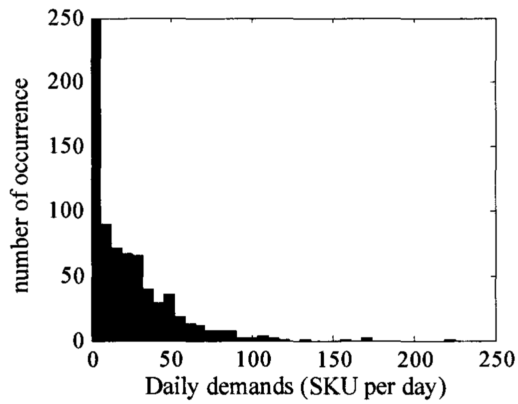
Details of the Industrial Supply Chain System in the Case Studies in Chapter 4

Figure H.1 (a-d) show the histograms of the daily customer demand of the 1^{st} or 2^{nd} SKU to the 1^{st} or 2^{nd} RDC in the case studies of the industrial supply chain system, which are obtained from the industrial historical data of the year 2004 and 2005.

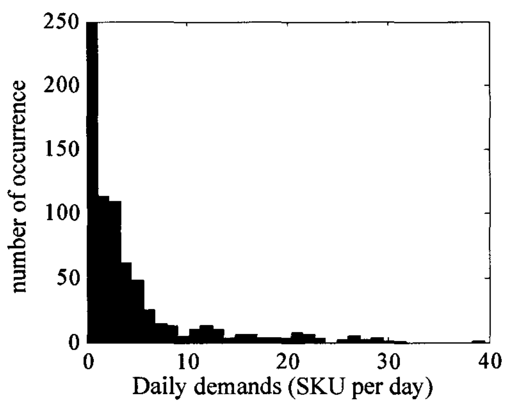
Table H-1 summarizes the parameters of the industrial supply chain system with the 1^{st} and the 2^{nd} IP/SKU and the 1^{st} and the 2^{nd} RDC.



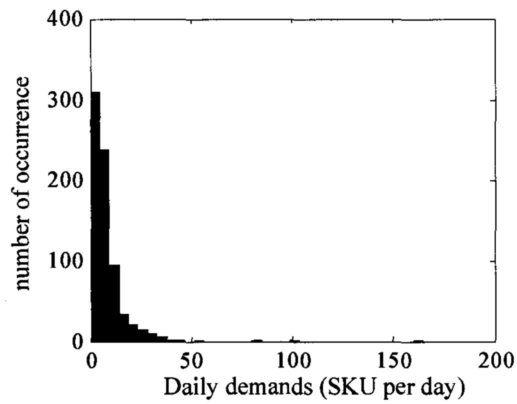
(a) The 1st SKU at the 1st RDC



(b) The 1st SKU at the 2nd RDC



(c) The 2nd SKU at the 1st RDC



(d) The 2nd SKU at the 2nd RDC

Figure H.1 Histograms of the daily demands in the industrial supply chain system

Table H-1 Parameters of the industrial supply chain system

Parameter	Value
Nominal SKU manufacturing rate R_s (SKU/hour)	16.7 (the 1st IP/SKU) 16.7 (the 2nd IP/SKU)
R_s range with 90% confidence (SKU/ hour)	13.3-22.2 (the 1st IP/SKU) 13.3-22.2 (the 2nd IP/SKU)
Unit converting coefficients $C_{IP\text{-}SKU,i}$ (IP/ SKU)	6.0 (the 1st IP/SKU) 6.6 (the 2nd IP/SKU)
Nominal SKU transportation time τ_j (hour)	144 (the 1st RDC) 144 (the 2nd RDC)
τ_j range with 90% confidence (hour)	132-156 (the 1st RDC) 132-156 (the 2nd RDC)
SKU Shipping Intervals (hour)	12 (the 1st RDC) 8 (the 2nd RDC)
SKU transportation capacity $F_{4,\max,j}$ (SKI/shipping interval)	40 (the 1st IP/SKU) 40 (the 2nd IP/SKU)