Robust Model Predictive Control

by

Yiyang Wang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2002

To my parents, for their love, support and endless encouragement.

Acknowledgment

First of all, I want to thank my advisor Professor James B. Rawlings for all the advice he has dispensed over the years. During my stay in Madison, Wisconsin, there were many times when I was frustrated with my lack of progress and was able to find new directions after productive discussions with Jim. Thank you Jim for all your guidance and support.

I want to thank my undergraduate advisor Dr. Thomas A. Badgwell for introducing me to Model Predictive Control and for recommending University of Wisconsin - Madison. Thank you Tom for your continued guidance.

I would like to thank all of the past and present members of the Rawlings Group: John Eaton for maintaining our computing system and answering all of my Latex questions; Rock Matthews, John Campbell, Rolf and Peter Findeisen, Chris Rao, Rahul Bindlish, Scott Middlebrooks, and Daniel Patience for showing me the ropes when I first joined the group; Matthew Tenny for answering all of my Linux questions, for showing me how to use NPSOL, and for proofreading my thesis; Gabriele Pannocchia for the many interesting discussions we had about robust MPC; Brian Odelson for the computer support he provided; Eric Haseltine for the interesting discussions we had about Madison, Kingsport, TN and graduate school in general; Mary Diaz for all the treats she brought into the office and for taking care of the administrative responsibilities.

Last but not least, I want to thank my parents Fusheng Wang and Lily Yang. I would not be where I am today without their constant support and love. They taught me to believe in myself and reach for the highest goals possible. As long as I am trying my best, nothing is impossible. Their life story is a constant inspiration to me.

YIYANG WANG

University of Wisconsin–Madison

January 2002

Robust Model Predictive Control

Yiyang Wang

Under the supervision of Professor James B. Rawlings

At the University of Wisconsin–Madison

The success of the single-model model predictive control (SMPC) method depends on the accuracy of the process model. Modeling errors cause sub-optimal control performance and may cause the system to become closed-loop unstable. The goal of this research project is to design a new robust model predictive control (RMPC) method that guarantees closed-loop stability and offset-free set point tracking in the presence of model uncertainty.

The first RMPC method I considered is a SMPC controller with zone regions instead of set points as the control objective for the controlled outputs. The addition of the zone regions relaxes the control objectives so that control actions are taken only when the controlled outputs are outside the zone limits. Even though this method apprears attractive in theory, it is unsuccessful in practice. The method remains closed-loop stable in the presence of model uncertainty if

the controlled outputs remain inside the zone limits. As soon as control actions are needed to move the controlled outputs into the zone region, SMPC with zones behave the same as SMPC with set points. In other words, if SMPC with set points is closed-loop unstable when model uncertainty is present, then SMPC with zone regions is closed-loop unstable as well.

Next, I proposed a new RMPC method that explicitly accounts for model uncertainty in the controller design procedure. A *min-max* optimization problem is used to determine the optimal control action subject to input and output constraints. the robust regulator uses a tree trajectory to forecast the time-varying model uncertainty. The controller design procedure uses integrators to reject non-zero disturbances and maintain the system at the set points. I developed a closed-loop stability condition that determines if there exist a feedback law $u = Kx$ and disturbance filters L_i's that satisfy the stability condition, then the new RMPC method can achieve offset-free non-zero set point tracking for a constrained system with time-varying model uncertainty described by an uncertainty set.

Table 7.1 summarizes the examples studies in this research project. The examples include model uncertainty sets with:

- process dynamic uncertainty,
- process gain uncertainty,
- stable and unstable models,

- time-delay uncertainty,
- ill-conditioned models,
- integrating models,

and systems with constraint saturation. The new RMPC method successfully controlled all of the above examples for offset-free non-zero set point tracking. The new RMPC controller increases the tolerance of the controller to model uncertainty, but the decrease in nominal performance is relatively small. Indeed, in most cases, the RMPC control performance is almost as good as the nominal SMPC performance.

Contents

Acknowledgment **ii**

Abstract **iv**

List of Tables **xi**

List of Figures **xii**

Chapter 1 Introduction **1**

Chapter 2 Model Predictive Control **9**

2.1 Introduction 9

2.2 Problem Statement 11

2.3 Infinite Horizon Regulator Stability 12

2.4 State Estimation 15

2.5 Output Feedback 17

2.6 Target Calculation 21

2.7 Closed-loop Stability 24

Chapter 3 Single-Model MPC with Zone Regions **30**

3.1 Introduction 30

3.2 Motivation 31

3.3 Zone Definition 32

3.4 Target Calculation 35

3.5 Regulator 38

3.6 State Estimator 41

3.7 Plant-Model Mismatch 42

3.8 Ill-Conditioned Distillation Column 46

3.9 Conclusion 61

3.10 Appendix 62

Chapter 4 Robust Control Literature Review **65**

4.1 Introduction 65

4.2 Uncertainty Description 66

4.3 H_∞ Control 70

4.4 Robust Predictive Control 73

4.5 Integral Control 78

4.6 Summary 84

Chapter 5 Robust Model Predictive Control **86**

5.1 Problem Statement 87

5.2 Definitions and Notations 89
5.2.1 System Theory 89
5.2.2 The New Robust Regulator 90
5.3 The New Robust Regulator Design 92
5.4 The Open-Loop Control Problem 96
5.5 Existence of K and F 101
5.6 Output Feedback 103
5.7 Target Calculation 105
5.8 Closed-loop Stability 110
5.9 Existence of $\tilde{L}$ 116
5.10 Summary of the RMPC Method 120
5.11 Conclusion 121
5.12 Appendix 122
5.12.1 Additional Definitions Needed for Theorem 5.4.1 and 5.4.2 Proofs 122
5.12.2 Proof for Theorem 5.4.1 123
5.12.3 Proof for Theorem 5.4.2 128
5.12.4 State at $t+1$ as Function of State at t 132

Chapter 6 Results and Discussion **133**

6.1 Introduction 133
6.2 Single-Input-Single-Output Examples 135

6.2.1 Unconstrained single state examples 135
6.2.2 Open-loop unstable system 142
6.2.3 Inverse response 145
6.2.4 Oscillatory system 149
6.2.5 Underdamped system 156
6.2.6 Angular positioning system 163
6.2.7 Constraint saturation example 169
6.2.8 Fired heater example 172

6.3 Multiple-Input-Multiple-Output Examples 183
6.3.1 Ill-conditioned distillation column 183
6.3.2 Three-by-three system 198

Chapter 7 Conclusion **202**

7.1 Single-Model MPC with Zone Regions 202
7.2 Robust MPC 203

Appendix A Notation **208**

Appendix B Definition **217**

Appendix C Linear Matrix Inequalities **219**

Bibliography **221**

Vita **235**

List of Tables

2.1 Terminal stability constraints 13

3.1 Pulse Disturbance Descriptions 58

5.1 Set definitions for $\mathcal{I} = 3$ and $N = 2$. 94

7.1 Summary 207

List of Figures

2.1 Predicted and reference trajectories used in the MPC controller design procedure 10

2.2 MPC closed-loop block diagram 19

2.3 Performance for the nominal plant $A_p = A_m = 0.4, B_p = B_m = 0.1, C_p = C_m = 1$. 26

2.4 Performance with plant-model mismatch $A_p = 0.4, B_p = 0.7, C_p = 1$ and $A_m = 0.4, B_m = 0.1, C_m = 1$. 27

3.1 Slack variable definition 33

3.2 Zone violations penalized in the regulator 34

3.3 Ill-conditioned distillation column 47

3.4 Composition for the unconstrained nominal plant with set point change 50

3.5 Composition for the constrained nominal plant with set point change 51

3.6 Reflux and boil-up flow rate for the constrained nominal plant with set point change 52

3.7 Composition for the constrained nominal plant with conversion of the set point to a zone 53

3.8 Composition for process with step disturbance and plant-model mismatch 56

3.9 Composition for process with pulse disturbances and plant-model mismatch 57

3.10 Composition for process with sensor noise and plant-model mismatch 59

3.11 System's stability as a function of sensor noise and plant-model mismatch 60

4.1 Parametric uncertainty description 66

4.2 SISO feedback loop 70

4.3 Constraint saturation example that exhibits wind-up with plant=model 2. 82

5.1 η as a function of N and $\mathcal{I}$. 93

5.2 Graphical illustration of the branches for $\mathcal{I} = 3$ and $N = 2$. 94

5.3 Tree trajectory with $\mathcal{I} = 2$ and $N = 4$ in the $x \in \mathbb{R}^2$ state space. . . . 95

5.4 Graphical illustration of the steady-state equality constraints without the steady-state bias $p_{i,s}$. 106

5.5 Graphical illustration of the steady-state bias $p_{i,s}$. 106

5.6 Block diagram for the RMPC method 110

6.1 RMPC performance for the model uncertainty description Ω_1 with the plant $A_p = 0.4$, $B_p = 0.7$. 136

6.2 SMPC performance with plant model mismatch, $A_m = 0.4$, $B_m = 0.1$ and $A_p = 0.4$, $B_p = 0.7$. 137

6.3 Comparison of RMPC and SMPC performance with plant model mismatch. $A_m = 0.4$, $B_m = 0.1$ for SMPC and $A_p = 0.4$, $B_p = 0.4$. 139

6.4 Comparison of the RMPC performance with uncertainty description Ω_2 and the SMPC performance with $A_m = A_p = 0.4$, $B_m = B_p = 0.4$. 140

6.5 Comparison of the RMPC performance with uncertainty description Ω_2 and the SMPC performance with $A_m = A_p = 0.4$, $B_m = B_p = 0.1$. 141

6.6 Closed-loop performance for the open-loop unstable system with model 1 as the plant. 143

6.7 Closed-loop performance for the open-loop unstable system with model 2 as the plant. 144

6.8 Open-loop response of the inverse response system 145

6.9 Closed-loop performance for the inverse response system with model 1 as the plant. 147

6.10 Closed-loop performance for the inverse response system with model 2 as the plant. 148

6.11 Suspension system exhibiting oscillatory open-loop response 149

6.12 Open-loop response of the suspension system in figure 6.11 to a unit step change in the input 150
6.13 Closed-loop performance for the suspension system with model 1 as the plant. 152
6.14 Closed-loop performance for the suspension system with model 2 as the plant. 153
6.15 Closed-loop performance for the suspension system with model 3 as the plant. 154
6.16 Open-loop response to a unit step change for the suspension system with $c_s = 400$. 157
6.17 Closed-loop performance for the underdamped system with model 1 as the plant. 158
6.18 Closed-loop performance for the underdamped system with model 2 as the plant. 159
6.19 Closed-loop performance for the underdamped system with model 3 as the plant. 160
6.20 Closed-loop performance for the underdamped system with model 3 as the plant. 161
6.21 Angular positioning system 163
6.22 Open-loop response for the angular positioning system (Figure 6.21).164

6.23 Closed-loop performance for the angular positioning system with model 1 as the plant. 166

6.24 Closed-loop performance for the angular positioning system with model 2 as the plant. 167

6.25 Closed-loop performance for the angular positioning system with $\mu_1 = 0.101$ and $\mu_2 = 0.899$ for the plant while the SMPC model is $\mu_1 = 0.9091$ and $\mu_2 = 0.0909$. 168

6.26 Comparison of closed-loop performance in the presence of constraint saturation for the uncertain models in equation 4.14. 170

6.27 Graphical illustration of disturbance updating using the uncertain models in equation 4.14. 171

6.28 Open-loop response to a unit step change in the fuel gas feed rate for the fired heater example. 172

6.29 Closed-loop performance for the fired heater system with model 1 as the plant. 176

6.30 Closed-loop performance for the fired heater system with model 2 as the plant. 177

6.31 Closed-loop performance for the fired heater system with the plant equal to the average of models 1 and 2. 178

6.32 Comparison of disturbance rejection for the fired heater system with model 1 as the plant. 180

6.33 Comparison of disturbance rejection for the fired heater system with model 2 as the plant. 181
6.34 Comparison of the closed-loop control performance of the RMPC and SMPC with input or output disturbance model for the fired heater system with model 2 as the plant. 182
6.35 Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant. 185
6.36 Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant. 186
6.37 Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant. 187
6.38 Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant. 188
6.39 Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant. 189
6.40 Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant and new Q_w and R_v for SMPC. . . 191
6.41 Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant and new Q_w and R_v for SMPC. . . 192
6.42 Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant and new Q_w and R_v for SMPC. . . 193

6.43 Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant and new Q_w and R_v for SMPC. . . 194
6.44 Closed-loop control performance for the ill-conditioned distillation column with time-varying plant and the new Q_w and R_v for SMPC using model 2. 195
6.45 Closed-loop control performance for the ill-conditioned distillation column with time-varying plant and the new Q_w and R_v for SMPC using model 2. 196
6.46 Closed-loop control performance for the three by three system with model 1 as the plant. 199
6.47 Closed-loop control performance for the three by three system with model 2 as the plant. 200

Chapter 1

Introduction

In the chemical process industries, the main objective is to convert raw materials and energy into finished products. The basic operating principles are:

1. operate the processing units safely;

2. maintain the process at the optimal production rate;

3. maintain the finished products at the specified quality requirements.

The purpose of process control is to make sure the chemical processes follow the basic operating principles above.

Industrial model-based control algorithms first made their appearance in the late 1970's. Industrial implementations began with model algorithmic control (MAC) [132] and dynamic matrix control (DMC) [36]. Model predictive control (MPC) is one of the advanced control theories that has been studied extensively in the research community. MPC, also known as receding horizon control or moving horizon control, includes a class of control theories that use linear or nonlinear process models to forecast system behavior.

The main difference between MPC and other model-based control theories is in its implementation. MPC solves for the optimal open-loop manipulated input trajectory that minimizes the difference between the predicted plant behavior and the desired plant behavior. It differs from other control theories in that the optimal control problem is solved on-line for the current state of the plant, rather than determined off-line as a feedback policy. The first control move in the optimal input trajectory is injected into the plant. Feedback is obtained by measuring the output after the optimal control move has been injected. The controller then solves a new open-loop optimal control problem on-line. The procedure is then repeated.

The explicit use of process models to make process predictions is conceptually appealing to industrial practitioners and partially accounts for the popularity of model-based control theories. MPC's ability to explicitly handle input and output constraints in the optimal control problem distinguishes it from other model-based control theories and increases its attractiveness. MPC has been widely applied in the petrochemical and related industries where satisfaction of constraints is particularly important because efficiency demands operating points on or close to the boundary of the admissible state and control regions. Richalet et al. [131, 132] and Cutler and Ramaker [36] first documented the industrial implementation of MPC. Qin and Badgwell [121, 122] provided a recent survey. They documented more than two thousand commercial applications of MPC ranging from the refining and chemical industries to the aerospace and automotive industries. Blanchini and Pesenti [18] discussed the use of MPC in multi-inventory control.

The success of MPC control performance is highly dependent on the accuracy of the open-loop predictions, which in turn depends on the accuracy of the process models. It is possible for the predicted trajectory to differ, perhaps considerably, from the actual trajectory followed by the plant being controlled [14]. The difference between the plant and the model is known as plant-model mismatch, which can cause the control performance to be sluggish, overly conservative or, in the worst-case scenario, unstable. Because process identification is difficult at best, process modeling errors occur frequently.

Although the output feedback feature in MPC can reduce the discrepancy between the plant and the forecasted behavior, MPC is not designed to explicitly handle plant-model mismatch. Robust model predictive control (RMPC) is the class of predictive control theory that increases the effectiveness of the control actions by explicitly accounting for the modeling errors in the controller design procedure. The three common causes for modeling uncertainty are state estimation error, modeling error, and unknown exogenous disturbances. The state estimation errors arise when output instead of state measurements are available, and the estimate of the state is used in the controller formulation in place of the actual state. Modeling uncertainties may be caused by process identification inaccuracies and changes in operating conditions. Bounded parametric uncertainty regions are used in the controller design to account for the modeling errors. When external disturbances are the cause for the model uncertainty, they are assumed to be bounded disturbances.

The main motivation for research in RMPC is to explicitly account for the model uncertainties in the controller design procedure. Control engineers know that plant-model mismatch may cause unstable or poor controller performance. A common method used by practitioners to decrease the likelihood of instability is to detune the controller so that the control actions become more conservative. The trade off for increased robustness to plant-model mismatch is more conservative and less efficient control performance.

Researchers in both academia and industry realize the importance of robust control. Instead of forecasting system behavior using one process model as in MPC, RMPC forecasts system behavior for every model in the uncertainty set. Two common descriptions used to account for model uncertainty are parameter uncertainties and bounded exogenous disturbances. The optimal control actions are determined by a *min-max* optimization that minimizes the deviations of the forecasted behavior from the desired behavior for the model with the largest deviations. The *min-max* optimization originated from research in dynamic game theory, and the terminology of “minimax controller” was adopted from the statistical decision theory of the 1950’s [5]. Campo and Morari [22], Genceli and Nikolaou [52], and Zheng and Morari [159] proposed using the *min-max* optimization in MPC with finite impulse response models. Lee and Cooley [80, 82] proposed using the *min-max* optimization in MPC on state-space models with time-varying parametric uncertainty in the B matrix. Lee and Yu [83] proposed using the *min-max* optimization on discrete state-space models with polytopic model uncertainty. The on-line *min-max* optimization is computationally intensive. Kothare et al. [69], Casavola et al. [24], and Lu and Arkun [90, 91] proposed using linear matrix inequalities (LMI) based optimization to solve for the gain in the state feedback control policy $u(t) = Kx(t)$ that minimizes the model in the polytope with the largest deviation from the origin. The LMI-based optimization is performed on-line to determine a new K at every time step.

Badgwell [7, 6] proposed using a nominal model to determine the optimal control action. Robust stability is achieved by adding the following constraint to the sum of the forecasted behavior’s deviations from the origin:

$$\Phi(t+1) \leq \Phi(t)$$

for all models in the polytope. Similarly, Slupphaug and Foss [146] used a nominal model to determine the optimal control actions while guaranteeing closed-loop stability for all models in the polytope by adding the constraint

$$\| x(t+1) \|^2 \leq \| x(t) \|^2 .$$

Pannocchia and Semino [116, 140] proposed modifying the nominal model to robustly stabilize the system for the given model uncertainty description. Chisci et al. [28, 29], Lee and Kouvaritakis [85] and Kouvaritakis et al. [73] proposed using the linear state feedback control policy in an invariant terminal region. A nominal model is used to determine the N control moves needed to move the forecasted plant behavior into the terminal region.

When the output set point is non-zero, besides determining if a robustly stabilizing control policy exists for convergence to the origin, RMPC needs to determine if the control policy can achieve non-zero set point tracking. Khammash and Zou [66] and Bemporad and Mosca [12] proposed using an input reference trajectory to achieve offset-free set point tracking. If the input reference trajectory is unknown, Bemporad [11], Christofides [31], Freeman and Kokotovic [49], and Blanchini [16] showed that if the model uncertainty is described by a nominal model with bounded disturbances, offset-free non-zero set point tracking can be achieved if the disturbances converge to zero. Rossiter and Kouvaritakis [134], Lee et al. [78], Rodrigues and Odloak [133], and Megias et al. [101] proposed using models in the velocity form and minimizing the sum of $y(t) - y_t$ and $u(t+1) - u(t)$ with the terminal constraint $u(t+j+1) - u(t+j) = 0$ for all $j \geq N$. Pannocchia and Rawlings [114] showed the steady state achieved by the above controller depends on the horizon length N when using process models in the velocity form.

Marconi and Isidori [96] stated that correct steady-state x_s and u_s values are needed to achieve "perfect tracking" of the set point. Kassmann et al. [64] proposed using a nominal model in the velocity form to compute the steady-state x_s and u_s values. Ralhan and Badgwell [124] proposed adding an estimated output disturbance to the process model to achieve offset-free control to the zero set point but do not discuss non-zero set point tracking.

With all the research that has been done in the robust MPC field, few researchers have addressed the issue of non-zero set point tracking. Most researchers proved closed-loop stability for convergence to the origin or a single known steady state. However, industrial processes rarely operate at the same steady state at all times. The addition of non-zero set points to robust control increases the complexity of the design procedure.

In this report, we first present in Chapter 2 a review of the MPC control theory in the absence of model uncertainty, which is also known as single-model MPC (SMPC). Chapter 3 compares the control performances of the SMPC algorithm with set points as the optimal operating conditions to the SMPC algorithms with zone regions. In Chapter 4 we summarize the published literature in the robust control field. Chapter 5 discusses the new robust MPC theory we are proposing. First, we define the model uncertainty description used in our controller design procedure. We then design a robust regulator that forecasts time-varying uncertainty. Next, we add integrators to achieve non-zero set point tracking.

The robustly stabilizing control policy is found by formulating the robust MPC controller as an LMI- based optimization problem that is solved off-line. Chapter 6 compares the simulation results using the RMPC algorithm to those using the SMPC algorithm with various process models. Chapter 7 summarizes the findings in this research project and provides Table 7.1, a summary of all the simulation examples examined in Chapter 6.

Chapter 2

Model Predictive Control

2.1 Introduction

Model predictive control (MPC) determines the optimal control actions by minimizing a performance objective that penalizes the difference between the predicted input and output trajectories from their reference trajectories, respectively. Figure 2.1 is a graphical representation of the MPC optimization problem. Feedback is incorporated into the control theory by injecting the first control action into the process and then solving the optimization problem again once process measurement becomes available.

Both researchers and industrial practitioners have studied the theoretical development and on-line implementation of MPC. Industrial applications began with IDCOM (identification and command) and DMC (dynamic matrix control) proposed by Richalet et al. [131, 132] and Cutler and Ramaker [36], respectively.

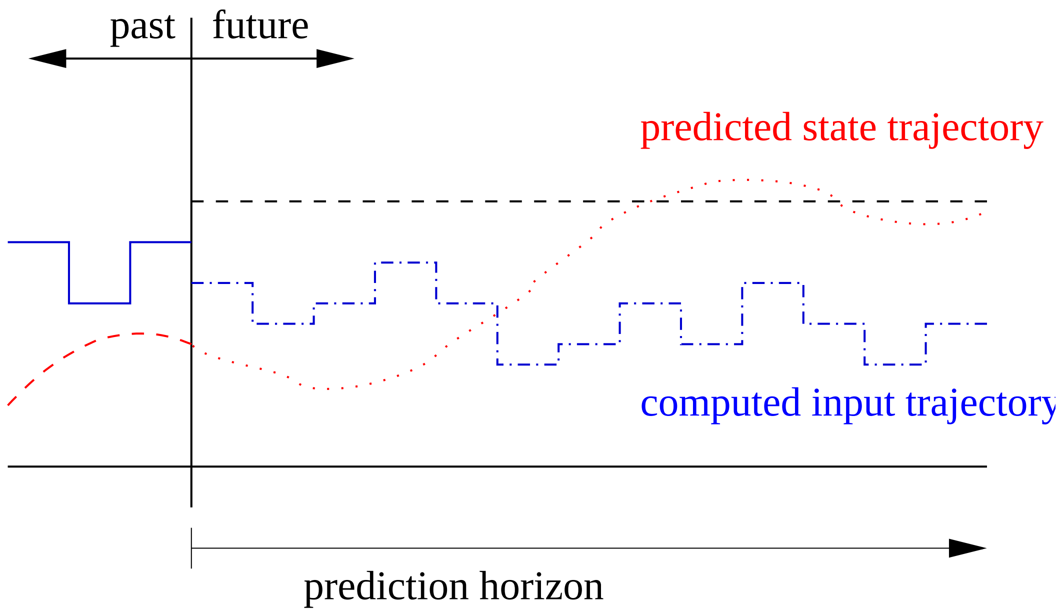

Figure 2.1: Predicted and reference trajectories used in the MPC controller design procedure

Qin and Badgwell [121, 122] provided the most recent survey of industrial applications of MPC. Theoretical investigation of open-loop optimal feedback began with Dreyfus [44]. Kleinman [67], Thomas [150], and Kwon and Pearson [76, 77] were among the first to prove stability for unconstrained linear systems. See Garcia et al. [51] and Mayne et al. [100] for summaries of the theoretical development of MPC. The MPC formulation given in this chapter is based on the formulation proposed by Muske and Rawlings [110] and Rawlings [127].

2.2 Problem Statement

Suppose the system forecast is

$$x(t+j+1|t) = Ax(t+j|t) + Bu(t+j|t) \tag{2.1}$$

$$y(t+j+1|t) = Cx(t+j+1|t)$$

in which $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$ are the model matrices and $x(t+j|t) \in \mathbb{R}^n$, $u(t+j|t) \in \mathbb{R}^m$, and $y(t+j|t) \in \mathbb{R}^p$ are the state, the manipulated input, and the controlled output vectors, respectively, with output measurement up to time t. Physical limitations of the process (e.g. valve saturation) and operating conditions (e.g. safety limits) require the following constraints on $u(t+j|t)$ and $y(t+j|t)$:

$$u_{\min} \leq u(t+j|t) \leq u_{\max} \tag{2.2}$$

$$y_{\min} \leq y(t+j|t) \leq y_{\max}.$$

The open-loop control actions are computed by the following optimization.

$$\min_{u(t|t),u(t+1|t),\ldots} \Phi \tag{2.3}$$

in which Φ is the infinite horizon performance objective and is defined as

$$\Phi = \sum_{j=0}^{\infty} y^T(t+j|t)\bar{Q}y(t+j|t) + u^T(t+j|t)Ru(t+j|t). \tag{2.4}$$

$\bar{Q} > 0$ and $R > 0$ penalize the output and input deviations from the origin, respectively. To reduce the notation complexity, let $x(t+j|t) = x(t+j)$, $u(t+j|t) = u(t+j)$, and $y(t+j|t) = y(t+j)$. Measurement up to time t is implied.

Definition 2.2.1 *The process model* (A, B) *is* **controllable** *if for every* $x(0)$ *and* $x(n)$ *there exists* $u(0), \ldots, u(n-1)$ *such that*

$$x(n) = A^n x(0) + \mathbf{C} \begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix}$$

The controllability matrix is

$$\mathbf{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \in \mathbb{R}^{n \times n(m)}$$

(A, B) is controllable if the rank of $\mathbf{C}$ is equal to n.

Definition 2.2.2 *Consider the following partitioned system*

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} u(t)$$

(A, B) *is* **stabilizable** *if* (A_{11}, B_1) *is controllable and the eigenvalues of* A_{22} *are inside the unit circle.*

2.3 Infinite Horizon Regulator Stability

The infinite horizon regulator is based on the minimization of the performance objective Φ subject to a process model (Equation 2.1) and the input and output constraints (Equation 2.2). The infinite horizon regulator computes the optimal

1	A terminal cost Φ_N
2	A terminal stability constraint $x(t+N) \in W$
3	A local control law $u(t) = f(x(t))$

Table 2.1: Terminal stability constraints

input trajectory denoted by $\pi^*(t) = [u^*(t), u^*(t+1), \ldots]^T$, and $u^*(t)$ is the optimal control action injected into the process at time t. The procedure is repeated at each successive control interval with feedback incorporated by using the plant measurements to update the state vector.

For unconstrained systems, researchers used dynamic programming to prove regulator stability. Mayne et al. [100] summarized the ingredients necessary to prove infinite horizon regulator stability for constrained systems (Table 2.1). The key is to reduce the infinite horizon regulator to a finite horizon regulator with a terminal constraint.

$$\Phi_N = \sum_{j=N}^{\infty} x^T(t+j)Qx(t+j) + u^T(t+j)Ru(t+j) \tag{2.5}$$

$\Phi_N \geq 0$ is the terminal cost, and $x(t+N)$ is the state at $j = N$ in the prediction horizon. W is an invariant terminal region where states and inputs evolve without constraint violations, and N is the finite prediction horizon length. The first terminal stability constraint used to prove stability is $x(t+N) = 0$ [25, 32, 65, 71, 98, 99, 104]. Muske and Rawlings [109, 110, 130] replaced the $x(t+N) = 0$ constraint with the control policy $u(t+j) = 0$ for all $j \geq N$ and showed methods for finding the terminal cost for both open-loop stable and unstable systems. Michalska and Mayne [102], Scokaert and Rawlings [138, 139] and Chmielewski and Manousiouthakis [30] proved stability by requiring the satisfaction of the terminal inequality constraint $x(t+j) \in W$ for all $j \geq N$, which is positively invariant under the control policy $u(t+j) = Kx(t+j)$. N is chosen to be sufficiently large so that the terminal state lies in the state admissible set W and remains in W without constraint violation [53, 54].

Optimization of a constrained infinite horizon objective function is not computationally tractable. The addition of a terminal stability constraint and cost changes the infinite horizon performance objective to a finite horizon performance objective. The terminal cost is $\Phi_N = x^T(t+N)\mathcal{Q}x(t+N)$ in which $\mathcal{Q} \geq 0$ is the terminal state penalty matrix. The optimization in Equation 2.3 becomes

$$\min_{u^N} x^T(t+N)\mathcal{Q}x(t+N) + \sum_{j=0}^{N-1}(x^T(t+j)C^TQCx(t+j) + u^T(t+j)Ru(t+j)) \quad (2.6)$$

in which u^N is a vector containing the N future open-loop control moves to be determined by the on-line optimizer.

$$u^N = \begin{bmatrix} u(t) \\ u(t+1) \\ u(t+2) \\ \vdots \\ u(t+N-1) \end{bmatrix} \quad (2.7)$$

For the terminal stability constraint $x(t+N) \in W$ in which $u(t+j) = Kx(t+j)$ for all $j \geq N$, $\mathcal{Q}$ is defined by the infinite sum.

$$\mathcal{Q} = \sum_{i=0}^{\infty}(A+BK)^{i^T}C^TQC(A+BK)^i \quad (2.8)$$

The open-loop infinite horizon dynamic regulator is stable if there exists a K such that there exists a $\mathcal{Q}$ such that the following discrete Lyapunov equation is satisfied.

$$(A + BK)^T \mathcal{Q}(A + BK) - \mathcal{Q} = -C^T QC \tag{2.9}$$

The existence of $\mathcal{Q}$ is equivalent to the eigenvalues of $(A + BK)$ lying inside the unit circle, which implies that the state $x(t)$ converges to zero as $t \longrightarrow \infty$ [75].

2.4 State Estimation

The objective of the state estimator is to estimate the initial state x_0 for a given string of measured outputs. The state can be estimated only if it can be observed.

Definition 2.4.1 *Given n data measurements $y(0), y(1), \ldots, y(n-1)$, (A, C) is* **observable** *if the initial state $x(0)$ can be uniquely determined.*

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} x(0) = \mathbf{O}x(0)$$

$\mathbf{O} \in \mathbb{R}^{n(p)\times n}$ *is the observability matrix.*

(A, C) is observable if the rank of $\mathbf{O}$ is equal to n.

Definition 2.4.2 *Consider the following partitioned system*

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}_t$$

(A, C) *is* **detectable** *if* (A_{11}, C_1) *is observable and the eigenvalues of* A_{22} *are inside the unit circle.*

Most output measurements have sensor noise. A filter is necessary to remove this noise and accurately estimate the state values. The process model used to determine the optimal filter includes white noise with known covariance in the output and state.

$$x(t+1) = Ax(t) + Bu(t) + G_w w(t) \tag{2.10}$$

$$y(t) = Cx(t) + v(t)$$

in which $w(t)$ and $v(t)$ are zero-mean, uncorrelated, normally distributed, stochastic variables.

Linear estimation theory was introduced by Kalman [61] and Kalman and Bucy [62] in the 1960's. They developed the optimal linear observer in Equation 2.11, which estimates $\hat{x}(t+1|t)$, the state at time $t+1$ using output measurements up to time t.

$$\hat{x}(t+1|t) = A\hat{x}(t|t-1) + Bu(t) + L(y(t) - C\hat{x}(t|t-1)) \tag{2.11}$$

L is the discrete Kalman filter that minimizes the mean-square error of the state estimate. It is computed from the solution of the discrete filtering steady-state Riccati equation with Q_w and R_v, the covariance matrices for $w(t)$ and $v(t)$, respectively.

$$\Xi = A[\Xi - \Xi C^T(C\Xi C^T + R_v)^{-1}C\Xi]A^T + G_w Q_w G_w^T \tag{2.12}$$
$$L = A\Xi C^T(C\Xi C^T + R_v)^{-1}$$

Equation 2.11 optimally reconstructs the states from the output measurements given the noise assumptions. The Riccati formulation in Equation 2.12 guarantees the filter is nominally stable if $R_v > 0$, (A, C) detectable, and $(A, G_w Q_w^{1/2})$ stabilizable [128].

2.5 Output Feedback

Output feedback is incorporated when output measurements are used to update the state estimates. Differences between $y(t)$, the measured output, and $\hat{y}(t+1|t) = C\hat{x}(t+1|t)$, the estimated output, can be caused by measurement noise or unmodeled disturbances. The optimal observer can remove sensor noise when estimating the state, but it cannot remove unmodeled disturbances. Disturbance modeling is necessary for successful disturbance rejection without offset [108, 115]. Davison and Smith [40] showed that a constant step disturbance model is stabilizing and eliminates steady-state offset. The two common disturbance models used by researchers are the input and output step disturbance models.

The input disturbance model places a step disturbance in the input. In the regulator, before the next output measurement becomes available, the disturbance is assumed to be constant. Its value is updated in the estimator when new output measurements become available. Equation 2.13 is the regulator model with the input disturbance model.

$$\begin{aligned} x(t+1) &= Ax(t) + B(u(t) + z(t)) \\ z(t+1) &= z(t) \\ y(t+1) &= Cx(t+1) \end{aligned} \tag{2.13}$$

in which $z(t) \in \mathbb{R}^m$ models an input step disturbance. When output measurements are available, estimates for both x and z are determined by

$$\begin{aligned} \hat{x}(t+1|t) &= A\hat{x}(t|t-1) + B(u(t) + \hat{z}(t|t-1)) + L_x(y(t) - C\hat{x}(t|t-1)) \\ \hat{z}(t+1|t) &= \hat{z}(t|t-1) + L_z(y(t) - C\hat{x}(t|t-1)) \end{aligned} \tag{2.14}$$

The filters L_x and L_z are observers for both the state and the disturbance vectors. L_x and L_z are solutions of the discrete steady-state Riccati equation (Equation 2.12) using the augmented process model $(\tilde{A}_z, \tilde{B}_z, \tilde{C}_z, \tilde{G}_w)$.

$$\begin{aligned} \begin{bmatrix} x(t+1) \\ z(t+1) \end{bmatrix} &= \tilde{A}_z \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \tilde{B}_z u(t) + \tilde{G}_w w(t) \\ y(t+1) &= \tilde{C}_z \begin{bmatrix} x(t+1) \\ z(t+1) \end{bmatrix} + v(t+1) \end{aligned} \tag{2.15}$$

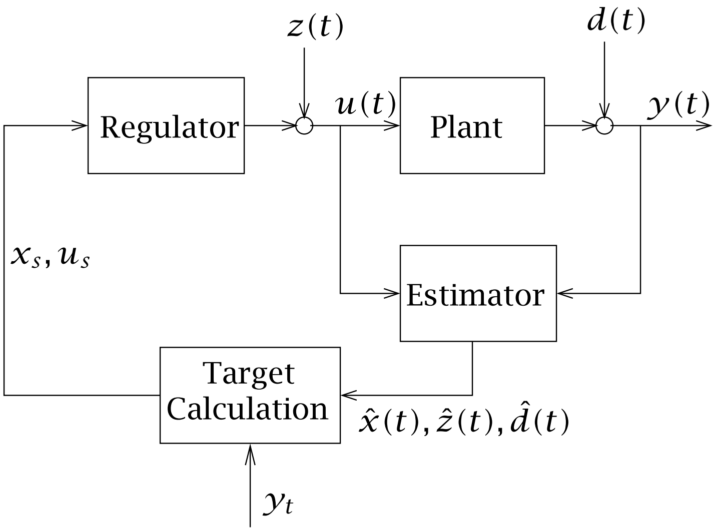

Figure 2.2: MPC closed-loop block diagram

in which

$$\tilde{A}_z = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \tilde{B}_z = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \tilde{G}_w = \begin{bmatrix} G_w \\ 0 \end{bmatrix}, \quad \tilde{C}_z = \begin{bmatrix} C & 0 \end{bmatrix}$$

and

$$L = \begin{bmatrix} L_x \\ L_z \end{bmatrix}$$

Muske and Badgwell [108] showed that $(\tilde{A}_z, \tilde{C}_z)$ is detectable.

The main difference between the output and the input disturbance models is the placement of the disturbance (see Figure 5.6). The output disturbance model places a step disturbance in the output. Similar to the input disturbance model, the disturbance is assumed to be constant in the regulator.

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

in which $d(t) \in \mathbb{R}^p$ denotes the output step disturbance. Equation 2.17 shows the observer used to update $\hat{x}(t)$ and $\hat{d}(t)$.

$$
\begin{aligned}
\hat{x}(t+1|t) &= A\hat{x}(t|t-1) + Bu(t) + L_x(y(t) - C\hat{x}(t|t-1) - \hat{d}(t|t-1)) \\
\hat{d}(t+1|t) &= \hat{d}(t|t-1) + L_d(y(t) - C\hat{x}(t|t-1) - \hat{d}(t|t-1))
\end{aligned}
\tag{2.17}
$$

Similar to the input disturbance model, the optimal filters L_x and L_d are determined by the solution of Equation 2.12 after state augmentation.

$$
\begin{aligned}
\begin{bmatrix} x(t+1) \\ d(t+1) \end{bmatrix} &= \tilde{A}_d \begin{bmatrix} x(t) \\ d(t) \end{bmatrix} + \tilde{B}_d u(t) + \tilde{G}_w w(t) \\
y(t+1) &= \tilde{C}_d \begin{bmatrix} x(t+1) \\ d(t+1) \end{bmatrix} + v(t+1)
\end{aligned}
\tag{2.18}
$$

in which

$$
\tilde{A}_d = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}, \qquad \tilde{B}_d = \begin{bmatrix} B \\ 0 \end{bmatrix}, \qquad \tilde{C}_d = \begin{bmatrix} C & I \end{bmatrix}
$$

and

$$
L = \begin{bmatrix} L_x \\ L_d \end{bmatrix}
$$

The augmented model $(\tilde{A}_d, \tilde{C}_d)$ is detectable [108, 115]. The output disturbance design implicitly assumes perfect, noise free measurements, which may lead to unacceptable performance. Shinskey [143] showed that MPC with an output disturbance model may perform poorly if the disturbance enters the system through the input rather than the output.

2.6 Target Calculation

Most physical processes operate with the target value or set point y_t for the controlled outputs. Input targets u_t may also be present, especially for processes that have more manipulated inputs than controlled outputs. The target values are the desired operating conditions. It is unusual that these desired values are the origin. In some operations, the desired values can vary over time (e.g. optimal trajectory during process start-up). Therefore, target, or set point, tracking is an important part of any control theory.

The target calculation is a standard idea that dates back to the early days of optimal control theory [75]. Cutler et al. [34] discussed the idea in the context of MPC. The objective for the target calculation is to determine the feasible steady-state values to which the dynamic regulator converges, that minimize their deviation from the target values. In the absence of constraints and disturbances, the steady-state values of the process model are equal to the target values. But when input and output constraints are present, the process may operate near or on the constraints, and the target values may be outside the constrained region. Exogenous disturbances can cause the optimal operating conditions to deviate from the feasible region, thus making the targets unreachable. If the target calculation is absent, the dynamic regulator assumes the target values are reachable. The difference between what can be achieved and what the controller forecasts may cause sub-optimal control performance. The steady-state values are recalculated every time the MPC controller executes because the disturbances entering the system or new input information from the operator may change the location of the optimal steady state.

Target tracking can be formulated as an optimization problem that uses a quadratic performance objective to minimize the deviation of the steady-state output and input from their target values. Either the input or output disturbance model can be used to determine the correct steady-state values that guarantee offset-free control for any plant dynamics provided the closed-loop system is stable and none of the constraints are active at steady state [59, 129].

When input target u_t is available, the steady-state input and state values, u_s and x_s, that remove constant disturbances are determined by Equation 2.19.

$$\min_{x_s,u_s} (u_s - u_t)^T R_s (u_s - u_t) \tag{2.19}$$

subject to

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} B\hat{z}(t) \\ y_t - \hat{d}(t) \end{bmatrix}$$

$$u_{\min} \leq u_s \leq u_{\max}$$

$$y_{\min} \leq y_s \leq y_{\max}$$

in which $\hat{z}(t)$ and $\hat{d}(t)$ are the input and output step disturbance estimates at time t. The matrix $R_s > 0$ is the penalty matrix on the deviation of the steady-state input from its target. Depending on which step disturbance model is chosen, one of the step disturbance vectors is deleted. If both step disturbance vectors are equal to zero, then there is no integral control. The disturbance models are the integrators in the MPC formulation. When unmodeled disturbances are present, the disturbance estimates cause the target calculation to shift the steady-state values so that the unmodeled disturbances can be rejected.

If the quadratic program in Equation 2.19 is infeasible or if no input target is available, then the output target tracking error is minimized to determine the best possible x_s and u_s.

$$\min_{x_s, u_s} (y_t - Cx_s - \hat{d}(t))^T Q_s (y_t - Cx_s - \hat{d}(t)) \tag{2.20}$$

subject to

$$\begin{bmatrix} I - A & -B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = B\hat{z}(t)$$

$$u_{\min} \le u_s \le u_{\max}$$

$$y_{\min} \le y_s \le y_{\max}$$

in which $Q_s > 0$ is the penalty matrix on the output tracking error.

Offset-free set point tracking can be accomplished by Equation 2.20 if there are as many manipulated inputs as there are controlled outputs. However, if there are more controlled outputs than manipulated inputs, then offset-free control cannot be achieved. If there are more manipulated inputs than controlled outputs, then the optimization problem in Equation 2.20 cannot find unique x_s and u_s values.

2.7 Closed-loop Stability

A system that is open-loop stable may not be closed-loop stable because of plant-model mismatch, which surfaces through output measurement feedback. For unconstrained linear systems, closed-loop stability is determined by the following augmented system.

$$\begin{bmatrix} x(t+1) \\ \hat{x}(t+1) \end{bmatrix} = \begin{bmatrix} A_p & B_pK \\ LC_p & A_m + B_mK - LC_m \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} = \tilde{A}_c \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} \tag{2.21}$$

in which (A_p, B_p, C_p) and (A_m, B_m, C_m) are the plant and the process models, respectively. The system is closed-loop stable if and only if the eigenvalues of $\tilde{A}_c$ are inside the unit circle.

When we have a nominal plant, no plant-model mismatch, Equation 2.21 is becomes Equation 2.22 after some algebraic manipulations.

$$\begin{aligned}\begin{bmatrix} x(t+1) \\ x(t+1) - \hat{x}(t+1) \end{bmatrix} &= \begin{bmatrix} A+BK & -BK \\ 0 & A-LC \end{bmatrix} \begin{bmatrix} x(t) \\ x(t) - \hat{x}(t) \end{bmatrix} \\ &= \tilde{A}_c' \begin{bmatrix} x(t) \\ x(t) - \hat{x}(t) \end{bmatrix} \end{aligned} \tag{2.22}$$

in which (A, B, C) describes the plant and the model. The eigenvalues of $\tilde{A}_c'$ are inside the unit circle if the eigenvalues of $(A + BK)$ and $(A - LC)$ are inside the unit circle. In other words, the nominal plant is closed-loop stable if there exist stabilizing feedback gain K and Kalman filter L. The filter guarantees the error between the estimated state $\hat{x}(t)$ and the actual state $x(t)$ converges to zero as $t \longrightarrow \infty$.

When plant-model mismatch is present, Equation 2.21 cannot be simplified because $(A_p, B_p, C_p) \neq (A_m, B_m, C_m)$. The closed-loop stability cannot be easily evaluated because (A_p, B_p, C_p) is not known and the nominal closed-loop stability does not guarantee closed-loop stability. Figures 2.3 and 2.4 are the simulation results for the following process model.

$$A_m = 0.4, \quad B_m = 0.1, \quad C_m = 1$$

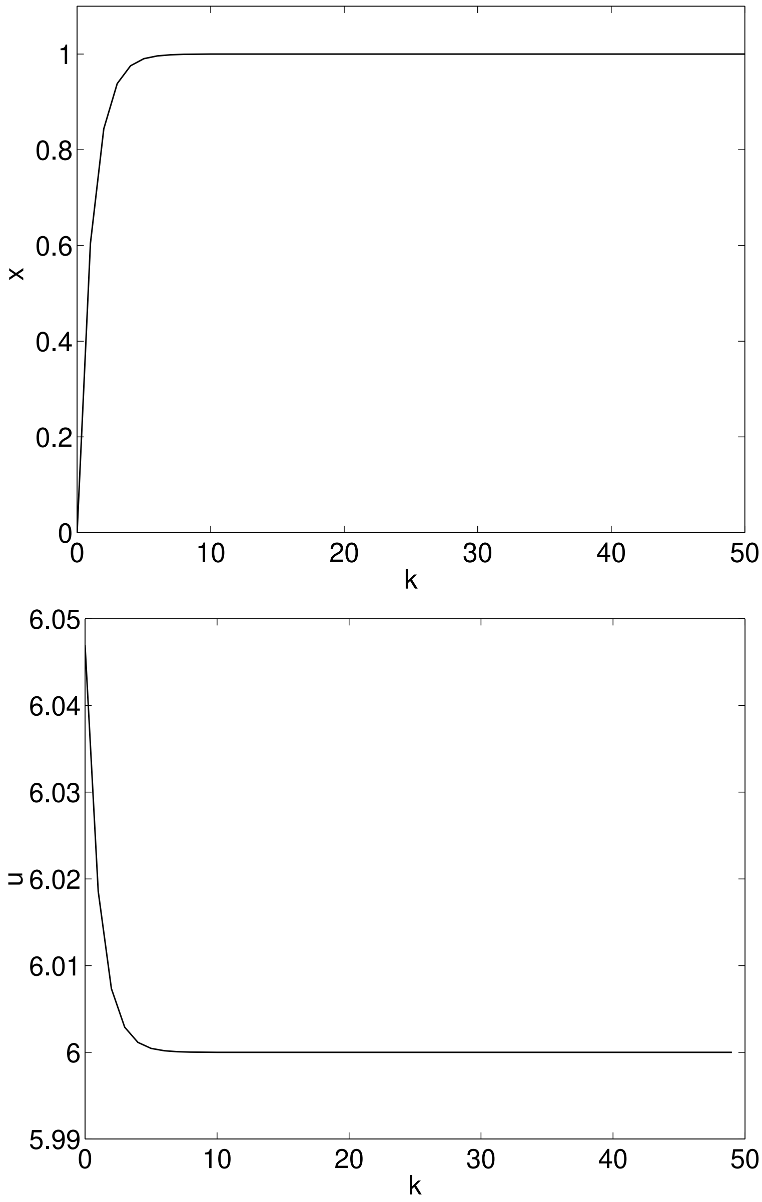

Figure 2.3: Performance for the nominal plant $A_p = A_m = 0.4, B_p = B_m = 0.1, C_p = C_m = 1$.

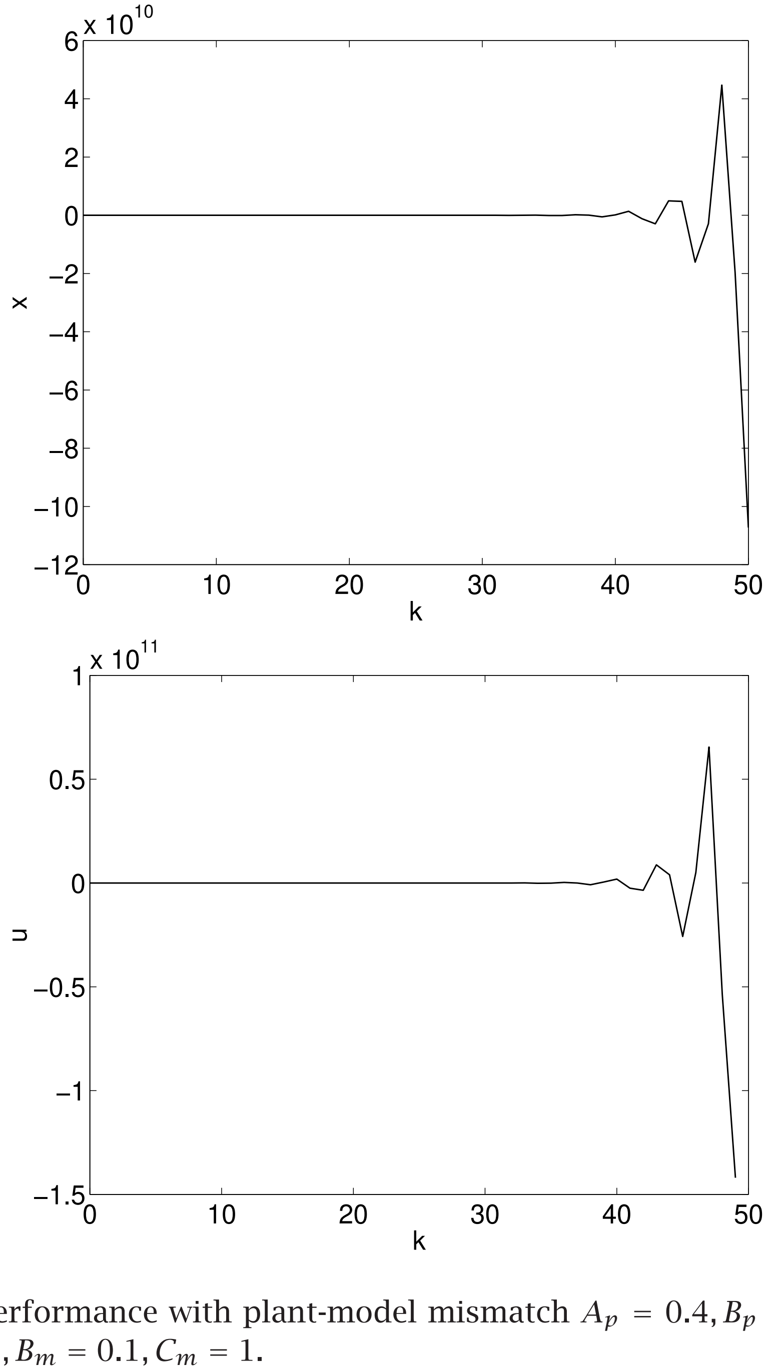

Figure 2.4: Performance with plant-model mismatch $A_p = 0.4, B_p = 0.7, C_p = 1$ and $A_m = 0.4, B_m = 0.1, C_m = 1$.

with $\bar{Q} = 1$, $R = 1$. Figure 2.3 shows the closed-loop performance with no plant-model mismatch. The SMPC algorithm successfully controls the system from the origin to the set point $y_t = 1$ using the output disturbance model. The eigenvalues of $\tilde{A}_c$ are

$$\begin{bmatrix} -0.1764 \\ 0.8027 \\ 0.3953 \end{bmatrix}$$

which are inside the unit circle. Figure 2.4 shows the unstable closed-loop performance in the presence of plant-model mismatch. The plant is given by the following model.

$$A_p = 0.4, \quad B_p = 0.7, \quad C_p = 1$$

The eigenvalues of $\tilde{A}_c$ with the plant (A_p, B_p, C_p) and the output disturbance model are

$$\begin{bmatrix} 0.4064 \\ 0.3076 + 1.0843i \\ 0.3076 - 1.0843i \end{bmatrix}$$

which are outside the unit circle.

The closed-loop instability shown in Figure 2.4 is caused by output feedback when there is plant-model mismatch, which is the main disadvantage for the SMPC formulation. The goal of the RMPC theory is to include model uncertainty descriptions in the regulator so that plant-model mismatch can occur only if the uncertainty description is not large enough to include all uncertainties. RMPC guarantees closed-loop stability for any plant that can be described by a model in the uncertainty description.

Chapter 3

Single-Model MPC with Zone Regions

3.1 Introduction

In conventional SMPC theory, we assume the controlled, or measured, variable has a specific set point. Often in industrial applications, not all of the controlled outputs have set points. For example, if the final chemical product compositions have predetermined market specifications, then the controlled outputs, product compositions, have set points. In the same process, there may be measured variables that have ranges of desired values. A common example would be the system pressure, which needs to fall within a specified range in order to satisfy process safety regulations. In other words, the system pressure has no set point. In this situation zone regions are useful.

Zone regions may also be necessary for over-specified processes. When there are too many input and output constraints, the process may not be able to meet all the set points. If one of the set points is changed into a zone region, the output specifications are relaxed slightly. The probability that the process will meet all of its specifications, zone regions and set points, increases.

This chapter investigates the effect of zone regions on the ability of an ill-conditioned distillation column to meet all of the required set points while the process is subjected to input constraints. The effect of plant-model mismatch on the stability of a two-point distillation column is evaluated. Before the process can be explored, the changes that need to be made in the SMPC's objective function are considered. When zone regions are present, the SMPC algorithm formulation needs to be altered slightly, and new variables need to be defined.

3.2 Motivation

Ever since model based control algorithms were first developed, researchers have realized the importance of having zone regions instead of set points as the control objectives of some control variables. Zone regions are relevant for processes with outputs that have ranges of desired values instead of specific values, also known as set points. They are also significant for control systems with more controlled variables than manipulated variables.

Most SMPC applications are designed for square systems[1] and systems with

[1] A square system is a system with the same number of manipulated inputs as controlled outputs.

more manipulated variables than controlled variables. When there are fewer manipulated variables than controlled variables, SMPC controllers can still control the process, but offset-free control cannot be achieved for all controlled outputs [115]. Currently, industrial practitioners have two different methods for controlling systems with more controlled variables than manipulated variables. The first one is to relinquish control on enough controlled variables so that the final control system has the same number of controlled variables as the number of available manipulated variables. The second solution is to minimize the deviation of all controlled variables from their set points. Maintaining control on more controlled variables than manipulated variables causes offset between all controlled variables and their respective set points. Another proposed solution is to use zone regions instead of set points as the control objectives for some of the outputs. In this case, all outputs are controlled. The ones with set points are controlled so that there is no offset, and the outputs with zone regions as their control objectives stay inside the zone boundaries.

3.3 Zone Definition

A zone region is defined by the minimum and maximum values of a controlled variable's desired range of values. The minimum value is the lower limit of the

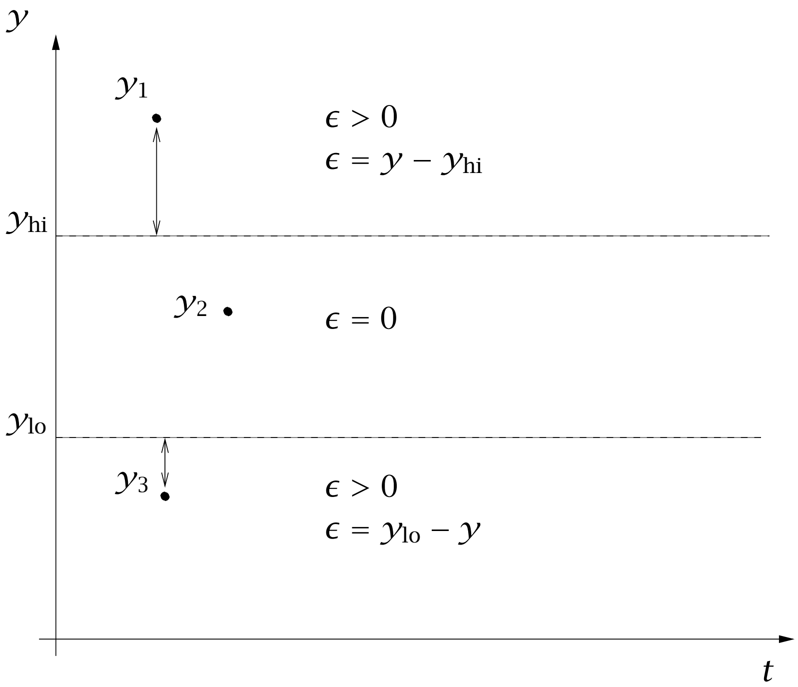

Figure 3.1: Slack variable definition

zone region, and the maximum value is the upper limit of the zone region.

$$\text{Lower limit} = y_{\text{lo}}$$

$$\text{Upper limit} = y_{\text{hi}}$$

The code used to implement zone regions in the current linear SMPC algorithm is similar to the code used to implement soft constraints [50, 120]. The difference between the output measurements and zone limits is defined by a slack variable ϵ. Whenever the measurement is outside the zone, ϵ is positive, but when the measurement is inside the zone, ϵ is equal to zero (see Figure 3.1).

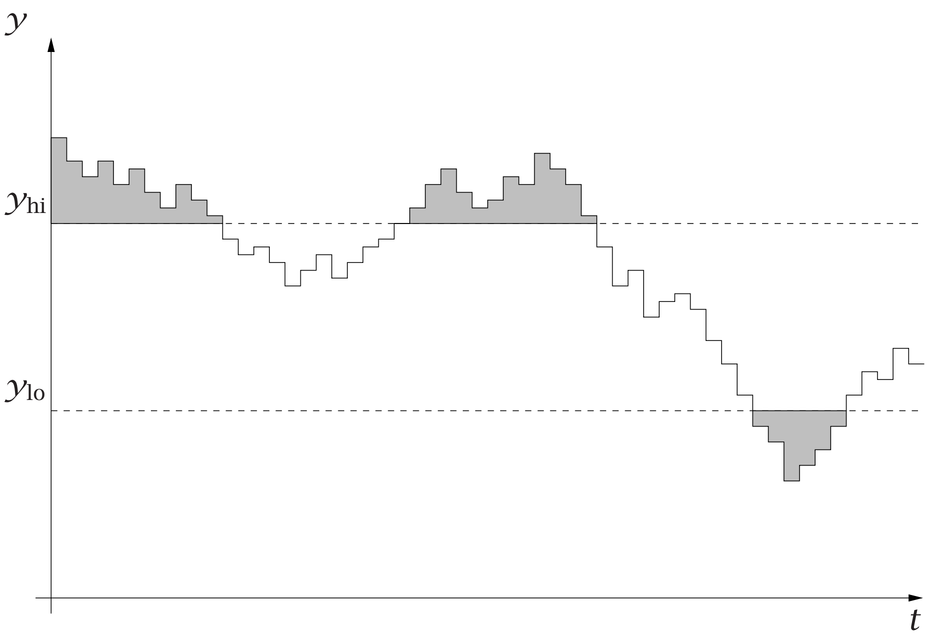

Figure 3.2: Zone violations penalized in the regulator

The slack variable ϵ is defined as:

$$y = y_{\mathrm{hi}} + \epsilon \quad \text{for} \quad y > y_{\mathrm{hi}}$$

$$y = y_{\mathrm{lo}} - \epsilon \quad \text{for} \quad y < y_{\mathrm{lo}}.$$

The value of ϵ measures the deviation of the controlled variable from the limits of the zone region. The SMPC regulator cost function uses ϵ to penalize zone violations similar to set point violations [35] (see Figure 3.2). When ϵ is positive, control actions are taken in an attempt to bring the output back into the zone region. If ϵ is equal to zero, the control algorithm does not take any control action even if the output moves around inside the zone [35, 57].

Slack variables can also be used for outputs that have set points. When the controlled variable has a set point instead of a zone region, both the upper and lower limits of the zone are set equal to the set point y_t.

$$y_{\text{lo}} = y_t$$

$$y_{\text{hi}} = y_t$$

3.4 Target Calculation

In most control processes, there are target values for both the outputs and inputs, y_t and u_t, respectively. These values are usually determined by process optimizations or product specifications. The output target values can be either set points or zone regions. Most controlled processes are limited by input and output constraints, such as valve opening or product composition. Because some input constraints may not allow the process to reach the output target values, target calculation is performed at each time step to minimize the deviations of input and output steady-state values from the target values.

In the conventional SMPC theory, when all the outputs have set points, the steady-state outputs and inputs, y_s and u_s, are calculated by minimizing the deviations of inputs and outputs from their targets y_t and u_t, respectively. The minimization is formulated as the following quadratic program [110]. The minimization is solved subjected to the steady-state dynamic equation as well as the input and output constraints.

$$\min_{x_s,u_s}(u_s - u_t)^T R_s(u_s - u_t) + (y_s - y_t)^T Q_s(y_s - y_t) \tag{3.1}$$

subject to

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_s \end{bmatrix}$$

$$u_{\min} \leq u_s \leq u_{\max}$$

$$y_{\min} \leq y_s \leq y_{\max}$$

R_s is a positive definite weighting matrix that penalizes input deviations from their targets. Q_s is a positive semidefinite weighting matrix that penalizes output deviations from their targets.

In the modified SMPC theory, which is used for processes with outputs that have set points and zone regions as the control objectives, the target calculation is performed by the following quadratic program.

$$\min_{u_s,\epsilon_s} (u_s - u_t)^T R_s (u_s - u_t) + \epsilon_s^T S_s \epsilon_s \tag{3.2}$$

subject to

$$\begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_s \end{bmatrix}$$

$$y_s \leq y_{\text{hi}} + \epsilon_s$$

$$y_s \geq y_{\text{lo}} - \epsilon_s$$

$$u_{\text{min}} \leq u_s \leq u_{\text{max}}$$

$$y_{\text{min}} \leq y_s \leq y_{\text{max}}$$

$$\epsilon_s \geq 0$$

ϵ_s is the slack variable for steady-state output values, and S_s is a positive semidefinite weighting matrix that penalizes the steady-state output deviation from the zone limits.

When ϵ_s is non-zero, the steady-state output falls outside of the zone region. In this case, the control objective in the regulator changes from a zone region to a set point, and the advantages of having zones are lost. To restore the zone region advantages, the zone limits are enlarged so that y_s is equal to either the upper or lower limit of the zone.

$$y'_{\text{hi}} = y_{\text{hi}} + \epsilon_s \quad \text{if} \quad y_s > y_{\text{hi}}$$

$$y'_{\text{lo}} = y_{\text{lo}} - \epsilon_s \quad \text{if} \quad y_s < y_{\text{lo}}$$

y'_{hi} and y'_{lo} are the new upper and lower limits of the zone, respectively.

3.5 Regulator

Once the steady-state output and input values are known, the receding horizon regulator is used to calculate the input moves at each successive control interval with feedback incorporated by using plant measurements to update the state vector at each time step (see Muske and Rawlings [110] for more detail).

The SMPC control algorithm uses a single discrete-time state-space model in its receding horizon regulator to predict the dynamic plant behavior.

$$\begin{aligned} x(t+j+1) &= Ax(t+j) + Bu(t+j) \\ y(t+j) &= Cx(t+j) \end{aligned} \tag{3.3}$$

The current SMPC algorithm uses the following open-loop objective function in the receding horizon regulator to calculate the N future input moves at time t[2]

$$\min_{u^N} \quad \sum_{j=0}^{\infty} y^T(t+j)\bar{Q}y(t+j) + u^T(t+j)Ru(t+j) \tag{3.4}$$

subject to Equation 3.3 and the constraints

$$\begin{aligned} u_{\min} &\leq u(t+j) \leq u_{\max} \\ y_{\min} &\leq y(t+j) \leq y_{\max}. \end{aligned} \tag{3.5}$$

$\bar{Q}$ is a symmetric positive semidefinite penalty matrix on the outputs $y(t+j)$. R is a symmetric positive definite penalty matrix on the inputs $u(t+j)$, the input vector at time j in the open-loop objective function. The vector u^N contains the

[2] All variables in the regulator are deviation variables.

N future open-loop control moves.

$$u^N = \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N-1) \end{bmatrix} \tag{3.6}$$

For all $j \geq N$, the input vector $u(t+j)$ is calculated by the following linear feedback control law.

$$u(t+j) = Kx(t+j) \quad \text{for} \quad j \geq N \tag{3.7}$$

K is the infinite horizon quadratic regulator gain matrix and is calculated by the Riccati equation

$$\Pi_z = C^TQC + A^T\Pi_zA - A^T\Pi_zB(R + B^T\Pi_zB)^{-1}B^T\Pi_zA \tag{3.8}$$

$$K = -(B^T\Pi_zB + R)^{-1}B^T\Pi_zA. \tag{3.9}$$

In the modified SMPC algorithm, where the outputs have both zone regions and set points as their control objectives, the penalty for output deviations from the steady-state values $y^T(t+j)Qy(t+j)$ are replaced by $\epsilon^T(t+j)S\epsilon(t+j)$, the penalty for output deviations from the zone limits. The open-loop objective function in the regulator changes to the following objective function.

$$\min_{u^N,\epsilon^N} \quad \sum_{j=0}^{\infty} u^T(t+j)Ru(t+j) + \epsilon^T(t+j)S\epsilon(t+j) \tag{3.10}$$

subject to Equations 3.3 and 3.5 and the following zone constraints:

$$y(t+j) \le y_{\mathrm{hi}} + \epsilon(t+j)$$

$$y(t+j) \ge y_{\mathrm{lo}} - \epsilon(t+j)$$

$$\epsilon(t+j) \ge 0.$$

S is a symmetric positive semidefinite matrix that penalizes deviations from the zone limits. The vector ϵ^N contains the zone limit deviations from time t to time $t+N$.

$$\epsilon^N = \begin{bmatrix} \epsilon(t) \\ \epsilon(t+1) \\ \vdots \\ \epsilon(t+N) \end{bmatrix} \tag{3.11}$$

Once again, the linear feedback control law (Equation 3.7) is used to calculate $u(t+j)$ for all $j \ge N$.

Once the open-loop objective function has been optimized, the first input value in u^N, $u^*(t)$ is injected into the plant. This procedure is repeated at each successive control interval with feedback incorporated by using the plant measurements to update the state vector at time t.

The infinite horizon open-loop objective function in Equation 3.10 can be expressed as the finite horizon open-loop objective shown below.

$$\min_{u^N,\epsilon^N} \sum_{j=0}^{N-1} (u^T(t+j)Ru(t+j) + \epsilon^T(t+j)S\epsilon(t+j)) + x^T(t+N)Qx(t+N) \tag{3.12}$$

For a stable system, $\mathcal{Q}$ can be determined by the solution of the discrete Lyapunov function (Equation 2.9). Muske and Rawlings [110] show methods for determining $\mathcal{Q}$ for both stable and unstable systems.

3.6 State Estimator

In a state-space model, the states often have no physical meaning and are not measurable. Even if the states are physically meaningful, not all states can be measured by sensors. In these cases, output measurements are used with an observer that reconstructs the states from the output measurements.

The standard linear observer model is

$$x(t+1) = Ax(t) + Bu(t) + G_w w(t) \tag{3.13}$$

$$y(t) = Cx(t) + v(t)$$

in which $w(t)$ and $v(t)$ are zero-mean, uncorrelated, normally distributed, stochastic variables that simulate sensor noises [110]. The optimal linear observer for this system in which $\hat{x}(t+1|t)$ is the state estimate at time $t+1$ given output measurements up to time t is

$$\hat{x}(t+1|t) = A\hat{x}(t|t-1) + Bu(t) + L(y(t) - C\hat{x}(t|t-1)). \tag{3.14}$$

The discrete Kalman filter gain L minimizes the mean-square error of the state estimate $\hat{x}(t+1|t)$. L is computed from the solution of the discrete filtering steady-state Riccati equation with Q_w and R_v as the covariance matrices for $w(t)$ and $v(t)$, respectively (Equation 2.12). The observer optimally reconstructs the states from the output measurements given the noise assumptions $w(t)$ and $v(t)$.

3.7 Plant-Model Mismatch

The term nominal plant refers to the case in which the plant is exactly equal to the regulator model. Process identification is difficult because of process complexity. Most regulator models do not model the plant perfectly, which leads to plant-model mismatch. The difference between the regulator model and the actual plant can cause steady-state offset or closed-loop instability. Integral control is used to eliminate the steady-state offset.

Integral control is incorporated into the SMPC control algorithm by adding either an input disturbance $z(t)$ or an output disturbance $d(t)$ into the regulator model [129]. When the input disturbance model is used, the regulator model becomes

$$\begin{aligned} x(t+j+1) &= Ax(t+j) + B(u(t+j) + z(t+j)) \\ z(t+j+1) &= z(t+j) \\ y(t+j) &= Cx(t+j). \end{aligned} \tag{3.15}$$

When the integral controller uses the output disturbance model, the regulator model is

$$x(t+j+1) = Ax(t+j) + Bu(t+j) \tag{3.16}$$

$$d(t+j+1) = d(t+j)$$

$$y(t+j) = Cx(t+j) + d(t+j).$$

For the modified SMPC control algorithm with both zone regions and set points, the integral control is accomplished by using the output disturbance $d(t)$ in the regulator model.

The output disturbance $d(t)$ is not asymptotically stable and is also uncontrollable. However, they are observable. The estimates for $\hat{d}(t)$ are used to remove disturbances from the nominal system. The steady-state input and state vectors, u_s and x_s, are determined by the following quadratic program.

$$\min_{u_s, x_s, \epsilon_s} (u_s - u_t)^T R_s (u_s - u_t) + \epsilon_s^T S_s \epsilon_s \tag{3.17}$$

subject to

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_s - \hat{d}(t) \end{bmatrix}$$

$$y_s \leq y_{\text{hi}} + \epsilon_s$$

$$y_s \geq y_{\text{lo}} - \epsilon_s$$

$$u_{\min} \leq u_s \leq u_{\max}$$

$$y_{\min} \leq y_s \leq y_{\max}$$

$$\epsilon_s \geq 0$$

$\hat{d}(t)$ is the estimate of the output disturbance [110]. If $\hat{d}(t)$ is set to zero, then the above quadratic program reduces to Equation 3.2. The state estimator with integral control using $\hat{d}(t)$ is

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + L_x(y(t) - C\hat{x}(t) - \hat{d}(t)) \tag{3.18}$$
$$\hat{d}(t+1) = \hat{d}(t) + L_d(y(t) - C\hat{x}(t) - \hat{d}(t))$$

L_x and L_d are the filter gains for the state and disturbance, respectively [129].

The presence of plant-model mismatch not only causes steady-state offset but can also cause the system to become closed-loop unstable. The following set of equations is used to determine the closed-loop stability of a given system. For stability analysis, the input and output constraints as well as the zone regions are assumed to be feasible.

- Estimator

$$\hat{x}(t+1) = A_m\hat{x}(t) + B_m u(t) + L_x(y(t) - C_m\hat{x}(t) - \hat{d}(t)) \tag{3.19}$$
$$\hat{d}(t+1) = \hat{d}(t) + L_d(y(t) - C_m\hat{x}(t) - \hat{d}(t))$$

- Plant

$$x(t+1) = A_p x(t) + B_p u(t)$$
$$y(t) = C_p x(t)$$

- Regulator

$$u(t) = K(\hat{x}(t) - x_s) + u_s$$

A_m, B_m, and C_m form the regulator model; A_p, B_p, and C_p describe the real plant. K is the infinite horizon regulator gain matrix calculated from A_m and B_m. $\hat{x}(t+1)$ and $x(t+1)$ are the estimated and real state vectors, respectively. u_s and x_s are calculated from the following equations.

$$u_s = G_m^{-1}(y_t - \hat{d}(t)) \tag{3.20}$$

$$x_s = (I - A_m)^{-1} B_m G_m^{-1}(y_t - \hat{d}(t))$$

G_m exists for an (A_m, B_m) controllable, (A_m, C_m) observable, and a none integrating A_m, and G_m is equal to

$$G_m = C_m (I - A_m)^{-1} B_m. \tag{3.21}$$

After some algebraic manipulations, Equation 3.19 is written as an augmented closed-loop model,

$$\begin{bmatrix} \hat{x} \\ \hat{d} \\ x \end{bmatrix}_{t+1} = \tilde{A} \begin{bmatrix} \hat{x} \\ \hat{d} \\ x \end{bmatrix}_t + \tilde{B} y_t. \tag{3.22}$$

$\tilde{A}$ and $\tilde{B}$ are given below.

$$\tilde{A} = \begin{bmatrix} A_m + B_m K - L_x C_m & B_m K (I - A_m)^{-1} B_m G_m^{-1} - B_m G_m^{-1} - L_x & L_x C_p \\ -L_d C_m & I - L_d & L_d C_p \\ B_p K & B_p K (I - A_m)^{-1} B_m G_m^{-1} - B_p G_m^{-1} & A_p \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} -B_m K (I - A_m)^{-1} B_m G_m^{-1} + B_m G_m^{-1} \\ 0 \\ -B_p K (I - A_m)^{-1} B_m G_m^{-1} + B_p G_m^{-1} \end{bmatrix}$$

The system's closed-loop stability is determined by the eigenvalues of the augmented matrix $\tilde{A}$. If any of the absolute value of the eigenvalues of $\tilde{A}$ is greater than or equal to one, the system is closed-loop unstable; otherwise, it is closed-loop stable.

3.8 Ill-Conditioned Distillation Column

The modified SMPC algorithm is first applied to an ill-conditioned distillation column to demonstrate the improvements in control performance provided by the zone regions. An ill-conditioned plant is difficult to control because the plant gain is strongly dependent on the input direction, or the plant has a high condition number. The condition number is the most reliable indicator of ill-conditioning. It is equal to the ratio of the largest to the smallest singular value of the gain matrix G_m. Inputs in directions corresponding to high plant gains are strongly

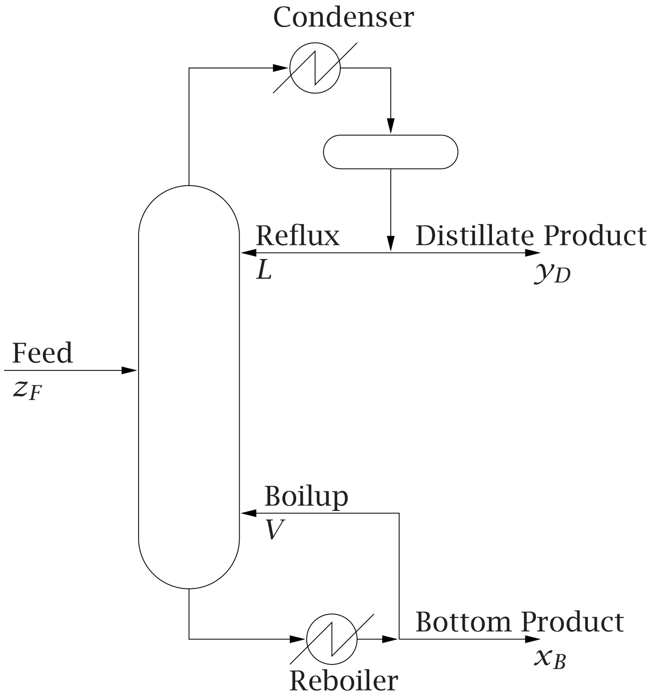

Figure 3.3: Ill-conditioned distillation column

amplified by the plant, while inputs in the directions corresponding to low plant gains are not.

The objective of the distillation column (Figure 3.3) is to separate the feed F, which is a mixture of a light and a heavy component, into a distillate product D and a bottom product B. The distillate product contains most of the light component, and the bottom product contains most of the heavy component. The feed has a light component composition of z_F. D and B have light component compositions of y_D and x_B, respectively.

The driving force for the separation is the difference between the volatilities of the light and heavy components. Skogestad et. al. [145] and Morari and Zafiriou [103] presented the following linear model for the ill-conditioned distillation column.

$$\begin{bmatrix} y_D \\ x_B \end{bmatrix} = G_{LV} \begin{bmatrix} L \\ V \end{bmatrix} \qquad G_{LV}(s) = \frac{1}{75s+1} \begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix} \tag{3.23}$$

The manipulated variables are L and V, the reflux and boil-up flow rates, respectively. The controlled variables are y_D and x_B. This linear two-point distillation column has a condition number of 142. The model is ill-conditioned which implies controlling both y_D and x_B independently is difficult. There is strong interaction between y_D and x_B.

The traditional method for controlling an ill-conditioned process with the same number of manipulated inputs as controlled outputs is to delete one or more of the controlled variables from the control objective [120, 57]. The disadvantage is the resulting steady-state offset of the controlled outputs that have been deleted. If the modified SMPC control algorithm with zone regions and set points is used instead of the conventional SMPC algorithm, one of the set points can be changed into a zone region so that all controlled outputs remain in the control objective.

The controlled performance of the two-point, ill-conditioned distillation column using the conventional and modified SMPC control algorithms are compared. The output target is $y_D = 0.975$ and $x_B = 0.01$ or

$$y_t = \begin{bmatrix} 0.975 \\ 0.01 \end{bmatrix}.$$

The input target u_t is the set of values that take the system to the output target y_t at steady state. When the modified SMPC algorithm with zone regions is used, the set point for y_D is changed into a zone with 0.95 and 1.00 as the lower and upper limits, respectively.

When no input or output constraints are present, both the conventional and modified SMPC algorithm take the system from an initial output condition of $y_D = 0.1$ and $x_B = 0.9$ to the output target. The only difference between the two is the system dynamics (see Figure 3.4). When the control objective for y_D is a zone region instead of a set point, the penalty $(y(t) - y_s)$ for x_B's deviation from the set point is greater than the penalty $\epsilon(t)$ for y_D's deviation from the zone limits because the lower limit of y_D's zone is smaller than y_D's target and because y_D is on an increasing trend as it approaches the zone. Therefore, x_B approaches its set point faster when y_D has a zone instead of a set point. The trade-off in this instance is y_D's slower approach to its target.

In the next set of simulations, the only constraint present is V's maximum value which is set equal to 37.5. In this case, there is significant difference between the control performance of the conventional and the modified SMPC algorithms when the output initial conditions are $y_D = 0.1$ and $x_B = 0.9$. When both y_D and x_B have set points, there is steady-state offset for both outputs from their target values because of the input constraints. When the set point for y_D is changed into a zone region, both x_B's set point and y_D's zone region

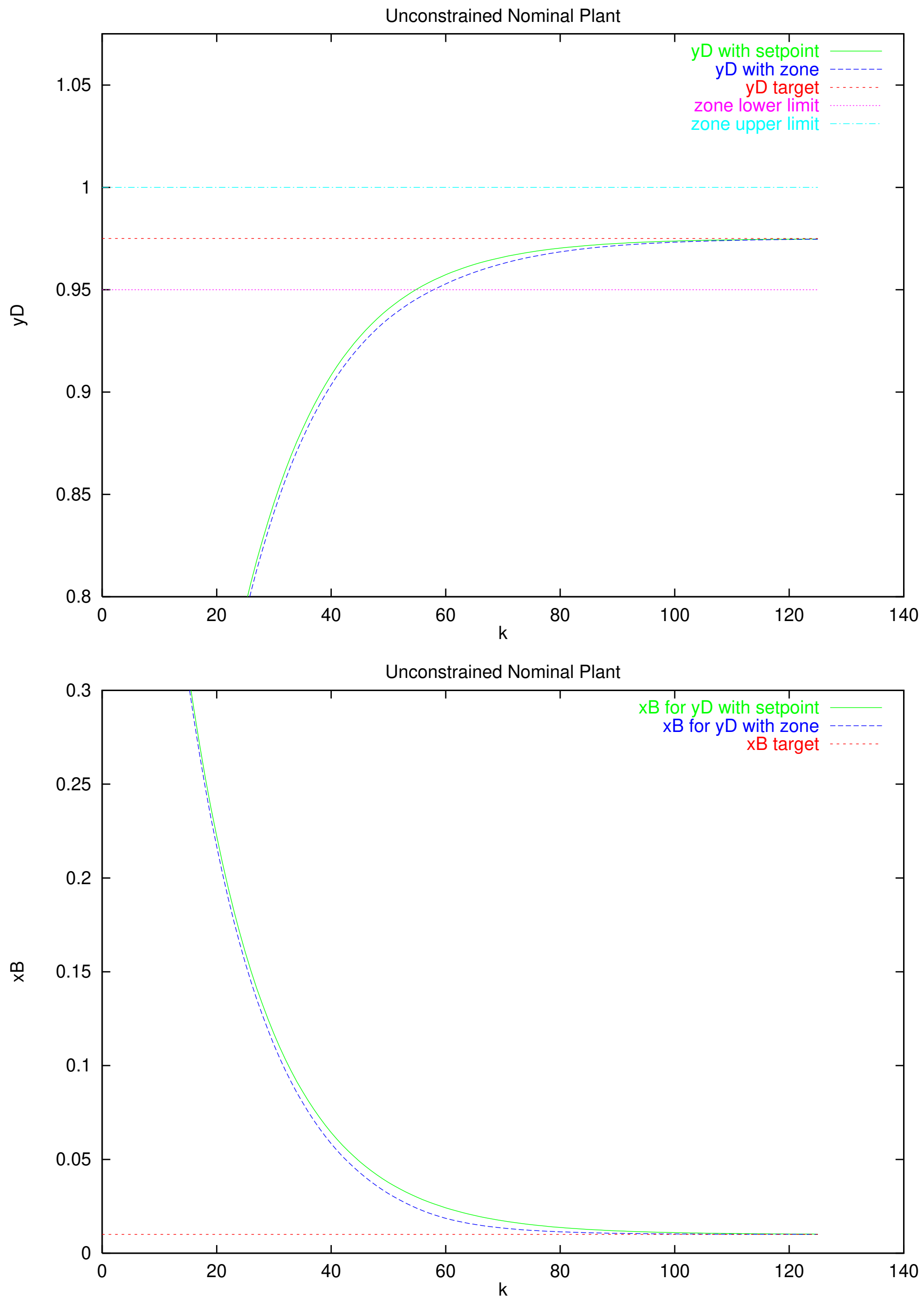

Figure 3.4: Composition for the unconstrained nominal plant with set point change

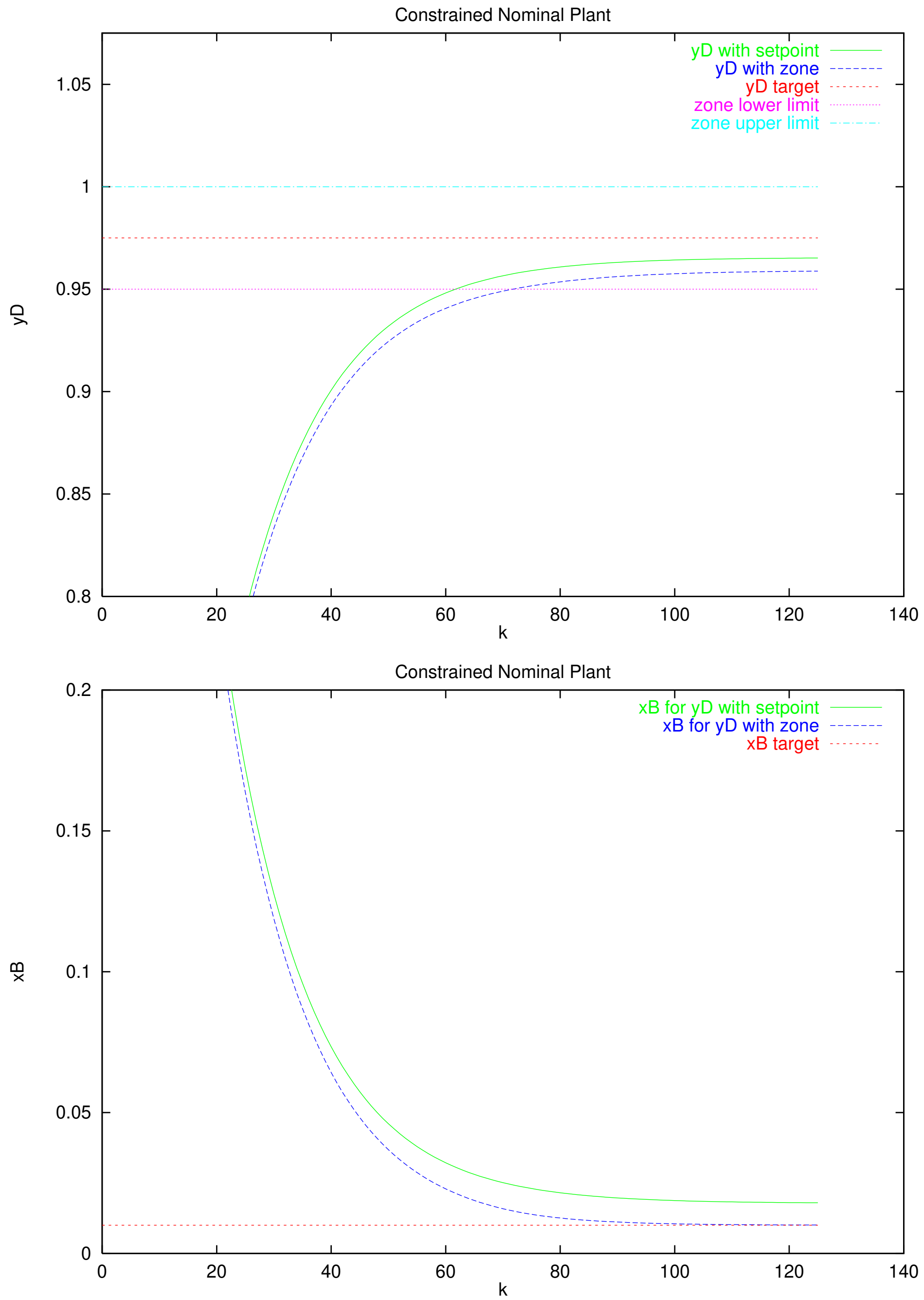

Figure 3.5: Composition for the constrained nominal plant with set point change

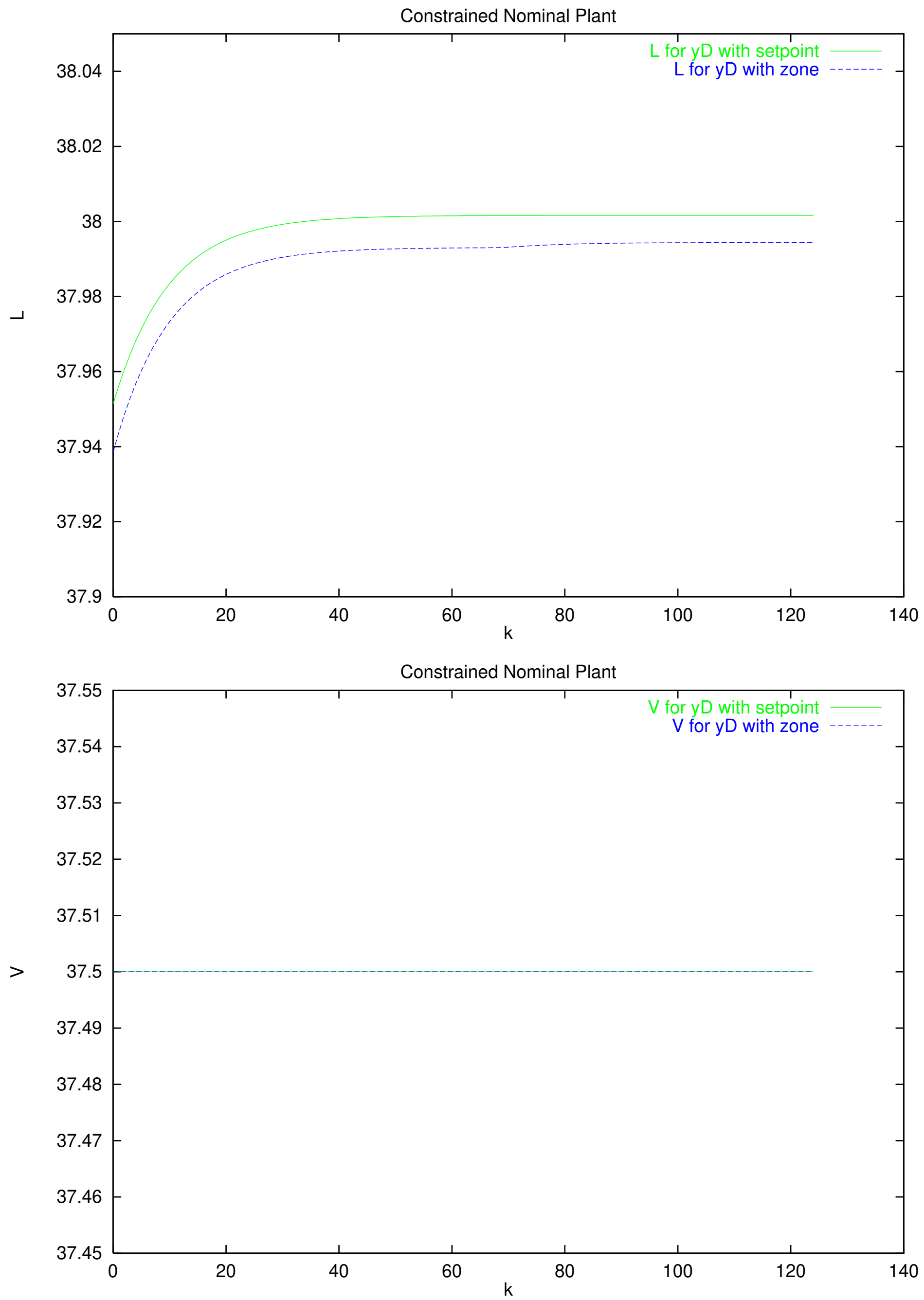

Figure 3.6: Reflux and boil-up flow rate for the constrained nominal plant with set point change

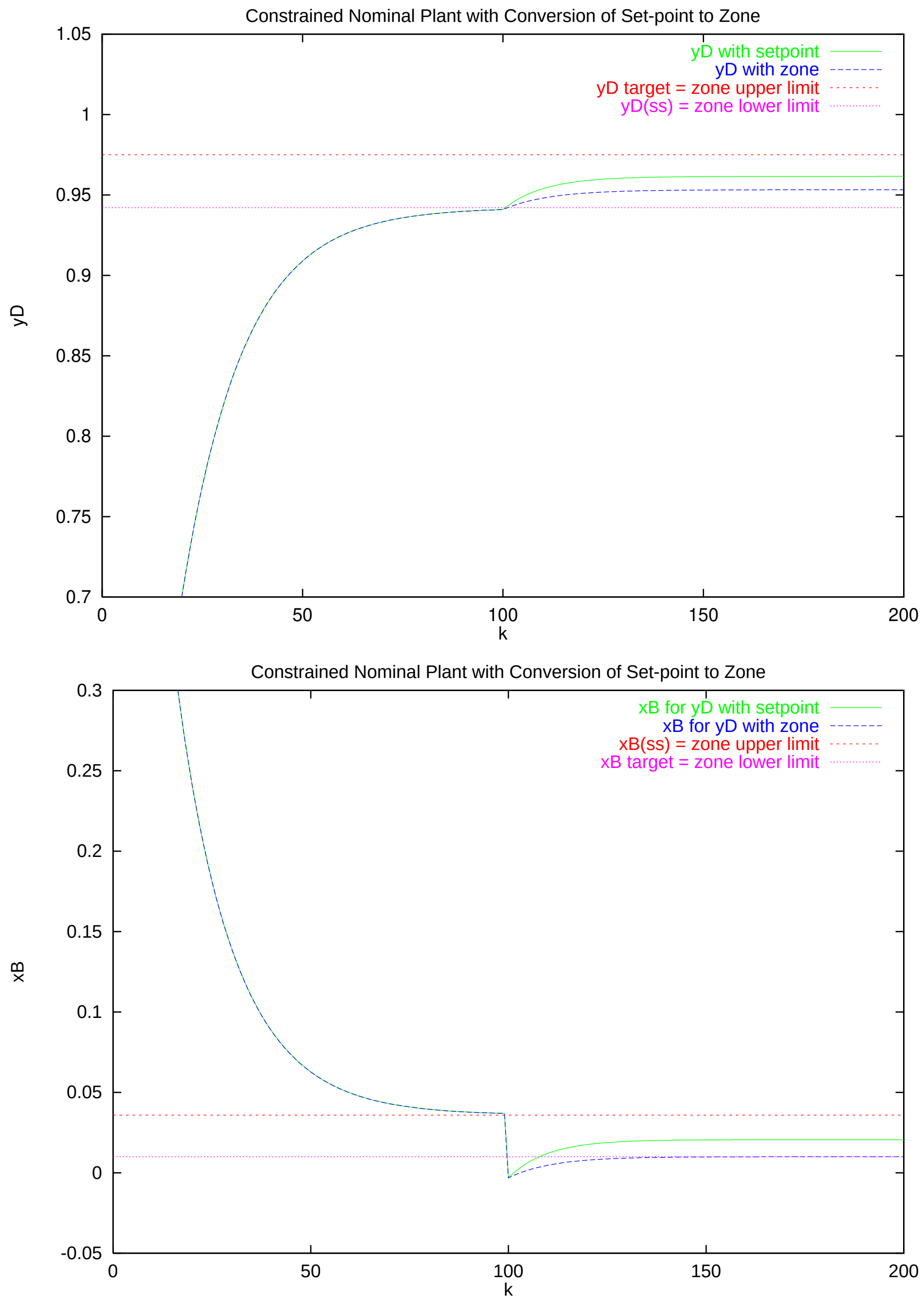

Figure 3.7: Composition for the constrained nominal plant with conversion of the set point to a zone

control objectives can be satisfied with no offset (see Figure 3.5). The trade-off for the elimination of offset from x_B's set point is a decrease in the steady-state y_D value, but because the lower y_D value is within the zone limits, both x_B's set point and y_D's zone control objectives are met satisfactorily. Figure 3.6 shows the corresponding input moves. Please note that V is equal to 37.5, the maximum value, at all times for both SMPC algorithms.

In the next set of nominal plant simulations, the initial control objectives are set points for both y_D and x_B. The maximum value of the reflux flow rate L is equal to 36.5. The input constraint causes steady-state offset for both y_D and x_B. At time $t = 100$, a x_B disturbance of -0.04 enters the system. This disturbance causes x_B to move closer to its set point. If both y_D and x_B have set points, the disturbance causes both y_D and x_B to move closer to their targets, but neither of the targets is met completely. If the set points for y_D and x_B are changed into zone regions, the relaxations in the control objectives allow x_B to go to its target at steady state (see Figure 3.7). When y_D and x_B are inside the zone, the input target values determine y_D and x_B's unique values inside the zone region. The zone limits for y_D and x_B are defined as the following.

- y_D's lower limit = steady-state y_D value with constraints.
- y_D's upper limit = y_D's target
- x_B's lower limit = x_B's target

- x_B's upper limit = steady-state x_B value with constraints.

Recall Equation 3.23 is the linear model used in the regulator. If the real plant is modeled by the following equation, the −0.036 difference between the (1, 2) element of the regulator gain matrix and the actual plant gain matrix causes the system to be closed-loop unstable. (One of the eigenvalues of $\tilde{A}$ is greater than one.)

$$\begin{bmatrix} y_D \\ x_B \end{bmatrix} = G_{LV} \begin{bmatrix} L \\ V \end{bmatrix} \qquad G_{LV}(s) = \frac{1}{75s+1} \begin{bmatrix} 0.878 & -0.9 \\ 1.082 & -1.096 \end{bmatrix} \tag{3.24}$$

In Figure 3.8, the system is initially at the target values of $y_D = 0.975$ and $x_B = 0.01$. At time $t = 20$, a step disturbance of −0.02 enters the y_D output variable. If the conventional SMPC algorithm is used to control the disturbance, the system becomes unstable because of the plant-model mismatch. When y_D's control objective is changed from a set point to a zone region while x_B's control objective remains as a set point, the −0.02 disturbance does not cause y_D to deviate from its zone limits, and x_B remains at its target value. No control actions are taken because none of the control objectives have been violated. The system maintains its stability even though the closed-loop stability analysis says the system is closed-loop unstable.

Figure 3.8 shows that if no control actions are needed when there is a disturbance, the system remains stable. Figure 3.9 shows that when the disturbance causes y_D to deviate from its zone region, the need for control actions to bring

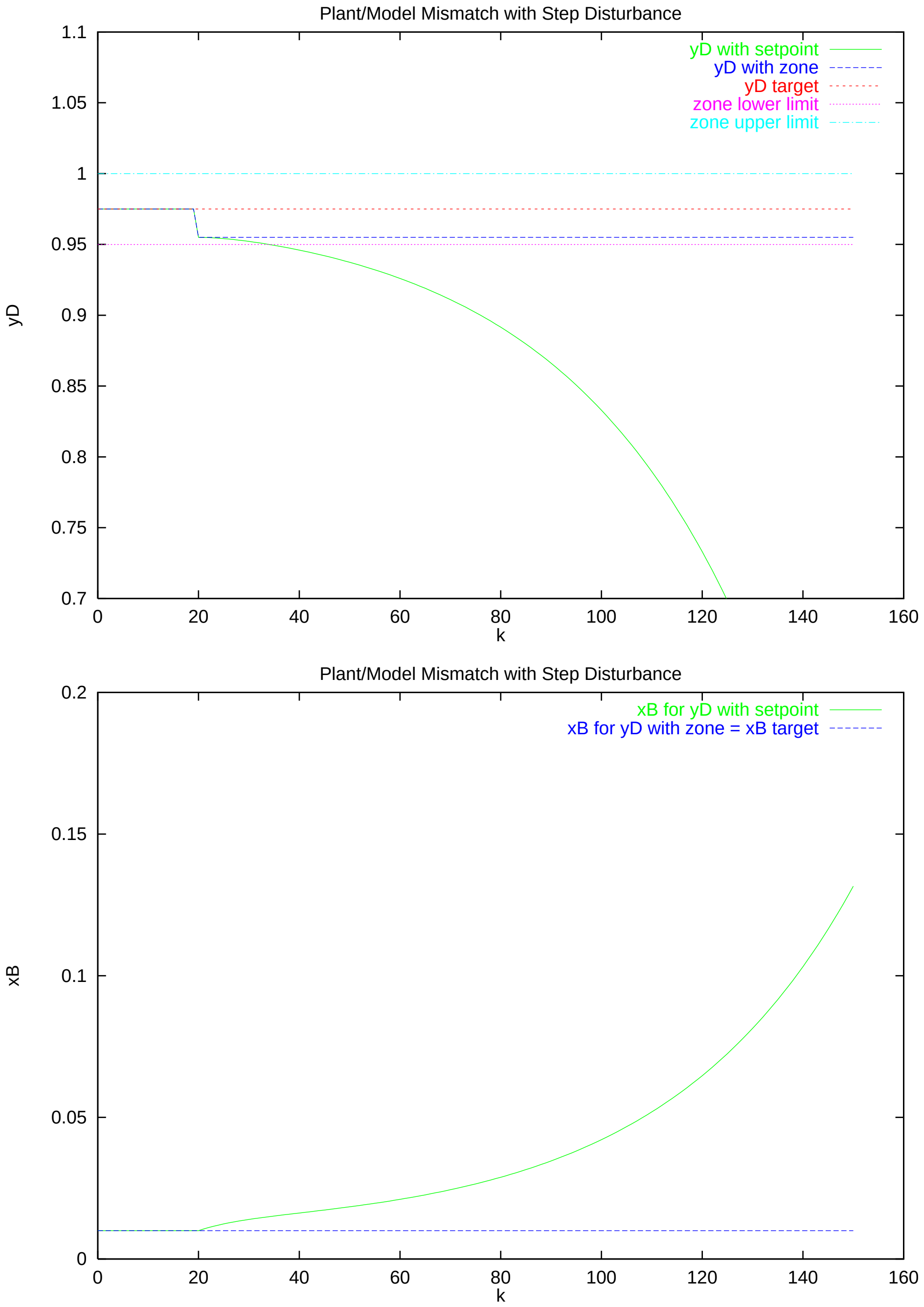

Figure 3.8: Composition for process with step disturbance and plant-model mismatch

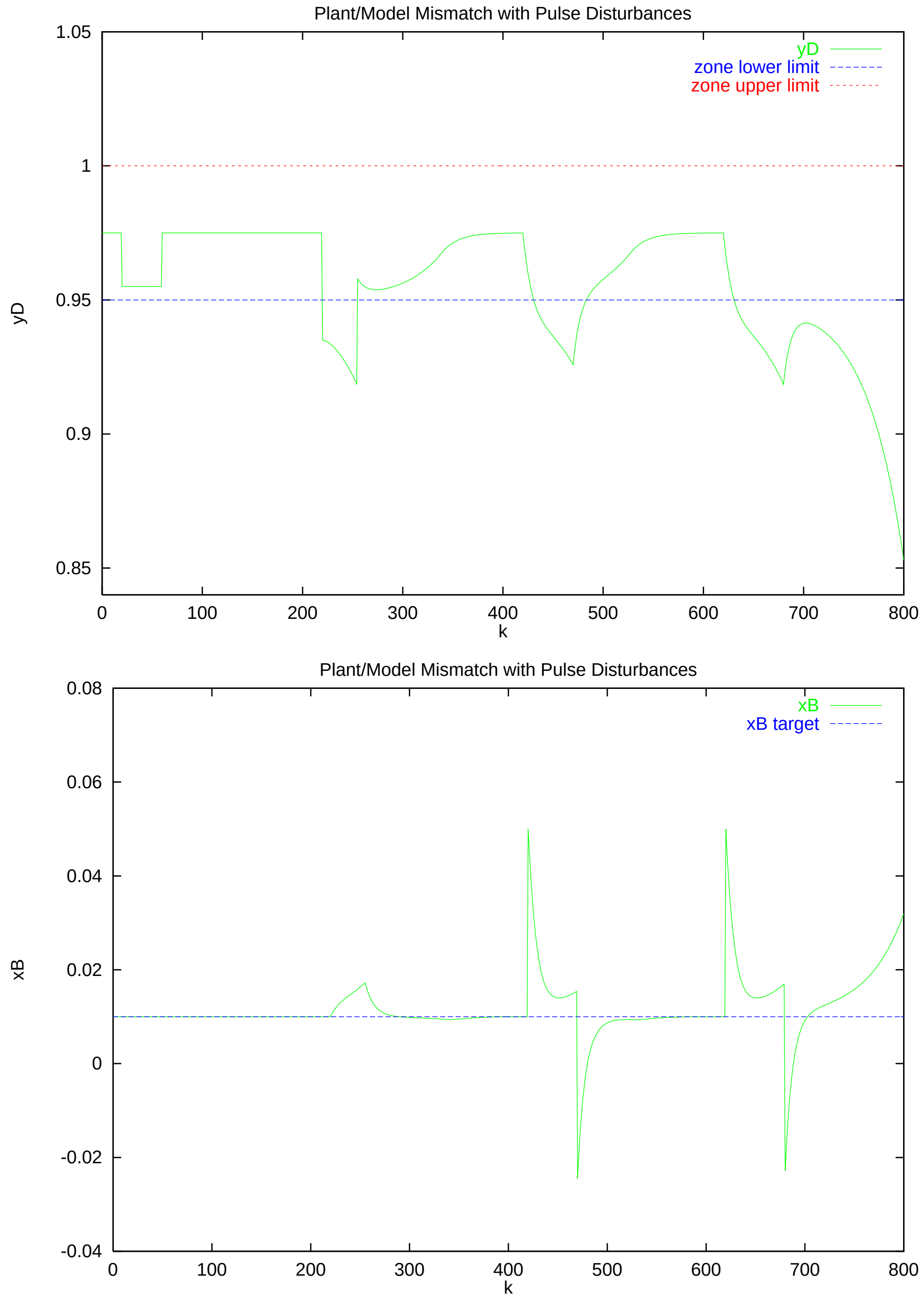

Figure 3.9: Composition for process with pulse disturbances and plant-model mismatch

Disturbance Variable	Disturbance Magnitude	Start Time	Pulse Duration
y_D	-0.02	$t = 20$	45
y_D	-0.04	$t = 220$	35
x_B	0.04	$t = 420$	50
x_B	0.04	$t = 620$	60

Table 3.1: Pulse Disturbance Descriptions

y_D back inside the zone region causes the system to become unstable. But if the step disturbances are changed into square pulse disturbances, the system remains stable if the second half of the square pulse disturbance takes y_D back inside its zone region. The system maintains its stability not only when disturbances are present in y_D but also when disturbances are present in x_B. For the last square pulse disturbance found in Figure 3.9, the second half of the square pulse disturbance is not large enough to bring y_D back inside the zone region. The system becomes unstable even though the modified MPC algorithm is used. (See Table 3.1 for descriptions of all the square pulse disturbances that are found in Figure 3.9).

Most measured output variables have sensor noise. When plant-model mismatch causes the system to be closed-loop unstable, small amounts of sensor noise cause the system to lose its stability when both y_D and x_B have set points as their control objectives. If y_D's set point is changed into a zone region, the system remains stable when the same amount of sensor noise is present. In Figure 3.10, sensor noise is present in x_B only. The system is able to accommodate the sensor

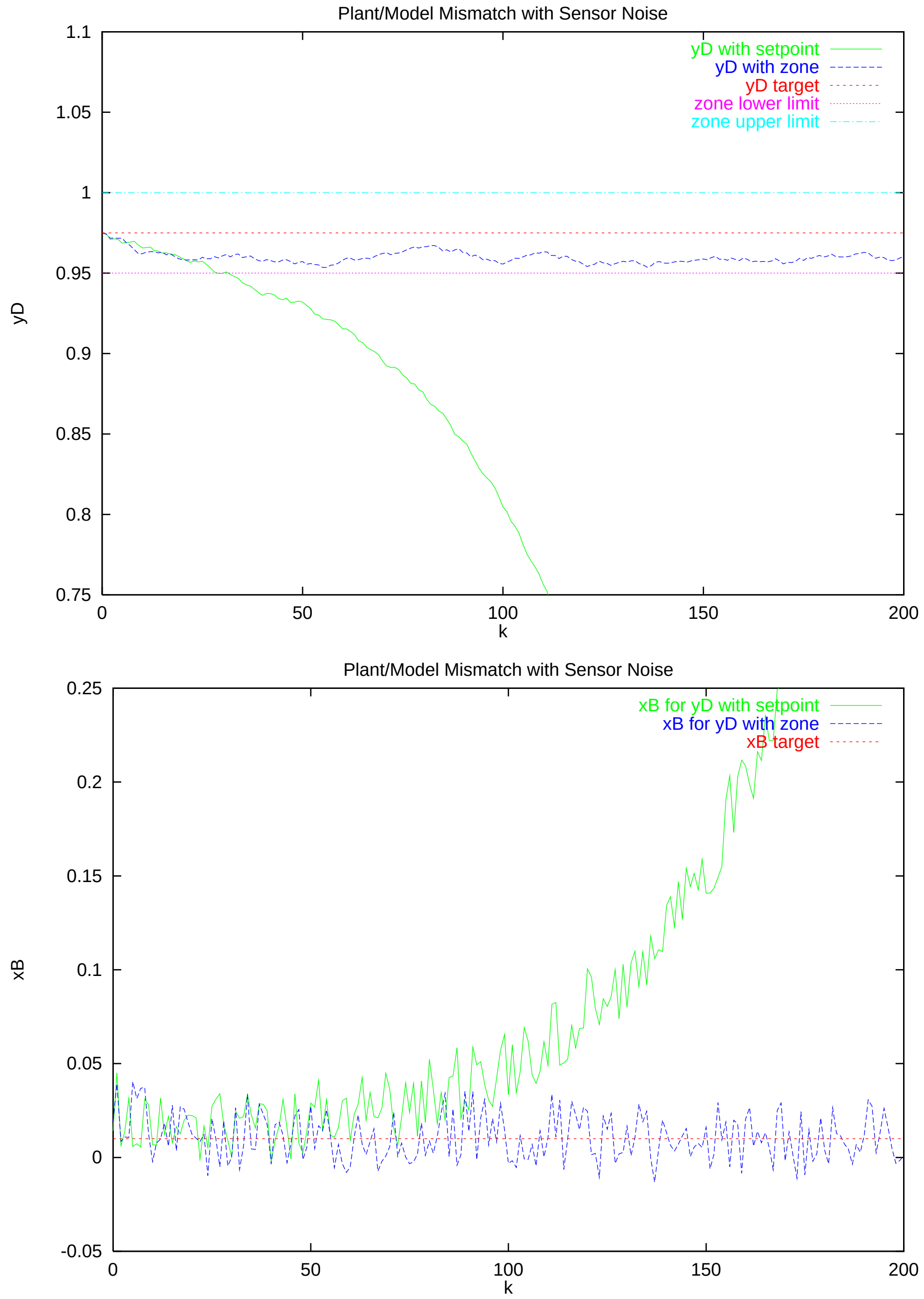

Figure 3.10: Composition for process with sensor noise and plant-model mismatch

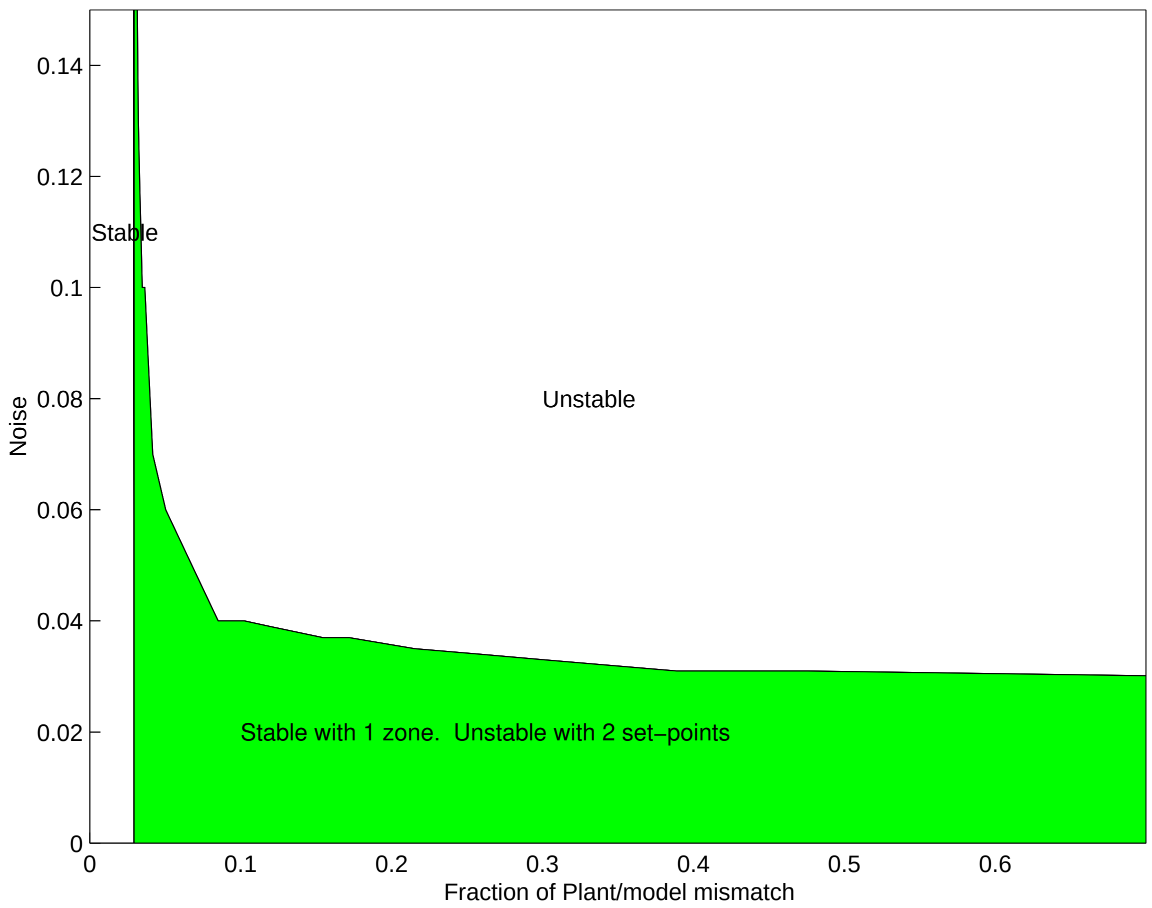

Figure 3.11: System's stability as a function of sensor noise and plant-model mismatch

noise and remains stable using the modified SMPC algorithm.

Figure 3.11 shows the system stability as a function of the amount of sensor noise and plant-model mismatch. The x-axis uses the fractional error between the $(1,2)$ element of the plant gain matrix and the regulator gain matrix to represent the amount of plant-model mismatch. The y-axis is the amount of sensor noise present in the system. When the plant-model mismatch has a fractional error less than 0.03, the system remains for any amount of sensor noise regardless of which MPC algorithm is used. When plant-model mismatch is greater than

0.03, any amount of sensor noise causes the system to become unstable when SMPC algorithm with set points only is used. However, if the SMPC algorithm with zone regions and set points is used instead, the system remains stable for small amounts of sensor noise. When the sensor noise is large, neither algorithms is able to control the system, and the system becomes unstable. Therefore, SMPC algorithm with zone regions and set points is more robust than MPC algorithm with set points only.

3.9 Conclusion

The modified SMPC algorithm with zone regions and set points in the control objectives is a more robust control algorithm compared to the conventional SMPC algorithm. The increase in robustness is demonstrated by the ill-conditioned distillation column example. The two-point distillation column is difficult to control because the outputs are highly correlated. Small process identification errors in the gain matrix cause the system to be closed-loop unstable. When the set point for one of the two outputs is changed into a zone region, the control algorithm's robustness increases. The control system is able to reject small disturbances and maintain closed-loop stability. The system is also able to accommodate sensor noise and maintain stability if one of the controlled outputs has a zone region instead of a set point as its control objective. When output disturbances enter the system, controllers with zones and set points maintain system stability better than controllers with only set points.

3.10 Appendix

The quadratic objective function (Equation 3.10) in the SMPC regulator can be solved by formulating the following quadratic program.

$$\min_{\pi} \quad \pi^T H \pi + 2h^T \pi \tag{3.25}$$

$$s.t. \quad L\pi \leq l$$

The vector π contains the N future input moves as well as the slack variables ϵ.

$$\pi = \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N-1) \\ \epsilon(t) \\ \epsilon(t+1) \\ \vdots \\ \epsilon(t+N) \end{bmatrix} \tag{3.26}$$

The matrices H, h, L, and l are computed as shown below, and $\mathcal{Q}$ is calculated by Equation 2.9

$$H = \begin{bmatrix} B^T {A^{N-1}}^T \mathcal{Q} A^{N-1} B + R & \cdots & B^T {A^{N-1}}^T \mathcal{Q} B & 0 & 0 & \cdots & 0 \\ B^T {A^{N-2}}^T \mathcal{Q} A^{N-1} B & \cdots & B^T {A^{N-2}}^T \mathcal{Q} B & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ B^T \mathcal{Q} A^{N-1} B & \cdots & B^T \mathcal{Q} B + R & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & S & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & S & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & S \end{bmatrix}$$

$$h = \begin{bmatrix} B^T {A^{N-1}}^T \mathcal{Q} A^N \\ B^T {A^{N-2}}^T \mathcal{Q} A^N \\ \vdots \\ B^T \mathcal{Q} A^N \\ 0 \\ \vdots \\ 0 \end{bmatrix} x(t)$$

$$L = \begin{bmatrix} -F & 0 & -I_e \\ F & -I_e 0 & \\ -I_u & 0 & 0 \\ I_u & 0 & 0 \\ -F & 0 & 0 \\ F & 0 & 0 \\ 0 & -I_e & 0 \\ 0 & 0 & -I_e \end{bmatrix} \quad l = \begin{bmatrix} -(y_{lo} - y_s) + fx(t) \\ (y_{hi} - y_s) - fx(t) \\ -(u_{min} - u_s) \\ (u_{max} - u_s) \\ -(y_{min} - y_s) + fx(t) \\ (y_{max} - y_s) - fx(t) \\ 0 \\ 0 \end{bmatrix}$$

The matrices F and f are given below.

$$F = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix} \quad f = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}$$

Chapter 4

Robust Control Literature Review

4.1 Introduction

In this chapter, we attempt to summarize the vast amount of research that has been published in the field of robust control. The main motivation for academic researchers has been development of robust control theories that can guarantee closed-loop stability in the presence of modeling errors. We begin the literature review with a discussion of the different model uncertainty descriptions, which can assist researchers in the controller design procedure. Next, we summarize the field of H_∞ control, which designs controllers that minimize the effect of the worst possible disturbance. Other robust control theories incorporate model uncertainty descriptions directly into the infinite horizon MPC design procedure. Finally, we discuss robust integral control for non-zero set point tracking with disturbance rejection.

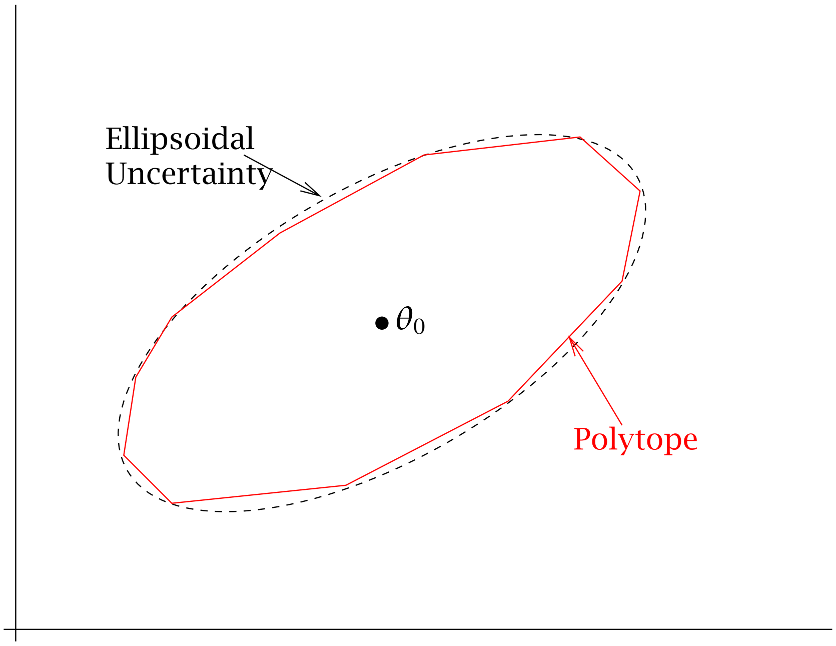

Figure 4.1: Parametric uncertainty description

4.2 Uncertainty Description

Researchers have used many different methods to characterize model uncertainty when studying robust control. The two common types of uncertainties are parametric uncertainty and bounded disturbance regions. Magni et al. [93] and Rami and Zhou [125] proposed using multiplicative input and state uncertainties. Parametric uncertainties are modeled as either bounded uncertainty regions or a polytope. The process model (A, B, C) can be expressed as functions of the parameter vector $\theta = [\theta_1, \cdots, \theta_{N_\theta}]^T \in \mathbb{R}^{N_\theta}$.

$$A = f_A(\theta), \qquad B = f_B(\theta), \qquad C = f_C(\theta)$$

If the process noise and model parameters are assumed to be normally distributed, the natural uncertainty description is an ellipsoidal bound on the parameters [10]. The parameter vector θ is assumed to lie in the set

$$\theta \in \Theta = \{\theta : (\theta - \theta_0)^T M (\theta - \theta_0) \leq 1\} \tag{4.1}$$

in which θ_0, the center of the ellipse, is the mean of the normal distribution, and the matrix $M = M^T > 0$ defines the size and the orientation of the ellipsoid. The square roots of the reciprocals of the eigenvalues of M are the lengths of the semi-axes of the ellipsoid, and the eigenvectors of M give the directions of the semi-axes. Ben-Tal and Nemirovski [13] provided a more extensive discussion of ellipsoidal uncertainty. They described ellipsoidal bounds in the general context of convex programming. Boyd et al. [19] motivated the use of ellipsoidal uncertainty descriptions for use in optimal control.

Ellipsoidal uncertainties can be approximated as convex polytopes whose bounds are formed by the linear combination of a class of process models [17]

$$\Pi = \{(A_1, B_1), \ldots, (A_{\mathcal{J}}, B_{\mathcal{J}})\}$$

Each model in Π can be expressed as functions of θ. Figure 4.1 is the graphical representation of the polytope in the θ space. In the state space, the polytope is defined by $\Omega = \mathrm{Co}\{(\mathrm{A}_1, \mathrm{B}_1), (\mathrm{A}_2, \mathrm{B}_2), \ldots, (\mathrm{A}_{\mathcal{J}}, \mathrm{B}_{\mathcal{J}})\}$ in which Co denotes the convex hull. The model (A, B) belongs to Ω if

$$A = \sum_{i=1}^{\mathcal{J}} \mu_i A_i, \qquad B = \sum_{i=1}^{\mathcal{J}} \mu_i B_i, \qquad \sum_{i=1}^{\mathcal{J}} \mu_i = 1, \qquad \mu_i \geq 0 \ \forall i = 1, \ldots, \mathcal{J}$$

Polytopic system models usually arise when the uncertain plant is modeled as a linear time-varying (or parameter-varying) system with state-space matrices.

In recent years, there has been a lot of renewed interest in linear parameter-varying systems, especially to provide useful theoretical foundations for gain scheduling control [4, 135, 142, 155]. Linear parameter-varying systems can be used to approximate nonlinear processes. Murray-Smith and Johansen [107] provides a summary of the current research in the use of multi-model paradigm for nonlinear process control. The technique of global linearization [21, 89, 101] suggests that such an approximation is possible, although it can be conservative. The family of linear models used to construct Ω is determined by linearizing the nonlinear process model at different operating points. Control actions for the nonlinear plant are determined by robustly stabilizing control moves for the polytopic system.

Besides polytopes, ellipsoidal uncertainties can be approximated as simple bounded uncertainty regions.

$$A = A_0 + \Delta A \quad B = B_0 + \Delta B \tag{4.2}$$

$$\Delta A \in \Omega_A \quad \Delta B \in \Omega_B$$

in which Ω_A and Ω_B are the bounds for the uncertainty regions of A and B, respectively. The "nominal model" is given by (A_0, B_0) and is equivalent to θ_0 in the ellipsoidal uncertainty description (Equation 4.1).

Model uncertainty can also be described by bounded exogenous disturbances. The disturbance is usually placed in the system state. In continuous-time state space,

$$\dot{x} = A(x) + B(x)u(x) + C(x)f(x) \tag{4.3}$$

in which $f(x)$ characterizes the model uncertainty as a state disturbance and is bounded. If $C(x) = B(x)$, uncertainty is considered to be matched. Uncertainties satisfy the matching condition only if they can be completely compensated by the control actions [86]. If $C(x) \neq B(x)$, the uncertainty can be separated into matched and unmatched portions. Corless and Manela [33], Hollot and Arabacioglu [60], Magana and Zak [92], Bahnasawi et al. [9], Farison and Kolla [46], and Wang and Ghosh [153] developed some of the robust control design methods for discrete-time systems with matched uncertainties. Lin and Olbrot [87] and Lin and Zhang [88] showed methods for finding a linear state feedback gain that robustly stabilizes both matched and unmatched disturbances.

Separating disturbance perturbations into matched and unmatched disturbances is one method of controlling uncertain systems with exogenous disturbances. Other methods are H_∞ control and model predictive control. Any knowledge about the characteristics of the disturbance is useful in determining robustly stabilizing control actions. For example, Sideris and Tchernychev [144] used the rate of uncertainty variation to determine a robustly stabilizing output feedback gain for a time-varying uncertain system.

4.3 H_∞ Control

H_∞ optimization of control systems began in 1979 with a conference paper by Zames [156], who considered the minimization of the infinity-norm for the sensitivity function (Equation 4.4) of a single-input-single-output (SISO) linear feedback system. The work dealt with some of the basic questions of "classical" control theory. It was soon extended to more general problems, especially when it was recognized that the approach allows dealing with robustness far more directly than other optimization methods.

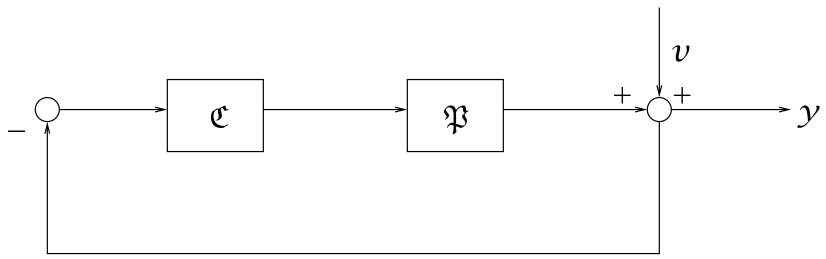

Figure 4.2: SISO feedback loop

Kwakernaak [74] presents a tutorial on H_∞ control. To summarize, consider the system described by Figure 4.2 in which signal v represents a disturbance acting on the system and y is the controlled output. The transfer function describing the relationship between v and y is $y(s) = \mathfrak{S}v(s)$, where

$$\mathfrak{S} = \frac{1}{1 + \mathfrak{P}\mathfrak{C}}. \tag{4.4}$$

$\mathfrak{S}$ is known as the sensitivity function for the feedback system in Figure 4.2. As the name implies, the sensitivity function characterizes the sensitivity of the control system output to disturbances.

H_∞ optimization was originally used by Zames [156, 157] to find a compensator $\mathfrak{C}$ that stabilizes the closed-loop system and minimizes the peak value of the sensitivity function. The peak value is defined as

$$\| \mathfrak{S} \|_\infty = \max_{\omega \in \mathbb{R}} |\mathfrak{S}(j\omega)|. \tag{4.5}$$

The justification is that if the peak value $\| \mathfrak{S} \|_\infty$ of the sensitivity function $\mathfrak{S}$ is small, then the magnitude of $\mathfrak{S}$ is small for all frequencies, so that disturbances are attenuated over all frequencies. Minimization of $\| \mathfrak{S} \|_\infty$ is the worst-case optimization, because it amounts to minimizing the effect on the output of the worst disturbance. Given the transfer function $\mathfrak{L} = \mathfrak{P}\mathfrak{C}$, model uncertainty is characterized by perturbations from the nominal value $\mathfrak{L}_0$ to the actual value $\mathfrak{L}$. The factor

$$\frac{|\mathfrak{L}(j\omega) - \mathfrak{L}_0(j\omega)|}{|\mathfrak{L}_0(j\omega)|} \tag{4.6}$$

is the relative size of the perturbation of $\mathfrak{L}$ from its nominal value $\mathfrak{L}_0$. Suppose this relative perturbation as a function of frequency is known to be bounded by

$$\frac{|\mathfrak{L}(j\omega) - \mathfrak{L}_0(j\omega)|}{|\mathfrak{L}_0(j\omega)|} \le |\mathfrak{W}(j\omega)| \quad \forall \omega \in \mathbb{R} \tag{4.7}$$

in which $\mathfrak{W}$ is a frequency dependent function. $\mathfrak{T}_0$ is the nominal closed-loop complementary sensitivity function for the SISO system.

$$\mathfrak{T}_0 = \frac{\mathfrak{L}_0}{1 + \mathfrak{L}_0} \tag{4.8}$$

The system is robustly stable if

$$\| \mathfrak{W}\mathfrak{T}_0 \|_\infty < 1 \tag{4.9}$$

for all bounded perturbations. The peak value criterion in Equation 4.9 arises from the Nyquist stability criterion.

A similar stability condition is available for systems with exogenous disturbances as the uncertainty description. The basis of H_∞ is to find a compensator such that the infinity-norm is less than a maximal acceptable value for all bounded disturbances or parametric uncertainties. Worst-case optimization suggests a game theory paradigm. The designer wishes to choose a compensator $\mathfrak{C}$ so that it offers the best protection against the worst disturbance, which explains why some researchers treat H_∞ optimization from the point of view of the differential game theory [26].

Although H_∞ control theory was first developed using transfer functions as process models, researchers have applied the H_∞ concepts to both continuous-time and discrete-time state-space models. Wang et al. [154] and Cao et al. [23] designed H_∞ controllers for continuous-time state-space models with bounded parametric uncertainty in (A, B). The goal of the H_∞ controller is to find the linear state feedback gain that stabilizes all bounded uncertainties by satisfying the infinity-norm constraint (Equation 4.9). De Souza and Shaked [43] proved robust stability for a polytopic system with continuous state-space models. Shaked et al. [141] used H_∞ theory to find the robustly stabilizing state feedback gain K for the polytope of discrete-time process models. Mahmoud [94] proposed robust H_∞ control for linear neural system.

H_∞ theory provides a theoretical framework for dealing with nominal stability and robust stability issues. Disturbance attenuation specifications can also be incorporated naturally into the design. However, standard H_∞ control has several shortcomings [95]:

1. the computational burden of the infinite horizon formulation makes an application to realistic problems virtually impossible.

2. the hard input and output constraints cannot be added to the controller design procedure in a straightforward manner.

4.4 Robust Predictive Control

Another large class of robust control theory uses the general concepts of infinite horizon predictive control. The structure of the model uncertainty description is used explicitly in the controller design to determine robustly stabilizing control policies. Robustly stabilizing controllers were first developed for linear unconstrained systems. Konstantopoulos and Antsaklis [68] and Farison and Kolla [47] used the solution of the linear quadratic regulator [75] to determine a full-state feedback control policy for the nominal model. Robust control was accomplished by deriving bounds on parametric uncertainties that could be stabilized by the nominal feedback gain. Kouvaritakis et al. [72, 73] and Lee and Kouvaritakis [85] proposed using a linear state feedback law with an extra design parameter, $u = Kx + c$, to stabilize $(A, B) \in \Omega$. In addition to adding the extra design parameter c to the control policy, Chisci et al. [28] discussed output feedback by the addition of state estimation using a nominal model, but they failed to discuss the effect the nominal model has on the control performance.

For constrained uncertain systems, researchers started proving robust stability by examining the nominal infinite horizon problem formulation and stability arguments. De Nicolao et al. [41] showed that the stability criterion requiring decreasing performance objective for the single-model nominal MPC control theory was sufficiently large to maintain closed-loop stability in the presence of small modeling errors. Michalska and Mayne [102] increased the robustness of their optimal controller by decreasing the size of the invariant region W. As the size of the invariant region decreases, the controller can stabilize systems with larger margins of error between the actual and the predicted trajectories. De Nicolao et al. [42] determined a linear state feedback gain that guaranteed a decreasing performance objective for all states in an invariant region.

Badgwell [6, 7, 8], Biegler [15], Chisci et al. [29], and Slupphaug and Foss [146] used a nominal model to determine the optimal control actions for both linear and nonlinear systems. An additional constraint was added to the optimization problem to guarantee decreasing performance objectives at each time step for all parametric uncertainties. Ralhan and Badgwell [124] used a step disturbance model in addition to the decreasing performance objective constraint to reject disturbances in the presence of model uncertainty. The main motivation for using a nominal model in robust control theory is that closed-loop performance may be less conservative and more optimal if the nominal model happens to be the model that accurately describes the plant. On the other hand, closed-loop performance may be poor if the plant differs significantly from the nominal model.

Pannocchia and Semino [116, 140] increased controller robustness by altering off-line the nominal model in such a way that it could tolerate the maximum amount of uncertainties and produce the best possible closed-loop control performance. Zheng [158] developed stability conditions for systems that were described by two possible models with regions of uncertainty around each model. Genceli and Nikolaou [52] provided stability conditions for nonlinear processes modeled by second order Volterra models with bounded disturbances.

Predictive controllers that explicitly consider the process model uncertainty when determining the optimal control policies are called robust predictive controllers. Researchers propose using the H_∞ concept of minimizing the "worst" disturbance affecting the process. Optimal control actions come from the solution of a *min-max* optimization problem that minimizes the "worst" performance objective for the given uncertainty descriptions. Campo and Morari [22], Prett and Garcia [117], Allwright and Papavasiliou [3], Zheng and Morari [159] proposed using an impulse response model with uncertain coefficients to calculate the set of possible output predictions. The impulse response models have large numbers of uncertain parameters, which increases the overall computational efforts of the control law and the complexity of the robust identification. Lee et al. [78] and Lee and Morari [79] reduced the complexity of the process models to the discrete-time state-space models. Oliveira et al. [113] used the orthonormal series functions.

Researchers prove robust stability for receding horizon control by using the closed-loop stability criteria that are similar to the criteria used to prove stability for the nominal MPC, i.e., terminal cost, terminal stability constraint and local control policy. Lee and Cooley [81, 82] and Lee and Yu [83] determined optimal control actions via an infinite-horizon *min-max* optimization problem. Stability was guaranteed for open-loop stable models by adding the constraint $u(t+N) = 0$. Mayne and Schroeder [97], Lee and Kouvaritakis [84], and Scokaert and Mayne [137] proposed a *min-max* control law that steered the state into an invariant region in which a state feedback law guaranteed convergence to the origin for all states.

Lee and Yu [83] showed that when uncertainty was assumed to be time-invariant, the open-loop prediction horizon used to determine optimal control actions can lead to poor closed-loop performance and instability. The worst-case parameter can change from one optimization to another due to the feedback update, which was not accounted for in the open-loop formulation. Lee and Yu

[83], Scokaert and Mayne [137], and Lee and Cooley [81, 82] designed *min-max* predictive controllers that explicitly considered time-varying uncertainties. Lee and Cooley [81, 82] computed a single optimal trajectory, which was more conservative than the one proposed by Scokaert and Mayne [137], which computes time-varying control trajectories.

The main disadvantage of *min-max* control theories is its computational intensity. These optimization problems are expensive to solve on-line. There is a need for computationally inexpensive techniques for robust MPC synthesis that are suitable for on-line implementation and that allow incorporation of a broad class of model uncertainty descriptions. Recent developments in the theory and application of optimization involving linear matrix inequalities (LMIs) [21] to MPC based techniques sparked an entirely new class of methodologies for robust predictive control. There are two main reasons why LMI optimization is relevant to MPC.

1. LMI-based optimization problems can be solved in times comparable to that required for the evaluation of an analytical solution for a similar problem. Thus, LMI optimization can be implemented on-line.

2. It is possible to recast much of existing robust control theory in the framework of LMIs [58, 119].

Kothare et al. [69] proposed using an LMI-based optimizer to find robustly stabilizing state feedback gains that minimized the "worst" performance objective for the given model uncertainty description subject to input and state constraints. Time-varying uncertainty was incorporated by state feedback and the recalculation of the feedback gain at each time instance. Casavola et al. [24] extended the concept above to uncertain systems with input constraint saturation. Lu and Arkun [91] introduced a quasi-scheduling feature by determining the first control action in the optimal control trajectory independent of the state feedback control policy. This formulation was possible only if the correct model at time t, $(A(t), B)$, was known. There was no uncertainty in B. Lu and Arkun [90] increased the model selectivity further by updating the original polytope at t to a smaller polytope around $(A(t), B)$. Slupphaug and Foss [146, 147, 148] separated the uncertain models into non-overlapping clusters. There was a different state feedback gain for each cluster. Robust stability was guaranteed if there was a common Lyapunov function for all models and feedback gains. Chilali et al. [27] proposed robust pole placement in the LMI regions.

4.5 Integral Control

In the 1970's, researchers began studying integral control for non-zero set point tracking in the presence of model uncertainty. Davison et al. [37, 38, 39], Schmitendorf and Barmish [136], Grasselli et al. [56], Freeman and Kokotovic [49], Christofides [31] proposed using a servo-compensator to achieve asymptotic tracking for either linear or nonlinear time-invariant multivariable systems. The control actions were determined by a linear state and error feedback control policy,

$$u = K_0 x + K e \tag{4.10}$$

in which e was the error defined as $e = y - y_t$. The feedback gains K_0 and K minimized a performance objective subject to the nominal process model. Robust stability was achieved by requiring K_0 and K be stabilizing for all bounded parametric uncertainties, $\Delta A \in \Omega_A$, $\Delta B \in \Omega_B$, $\Delta C \in \Omega_C$. However, the size and location of the stabilizable uncertainties could not be quantified analytically. According to Davison [37], robust asymptotic tracking could be accomplished only if the following two conditions were satisfied:

1. the system to be regulated has at least the same number of independent inputs as there are independent outputs to be regulated.

2. the independent outputs to be regulated must be physically measurable.

Even though Equation 4.10 can successfully perform robust asymptotic tracking of non-zero set points, it can neither handle constraints nor time-varying uncertainties. Bemporad and Mosca [11, 12], Blanchini [16], and Khammash and Zou [66] discussed constraint handling by taking control actions on the difference between the reference input trajectory and the past input value but did not discuss how the reference input trajectory was determined.

The second integral control design found in the robust control literature uses input output process models with state measurement. In this case, the state is constructed from past inputs and outputs [43, 141]. The process models are ARMAX (Auto Regressive Moving Average Exogenous Input) polytopic models. The state is defined as

$$x(t) = \begin{bmatrix} y(t)^T & \dots & y^T(t-n+1) & u^T(t-n+1) & \dots & u^T(t-1) & \xi^T(t) \end{bmatrix}^T, \tag{4.11}$$

and $\xi(t)$ is an integration of the tracking error [75].

$$\xi(t) = \sum_{j=0}^{t} (y_t - y(j)) \tag{4.12}$$

Control actions are taken whenever $\xi(t) \neq 0$ is not equal to zero, which occurs if $y(t)$ is not equal to y_t [123]. This integral control formulation works well for unconstrained systems. In the presence of constraints, wind-up can occur when set point is outside the feasible region and the system is operating on one or more constraints. Figure 4.3 shows closed-loop performance for the following system.

$$y(s) = G_i u(s) \tag{4.13}$$

$$-\infty \leq u(t) \leq 8$$

$$-\infty \leq y(t) \leq \infty$$

in which

$$G_1 = \frac{0.1}{s+10} \qquad G_2 = \frac{1}{s+10} \tag{4.14}$$

After realization and discretization, the ARMAX models are

$$\text{Model 1}: \quad y(t+1) = 0.3679y(t) + 0.0063u(t)$$

$$\text{Model 2}: \quad y(t+1) = 0.3679y(t) + 0.0632u(t)$$

Model 2 is the plant. Figure 4.3 shows the closed-loop performance for a set point change from 0 to 1. The new set point is initially infeasible because of the input constraint $u_{\max}$. At time $t = 50$, a step disturbance of 0.3 enters the system and causes $y_t = 1$ to be reachable. Between $t = 0$ to $t = 50$, constraint saturation causes the error factor $\xi(t)$ to build up. Even though a control action less than the input constraint $u_{\max}$ is sufficient to control $y(t)$ to y_t after the disturbance entrance, the controller uses the maximum control action allowable trying to decrease $\xi(t)$. Once $\xi(t)$ is small enough so that maximum control action is no longer needed, manipulated input decreases from the $u_{\max}$ value and controls $y(t)$ to y_t. The time between disturbance entrance ($t = 50$) and input decrease from $u_{\max}$ (approximately $t = 85$) is known as wind-up. The delay is the main disadvantage for the ARMAX model integrator formulation.

Another integral control design proposes taking control actions on the difference between the output measurement and set point. This integral control method can be applied to input output process models [78, 118, 133, 134], and orthonormal series functions [113]. All of which are in discrete time. The main disadvantage for this integral control formulation is the complexity of the relationship between models for Δu and Δy, which are in the velocity form, and

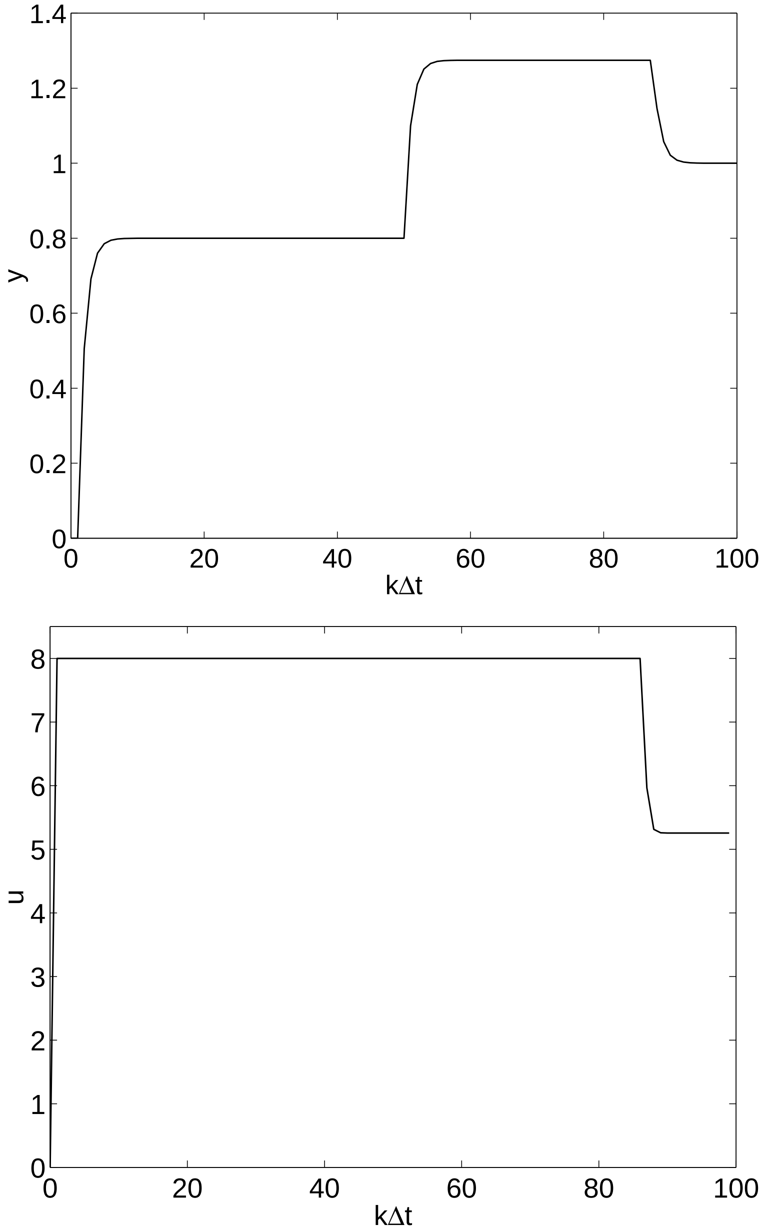

Figure 4.3: Constraint saturation example that exhibits wind-up with plant=model 2.

models for u and y. If the model is time invariant, the model for $(\Delta u, \Delta y)$ is identical to the model for (u, y). If the model is time varying, there is no straightforward transformation between models for (u, y) and $(\Delta u, \Delta y)$. For time-invariant uncertainties, Pannocchia and Rawlings [114] summarized the advantages and disadvantages of the velocity form. The state-space model in the velocity form is not stabilizable, which causes the failure of the infinite horizon regulator with the terminal region W and the terminal state feedback control policy $u(t) = Kx(t)$.

Integral control can also be achieved by performing steady-state target calculations to determine the correct steady-state input values that reject all disturbances in the uncertainty description. Marconi and Isidori [96] stated that the correct steady-state x_s and u_s values are needed to achieve “perfect tracking” of the set point. Kassmann et al. [64] presented a steady-state target calculation that explicitly accounts for model uncertainty. The steady-state input values were determined by minimizing the steady-state deviations from target for the nominal model, which was in the velocity form, subject to stability constraints for all models in the uncertainty description.

Other non-zero set point tracking controller designs include the one proposed by Mayne and Schroeder [97] and Lee and Cooley [81, 82]. Mayne and Schroeder [97] used the Francis-Wonham condition [48] to achieve convergence of the output to the set point with state measurement for systems with no parametric uncertainty and exogenous disturbances. In the presence of disturbances, convergence to the set point could not be guaranteed. Lee and Cooley [81, 82] proposed adding integrators into the process model to successfully reject disturbances. Stability was guaranteed by the addition of a terminal stability constraint that required all integrators to converge to zero at the end of the prediction horizon. However, disturbance rejection was accomplished with state measurement, and non-zero set point tracking was not addressed. Megias et al [101] proposed using the terminal constraint $y(t+N) - y_t = 0$ to achieve offset-free set point tracking. The robust set point tracking methods proposed by Lee and Cooley [81, 82] and Megias et al. [101] involve intensive on-line computations and cannot determine if offset-free set point tracking is possible off-line.

4.6 Summary

A lot of research has been done in robust predictive control, but few addressed the issue of integral control for non-zero set point tracking in a state-space setting with output feedback. Most of the stability criteria, which guarantee robust convergence of the state to the origin, require state measurement. Few plants, if any, have the origin as their desired operating condition. As discussed in Section 4.5, the published work in the robust integral control field are limited to the use of input output process models, which lack the more detailed process descriptions provided by the state-space models, or models in the velocity form, which cannot be transformed to the regular state-space model in a straightforward manner. In the state-space setting, Davison and Goldenberg [39] introduced integral control for unconstrained time-invariant systems. But no one, to our knowledge, has addressed non-zero set point tracking in the absence of state measurement for time-varying systems with model uncertainty.

In light of this literature analysis, we state the main motivation for our research project. Our goal is to design a robust MPC controller that guarantees convergence of the output to non-zero set points in the presence of time-varying model uncertainty. Our theory utilizes the structure of the polytopic uncertainty description to design robustly stabilizing feedback control policies. An off-line closed-loop stability condition is developed that determines if offset-free set point tracking can be accomplished for all models in the uncertainty description. State measurement is replaced with output measurement and state estimation, which is more applicable for the industrial practitioners.

Chapter 5

Robust Model Predictive Control

In this chapter, we propose a new robust MPC method that guarantees stability and offset-free set point tracking in the presence of model uncertainty. A *min-max* optimization problem that explicitly accounts for model uncertainty is used to determine the optimal control actions subject to the input and the output constraints. The robust regulator uses a tree trajectory to forecast time-varying model uncertainty. The controller design procedure uses integrators to reject disturbances and maintain the process at the optimal operating conditions or set points. Constraints may cause offset, which occurs when the set points are unreachable. In the feasible region where constraints are not active, the robust MPC theory we propose achieves offset-free non-zero set point tracking if there exists a control policy that robustly stabilizes all models in the uncertainty set.

The new RMPC method achieves non-zero set point tracking for a process with time-varying model uncertainty and input and output constraints. Scokaert and Mayne [137] proposed using time-varying sequences of bounded disturbances to model the uncertainty. We propose using time-varying sequences of models in a polytope to forecast the model uncertainties. Integral control is achieved by adding state disturbances to all models on the vertices of the polytope. Offset-free non-zero set point tracking is achieved by computing the steady-state x_s and u_s values that minimize the sum of the deviations of the predicted steady-state output from the set point for all models in the uncertainty description. The values of the steady-state biases are chosen so that all models in the uncertainty description have the same x_s and u_s values. Because every model in the polytope has the same x_s and u_s values, offset-free control can be accomplished by designing a robust control policy that controls the state to the same x_s for all models in the uncertainty description.

5.1 Problem Statement

The uncertain dynamic system is described by the following discrete state-space model:

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t) \\ y(t+1) &= Cx(t+1) \end{aligned} \tag{5.1}$$

in which $(A(t), B(t))$ are the time-varying state-space model matrices; $C \in \mathbb{R}^{p \times n}$ describes the relationship between the output and the state without any uncertainty; $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, the manipulated input, also known as the control decision variable, and the controlled output vectors, respectively. The controlled output and the manipulated input are subject to the following constraints:

$$u(t) \in \mathbb{U}, \quad x(t) \in \mathbb{X}, \quad y(t) \in \mathbb{Y} \tag{5.2}$$

in which $\mathbb{U}$, $\mathbb{X}$, and $\mathbb{Y}$ are convex and closed subsets of $\mathbb{R}^m$, $\mathbb{R}^n$, and $\mathbb{R}^p$, respectively.

When model uncertainty is present, the exact plant model $(A(t), B(t))$ is not known. The model uncertainty region is described by Ω, the convex hull of $\Pi = \{(A_1, B_1), (A_2, B_2), \ldots, (A_{\mathcal{I}}, B_{\mathcal{I}})\}$. The convex hull is defined as the linear convex combination of all models in Π. Let $i \in \mathbb{I}$ be the model index. $(A(t), B(t))$ is in Ω if and only if there exist $\mu_1(t), \mu_2(t), \ldots, \mu_{\mathcal{I}}(t) \in \mathbb{R}$ such that

$$A(t) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) A_i \qquad \text{and} \qquad B(t) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) B_i$$

for any

$$0 \leq \mu_i(t) \leq 1 \qquad \text{and} \qquad \sum_{i=1}^{\mathcal{I}} \mu_i(t) = 1$$

The system is said to be at steady state at time T if

$$u_s = u(s+1) = u(s) \quad \text{for all } s \geq T \tag{5.3}$$

$$x_s = x(s+1) = x(s) \quad \text{for all } s \geq T$$

$$y_s = y(s+1) = y(s) \quad \text{for all } s \geq T$$

in which $u_s \in \mathbb{R}^m$, $x_s \in \mathbb{R}^n$, and $y_s \in \mathbb{R}^p$ are the steady-state control, state, and controlled output vectors that satisfy the hard constraints

$$u_s \in \mathbb{U}, \quad x_s \in \mathbb{X}, \quad y_s \in \mathbb{Y}$$

Because there is no uncertainty in the C matrix,

$$y_s = Cx_s$$

at steady state.

5.2 Definitions and Notations

5.2.1 System Theory

D1: A function $\alpha : \mathbb{X} \to \mathbb{R}$ in which $\mathbb{X} \subset \mathbb{R}^n$ is positive definite if $\alpha(x) > 0$ for all $x \in \mathbb{X}$, $x \neq 0$, and $\alpha(0) = 0$.

D2: A function $\alpha : \mathbb{X} \to \mathbb{R}_+$ in which $\mathbb{X} \subset \mathbb{R}^n$ is said to be proper if it is continuous, positive definite, strictly increasing, and $\alpha(s) \to \infty$ as $s \to \infty$.

D3: The symbol $x^{(r)}$ is denotes the r^{th} element of the vector $x \in \mathbb{R}^n$.

D4: A control law is a function $h(\cdot) : \mathbb{X} \to \mathbb{U}$. A control policy is a sequence of control laws.

D5: The system $x(s+1) = A_i x(s) + B_i u(s)$, $y(s) = Cx(s)$ is said to be admissible if it satisfies the state and control constraints $x(s) \in \mathbb{X}$ and $u(s) \in \mathbb{U}$ for all $s \geq 0$ and for all models $i \in \{1, \ldots, \mathcal{I}\}$. A control sequence $\{u(s), u(s+1), \ldots\}$ is said to be admissible (for a given initial state) if the associated system is admissible and is said to be feasible if it satisfies all the constraints in an associated optimal control problem.

D6: The system $x(s+1) = Ax(s) + Bu(s)$ in which (A, B) is time invariant is said to be controllable if there exists a positive integer n_c and a proper function W_c such that for every system satisfying $x(0) = x$ and $(x(s), u(s)) = (0, 0)$ for all $s \geq n_c$.

$$\sum_{s=0}^{n_c-1} \| (x(s), u(s)) \| \leq W_c(\| x \|)$$

5.2.2 The New Robust Regulator

We require some terminology to describe the level, node, and branch of the tree trajectory used by the robust regulator to forecast the system behavior with time-varying model uncertainty.

D7: The integer $N \in \mathbb{I}$ denotes the finite prediction horizon length.

D8: The integer $j \in \mathbb{I}$ denotes the tree trajectory level which is equal to the number of steps into the forecast horizon, $j = 0, 1, \ldots, N$.

D9: The integer string $\tau \in \mathbb{I}^j$ is the node index at level j that keeps track of the sequence of models needed to arrive at the node τ.

D10: The branch index $l \in \mathbb{N}$ denotes the collection of nodes for one possible time-varying combination of (A_i, B_i) in Π.

D11: Let $a \in \mathbb{N}$ and $i \in \mathbb{N}$ be integers and $\tau \in \mathbb{N}^a$ be a node. The operator $\mathbf{P}$ is defined as

$$\mathbf{P}(\tau, i) = \tau \tag{5.4}$$

in which $(\tau, i) \in \mathbb{N}^{a+1}$ is a node index of length $a+1$. The operator $\mathbf{P}$ removes the last integer in the node index (τ, i).

D12: The unit integer string $\mathbf{1}_N \in \mathbb{N}^N$ is defined as

$$\mathbf{1}_N = \underbrace{1, 1, 1, \ldots, 1}_{N \text{ times}} \tag{5.5}$$

D13: If $a \in \mathbb{N}$ and $b \in \mathbb{N}$, then $(a)_b$ is defined as the number a in base b.

D14: Let $a \in \mathbb{N}^n$ and $b \in \mathbb{N}^m$ be integer strings of length n and m, respectively, with $n \geq m$. If $a = a_n, a_{n-1}, \ldots, a_2, a_1$ and $b = b_m, b_{m-1}, \ldots, b_2, b_1$, then the operator $\oplus$ is defined as

$$a \oplus b = a \oplus \overline{b} = a_n + \overline{b}_n, a_{n-1} + \overline{b}_{n-1}, \ldots, a_2 + \overline{b}_2, a_1 + \overline{b}_1 \tag{5.6}$$

in which $\overline{b} = 0, \ldots, 0, b_m, \ldots, b_1$ such that $\overline{b} \in \mathbb{N}^n$.

D15: If $a \in \mathbb{N}^n$ and $b \in \mathbb{N}^m$ are the node indices of length n and m, respectively, then the operator ; separates node a from node b in the set $\{a\ ;\ b\}$.

D16: The node index $\tau_l \in \mathbb{I}^N$ denotes the terminal node of branch l and is defined as

$$\tau_l = (l-1)_{\mathcal{I}} \oplus \mathbf{1}_N \tag{5.7}$$

D17: The set T_l contains all the nodes on branch l and is defined as

$$T_l = \{\tau_l \, ; \, \mathbf{P}(\tau_l) \, ; \, \mathbf{P}^2(\tau_l) \, ; \, \ldots \, ; \, \mathbf{P}^{N-1}(\tau_l)\} \tag{5.8}$$

D18: The set $\overline{T}_l$ is a subset of T_l and is defined as

$$\overline{T}_l = T_l \setminus \tau_l \tag{5.9}$$

which implies $\tau_l \notin T_l \cap \overline{T}_l$. The set $\overline{T}_l$ contains all the nodes in T_l except for the terminal node τ_l.

5.3 The New Robust Regulator Design

The regulator determines the optimal manipulated input trajectory based on the system behavior predictions of a model. To correctly model the time-varying uncertainty, we propose a method similar to the one proposed by Scokaert and Mayne [137] that produces a tree trajectory for all possible time-varying combinations of the models in Π. The components of the tree trajectory are: level, node, and branch. The tree trajectory is rooted at $x(t)$ and $u(t)$ in the state and input space, respectively. If τ is the node index at level j, then (τ, i) is the node

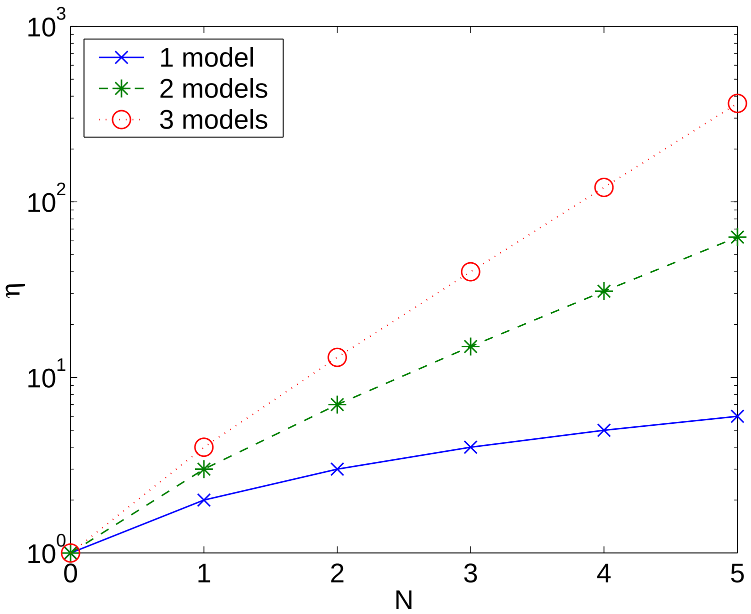

Figure 5.1: η as a function of N and $\mathcal{I}$.

index at level $j+1$ under the i^{th} model forecast.

$$x_{\tau,i}(t) = A_i x_\tau(t) + B_i u_\tau(t) \quad \text{for} \quad i = 1, \ldots, \mathcal{I} \tag{5.10}$$

$$y_{\tau,i}(t) = C x_{\tau,i}(t)$$

in which $x_\tau(t) \in \mathbb{R}^n$, $u_\tau(t) \in \mathbb{R}^m$, and $y_\tau(t) \in \mathbb{R}^p$ are the state, the control decision variable, and the controlled output vectors at node τ, respectively. The total number of nodes η depends on $\mathcal{I}$ and the prediction horizon length N, with N having the larger effect (see Figure 5.1).

$$\eta = \begin{cases} N+1 & \text{if} \quad \mathcal{I} = 1 \\ \frac{\mathcal{I}^{N+1}-1}{\mathcal{I}-1} & \text{if} \quad \mathcal{I} > 1 \end{cases}$$

Each branch of the tree trajectory is formed by one time-varying sequence of (A_i, B_i) in Π. Figure 5.2 illustrates all the branches in the $\mathcal{I} = 3$ and $N = 2$

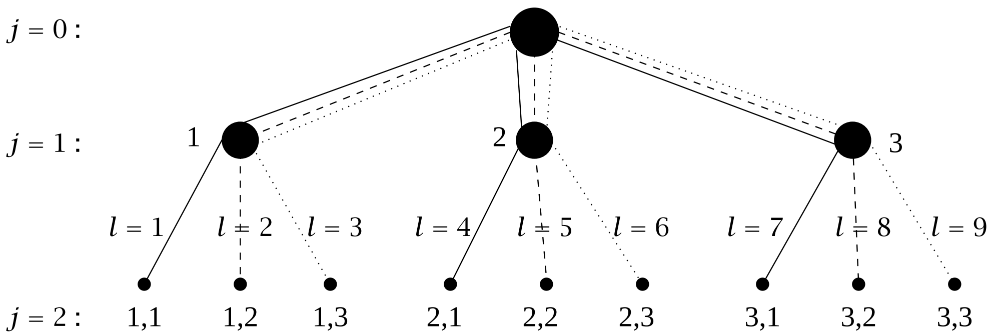

Figure 5.2: Graphical illustration of the branches for $\mathcal{I} = 3$ and $N = 2$.

Branch l	Terminal node τ_l	Set T_l	Set $\overline{T}_l$
1	$1, 1$	$\{1, 1 \,;\, 1\}$	$\{1\}$
2	$1, 2$	$\{1, 2 \,;\, 1\}$	$\{1\}$
3	$1, 3$	$\{1, 3 \,;\, 1\}$	$\{1\}$
4	$2, 1$	$\{2, 1 \,;\, 2\}$	$\{2\}$
5	$2, 2$	$\{2, 2 \,;\, 2\}$	$\{2\}$
6	$2, 3$	$\{2, 3 \,;\, 2\}$	$\{2\}$
7	$3, 1$	$\{3, 1 \,;\, 3\}$	$\{3\}$
8	$3, 2$	$\{3, 2 \,;\, 3\}$	$\{3\}$
9	$3, 3$	$\{3, 3 \,;\, 3\}$	$\{3\}$

Table 5.1: Set definitions for $\mathcal{I} = 3$ and $N = 2$.

tree trajectory. The total number of branches $\mathcal{L}$ is dependent on the number of models $\mathcal{I}$ and the prediction horizon length N.

$$\mathcal{L} = \mathcal{I}^N$$

Table 5.1 lists the branches, node sets T_l and $\overline{T}_l$, and the terminal nodes τ_l for the $\mathcal{I} = 3$ and $N = 2$ tree trajectory shown in Figure 5.2.

Figure 5.3 is a graphical illustration of the tree trajectory for $\mathcal{I} = 2$ and $N = 4$ in the $x \in \mathbb{R}^2$ state space. Each node has a control decision variable. The total number of decision variables is equal to the total number of nodes η. For

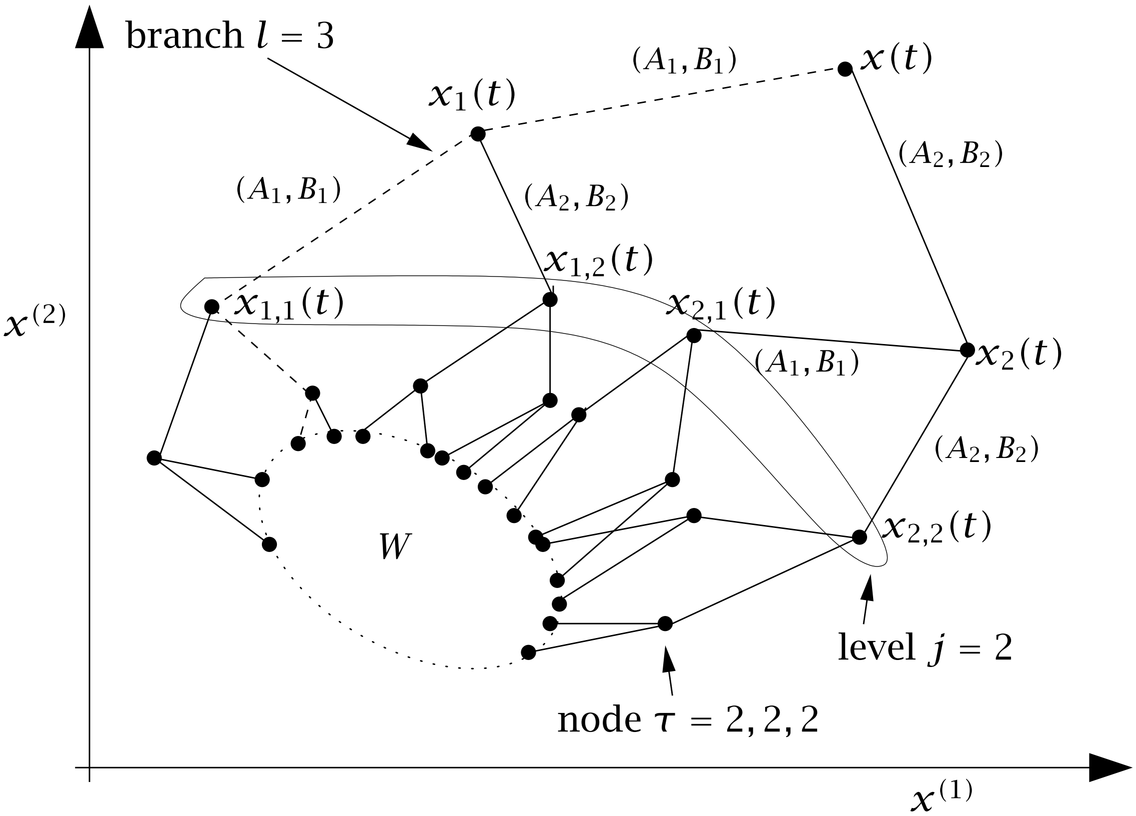

Figure 5.3: Tree trajectory with $\mathcal{I} = 2$ and $N = 4$ in the $x \in \mathbb{R}^2$ state space.

the tree trajectory in Figure 5.3, there are

$$\mathcal{L} = 2^4 = 16$$

branches and

$$\eta = \frac{2^5 - 1}{2 - 1} = 31$$

decision variables. The forecasted state and output at node τ on level $j = 2$ and branch $l = 3$ as shown in Figure 5.3 are:

$$x_{1,1}(t) = A_1 x_1(t) + B_1 u_1(t)$$

$$y_{1,1}(t) = C x_{1,1}(t)$$

5.4 The Open-Loop Control Problem

The objective of the control problem is to find the control actions that, once implemented, cause all branches in the tree trajectory to converge to x_s and u_s. In nominal MPC, the regulator computes the optimal input trajectory by minimizing a performance objective that is the sum of the deviations of the state and input from the steady-state x_s and u_s values. The same concept is extended to the robust MPC regulator. We define the following deviation variables used to compute the performance objective.

$$\overline{u}_\tau(t) = u_\tau(t) - u_s$$

$$\overline{x}_\tau(t) = x_\tau(t) - x_s$$

$$\overline{y}_\tau(t) = y_\tau(t) - Cx_s.$$

The state and output deviation variables are defined as

$$\begin{aligned}
&\overline{x}_i(t) = A_i\overline{x}(t) + B_i\overline{u}(t) \quad \text{for } \ i = 1, \ldots, \mathcal{I} \qquad (5.11)\\
&\overline{x}_{\tau,i}(t) = A_i\overline{x}_\tau(t) + B_i\overline{u}_\tau(t) \quad \text{for } \ \tau \in \overline{T}_l \ \text{ and } \ i = 1, \ldots, \mathcal{I}\\
&\overline{y}_\tau(t) = C\overline{x}_\tau(t)
\end{aligned}$$

subject to

$$\begin{aligned}
&\overline{u}_\tau(t) \in \mathbb{U}_d \quad \text{for } \tau \in T_l \text{ and } l = 1, \ldots, \mathcal{L} \qquad (5.12)\\
&\overline{x}_\tau(t) \in \mathbb{X}_d \quad \text{for } \tau \in T_l \text{ and } l = 1, \ldots, \mathcal{L}\\
&\overline{y}_\tau(t) \in \mathbb{Y}_d \quad \text{for } \tau \in T_l \text{ and } l = 1, \ldots, \mathcal{L}\\
&\overline{x}_{T_l}(t) \in W_d \quad \text{for } l = 1, \ldots, \mathcal{L}
\end{aligned}$$

in which

$$\mathbb{U}_d = \mathbb{U} - u_s, \ \ \mathbb{X}_d = \mathbb{X} - x_s, \ \ \mathbb{Y}_d = \mathbb{Y} - y_s, \ \ W_d = W - x_s \qquad (5.13)$$

The subsets $\mathbb{U}_d$, $\mathbb{X}_d$, and $\mathbb{Y}_d$ are convex, closed subsets of $\mathbb{R}^m$, $\mathbb{R}^n$, $\mathbb{R}^p$, respectively, and contain the origin in their interiors. The output admissible set W_d is compact, contains the origin in its interior, and is positively invariant for

$$\overline{x}_{\tau,i}(t) = (A_i + B_iK)\overline{x}_\tau(t)$$

for any $i = 1, \ldots, \mathcal{I}$, $\overline{x}_\tau(t) \in W_d$, and $W_d \subset \mathbb{X}_d$. The control law in W_d is

$$h_W(\overline{x}_\tau(t)) = \overline{u}_\tau(t) = K\overline{x}_\tau(t) \quad \text{for any } \overline{x}_\tau(t) \in W_d \tag{5.14}$$

which is admissible because it satisfies the control constraint $h_W(\cdot) \subset \mathbb{U}_d$.

Let

$$U_l(t) = \left\{\overline{u}(t), \{\overline{u}_\tau(t) | \tau \in \overline{T}_l\}\right\} \quad \text{for } l = 1, \ldots, \mathcal{L}$$

denote the sequence of control actions for the l^{th} branch of the tree trajectory. The open-loop control problem in the robust regulator is

$$\min_{U_l(t)} \max_{l=1,\ldots,\mathcal{L}} \Phi_l(t; U_l(t)) \tag{5.15}$$

subject to Equations 5.11 and 5.12 in which $\Phi_l(t; U_l(t))$ is the performance objective for the l^{th} branch using the control sequence $U_l(t)$ and is defined as

$$\Phi_l(t; U_l(t)) = \phi(\overline{x}(t), \overline{u}(t)) + \sum_{\tau \in \overline{T}_l} \phi(\overline{x}_\tau(t), \overline{u}_\tau(t)) + \phi_N(\overline{x}_{\tau_l}(t)). \tag{5.16}$$

To reduce the notation complexity, let

$$\Phi_l(t) = \Phi_l(t; U_l(t)) \tag{5.17}$$

unless we need to stress the functional dependence of $\Phi_l(t)$ on $U_l(t)$. The solution to Equation 5.15 is denoted as $U^*(t)$, which is the sequence of control actions that minimizes the deviations of the state and the control from x_s and u_s for the branch with the "worst" or the "largest" performance objective. The control action implemented at time t is $u(t) = \overline{u}^* + u_s$, the first control action in $U^*(t)$ which is in the deviation variable form plus the steady-state control u_s.

Remarks

R1: The function

$$\phi(x, u) = x^T Q x + u^T R u$$

in which $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite is continuous, positive definite, and proper.

R2: The function

$$\phi_N(x) = x^T F x$$

in which $F \in \mathbb{R}^{n \times n}$ is symmetric and positive definite is continuous, positive definite, and proper.

R3: Because both $\phi(x, u)$ and $\phi_N(x)$ are continuous, positive definite and proper, $\Phi_l(t)$ is continuous, positive definite, and proper.

Assumption

A1: The control constraint sets $\mathbb{U}_d$, $\mathbb{X}_d$, and $\mathbb{Y}_d$ are compact and convex, and contain the origin in their interiors.

Theorem 5.4.1 *Suppose A1 holds. If there exist K and $F = F^T > 0$ such that*

$$F - Q - K^T R K - (A_i + B_i K)^T F (A_i + B_i K) \geq 0 \quad \text{for } i = 1, \ldots, \mathcal{I} \tag{5.18}$$

then Equation 5.15 generates a control policy that guarantees convergence of $\overline{x}(t)$ and $\overline{u}(t)$ to the origin for all time-varying sequences of (A_i, B_i) in Π.

Proof. See Appendix 5.12.2 □

Remarks

R4: Theorem 5.4.1 guarantees convergence of $\overline{x}(t)$ and $\overline{u}(t)$ to the origin for time-varying sequences of (A_i, B_i) in Π and *not* time-varying sequences of $(A(t), B(t))$ in Ω.

R5: The terminal region W_d is time-invariant and contains the origin in its interior.

Theorem 5.4.2 *If there exist K and $F = F^T > 0$ such that Theorem 5.4.1 holds, then Equation 5.15 generates a control policy that guarantees convergence of $x(t)$ and $u(t)$ to x_s and u_s, respectively, for any time-varying sequence of $(A(t), B(t))$ in Ω.*

Proof. See Appendix 5.12.3 □

Both Kothare et al. [69] and Lu and Arkun [91] proposed control theories with a robust stability constraint similar to Equation 5.18. Kothare et al. [69] proposed a finite prediction horizon length of $N = 0$ in which K and F are recalculated at each time step. Lu and Arkun [91] used $N = 1$ with knowledge of the correct model at time t and the linear state feedback policy, which is recalculated after every state measurement. Our regulator formulation has the following features:

- K and F are computed off-line and are not recalculated during the controller operation.

- The control actions implemented are determined by the tree trajectory forecasts and not the control policy $\overline{u}(t) = K\overline{x}(t)$. This control policy is used only to approximate the infinite-horizon cost, and it is not used as a control policy.

5.5 Existence of K and F

The state feedback gain K in the control law $\overline{u}_\tau(t) = K\overline{x}_\tau(t)$ for all $\overline{x}_\tau(t) \in W_d$ that guarantees decreasing robust performance objective can be transformed to

$$K = YP^{-1} \quad \text{and} \quad F = P^{-1} \tag{5.19}$$

in which $P = P^T > 0$ and Y are obtained from the solution (if it exists) of the following LMI-based optimization problem.

$$\min_{\gamma, Y, P} \ \gamma \tag{5.20}$$

subject to

$$\gamma \geq 0 \tag{5.21}$$

$$P > 0 \tag{5.22}$$

$$\begin{bmatrix} P & (A_iP+B_iY)^T & (Q^{1/2}P)^T & (R^{1/2}Y)^T \\ (A_iP+B_iY) & P & 0 & 0 \\ Q^{1/2}P & 0 & I & 0 \\ R^{1/2}Y & 0 & 0 & I \end{bmatrix} \geq \gamma I \ \forall i = 1,\ldots,\mathcal{I} \quad (5.23)$$

Equation 5.23 is an LMI and is equal to the Schur complement of Equation 5.18 after the variable transformation (Equation 5.19).

Effective algorithms are available for finding the solutions of the LMI-based optimization problems. These algorithms converge to the global optimum with non-heuristic stopping criteria. Boyd and El Ghaoui [20], Alizadeh et al. [2], Nesterov and Nemirovsky [111] and Vandenberghe and Boyd [152] showed that upon termination, the algorithms arrive at a solution that is within some pre-specified numerical tolerance of the global optimum. Numerical experience shows that these algorithms solve LMI problems efficiently.

Lemma 5.5.1 *If there exist Y and $P = P^T > 0$ such that $K = YP^{-1}$, $F = P^{-1}$, and*

$$F - Q - K^TRK - (A_i + B_iK)^TF(A_i + B_iK) \geq 0 \quad \forall i = 1,\ldots,\mathcal{I}$$

then

$$F - Q - K^TRK - (A + BK)^TF(A + BK) \geq 0$$

for all (A, B) in Ω.

Proof. The Schur complement of $F - Q - K^TRK - (A + BK)^TF(A + BK)$ is a linear convex combination of the Schur complements of $F - Q - K^TRK - (A_i +$

$B_iK)^T F(A_i + B_iK)$. If there exist Y and P such that Equation 5.23 holds, then the linear convex combination of the Schur complement in Equation 5.23 is positive semidefinite. □

5.6 Output Feedback

Feedback is an essential part of any control theory and is used in predictive control to determine the difference, if any, between the predicted and the measured state. In the absence of plant-model mismatch, there is no discrepancy between the nominal MPC's predicted state trajectory and the measured state trajectory. Feedback is not necessary when there is no modeling uncertainty. However, the RMPC method we are proposing is designed specifically to handle model uncertainties. Feedback is therefore an essential part of our theory.

We propose using the input output ARMAX (Auto Regressive Moving Average Exogenous Input) model to describe the system behavior. The state is formed by storing all past inputs and outputs that affect the output at the next time step. If the ARMAX model is

$$\begin{aligned} y(t+1) \quad &= a_1 y(t) + a_2 y(t-1) + \cdots + a_{n_y} y(t - n_y + 1) \\ &+ b_1 u(t) + b_2 u(t-1) + \cdots + b_{n_u} u(t - n_u + 1), \end{aligned}$$

the state is defined as

$$x(t) = \begin{bmatrix} y(t) & \ldots & y(t-n_y+1) & u(t-1) & \ldots & u(t-n_u+1) \end{bmatrix}^T, \quad (5.24)$$

and the state-space model is

$$x(t+1) = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n_y} & b_2 & \cdots & b_{n_u} \\ I & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ 0 \\ \vdots \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} I & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} x(t).$$

In principle, because all $y(t)$ and $u(t)$ are measurable, the state $x(t)$ is measurable. Measurement noise can be removed by filtering the signals $y(t)$ and $u(t)$ before constructing $x(t)$.

When model uncertainty is present, we propose using the integrating disturbance models to reconcile the difference between the model predictions and the measured states. Because the correct model is not known, a different integrating state is added to each (A_i, B_i).

$$x_{\tau,i}(t) = A_i x_\tau(t) + B_i u_\tau(t) + p_{i,s} \tag{5.25}$$
$$y_{\tau,i}(t) = C x_{\tau,i}(t)$$

in which $p_{i,s} \in \mathbb{R}^n$ is the steady-state bias for model i, and $p_{i,s}$ remains constant until there is a new measurement.

The model disturbances are updated whenever new measurements are available via

$$p_i(t+1) = p_{i,s} + L_i(x(t+1) - x_i(t)) \tag{5.26}$$

in which $p_i(t+1) \in \mathbb{R}^n$ is the disturbance that integrates the difference between the estimated state and the predicted state from the previous time step. There is a different disturbance filter gain $L_i \in \mathbb{R}^{n \times n}$ for each model i.

5.7 Target Calculation

The set point is the desired value of the controlled output $y(t)$, but the regulator performance objective penalizes deviations of the state $x(t)$ and input $u(t)$ from the steady-state x_s and u_s values, respectively. The target calculation therefore has two main objectives. First, it is necessary to determine if the set point is reachable for the given process constraints. Second, the target calculation determines the best steady-state x_s and u_s that minimize the sum of the difference between the steady-state controlled output y_s and the set point y_t for models $i = 1, \ldots, \mathcal{I}$.

The steady-state u_s, x_s, and y_s values are defined by Equation 5.3. In terms of the process models, there is an $x_{i,s} \in \mathbb{R}^n$ such that

$$x_{i,s} = A_i x_{i,s} + B_i u_s$$

for every $u_s \in \mathbb{R}^m$. The value of the $x_{i,s}$ depends on the model i (see Figure 5.4). The steady-state biases $p_{i,s} \in \mathbb{R}^n$ are defined so that

$$\begin{aligned} x_{i,s} &= A_i x_{i,s} + B_i u_s + p_{i,s} \\ x_s &= x_{i,s} \end{aligned} \tag{5.27}$$

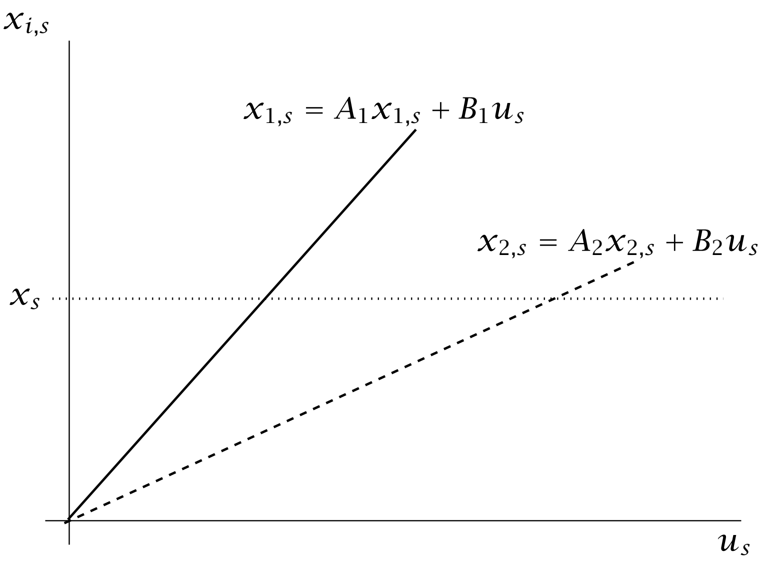

Figure 5.4: Graphical illustration of the steady-state equality constraints without the steady-state bias $p_{i,s}$.

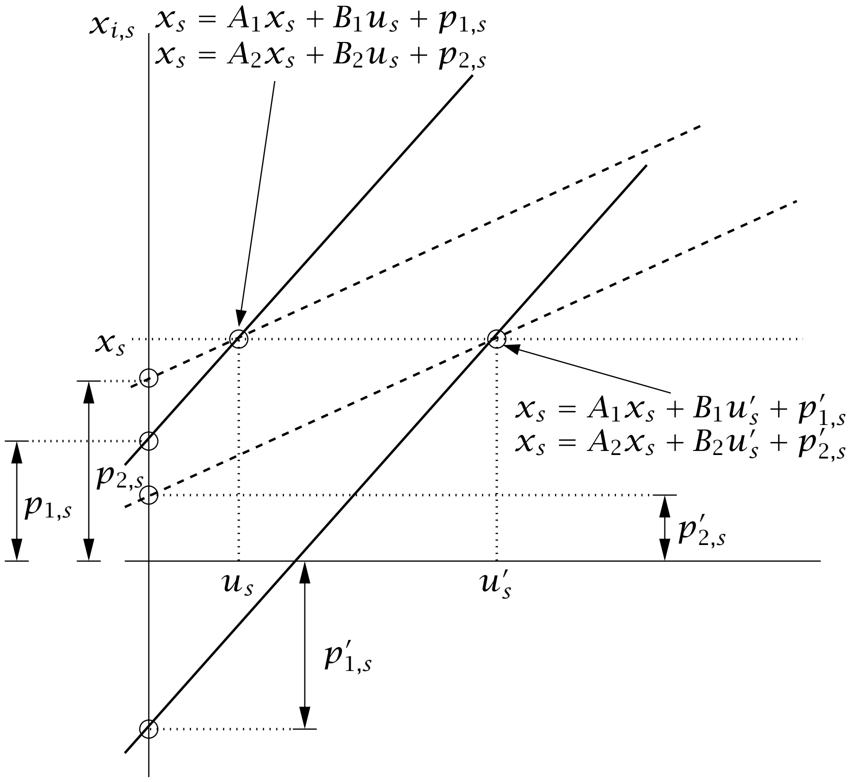

Figure 5.5: Graphical illustration of the steady-state bias $p_{i,s}$.

Figure 5.5 shows that there are multiple steady-state input values that satisfy the steady-state equality constraint in Equation 5.27 because of the extra degree of freedom provided by the steady-state bias $p_{i,s}$. Even though

$$u_s \neq u_s', \qquad p_{1,s} \neq p_{1,s}', \qquad p_{2,s} \neq p_{2,s}'$$

they satisfy the steady-state equality constraint with the same x_s value.

In single-model MPC (SMPC), the target calculation is used to determine the steady-state x_s and u_s values. We modify the RMPC target calculation so that it determines not only the x_s and u_s values but the unique $p_{i,s}$ values that satisfy Equation 5.27.

Together, the steady-state biases $p_{i,s}$ and the model disturbances $p_i(t)$ act as the integrator needed to achieve offset-free set point tracking. The model disturbance $p_i(t)$ integrates the difference between the plant and the forecasted state using model i. The steady-state bias $p_{i,s}$ are chosen so that the difference between $p_{i,s}$ and $p_i(t)$ is as small as possible while satisfying the steady-state equality constraints in Equation 5.27. With integral control, the model forecast is Equation 5.25. Once the new state measurement is available, the model disturbances are updated by Equation 5.26.

Assumption

A2: The A_i matrices are non-integrating for all $i = 1, \ldots, \mathcal{I}$.

With the new updated model disturbances, the optimal steady-state u_s, x_s, and y_s values are determined by the following optimization problem if assumption A2 holds.

$$\min_{u_s, p_{1,s}, \ldots, p_{\mathcal{I},s}} \quad \Phi_s \tag{5.28}$$

subject to

$$\begin{aligned} x_s &= (I - A_i)^{-1}(B_i u_s + p_{i,s}) \quad \text{for } i = 1, \ldots, \mathcal{I} \\ u_s &\in \mathbb{U} \\ (I - A_i)^{-1}(B_i u_s + p_{i,s}) &\in \mathbb{X} \quad \text{for } i = 1, \ldots, \mathcal{I} \\ G_i u_s + C(I - A_i)^{-1} p_{i,s} &\in \mathbb{Y} \quad \text{for } i = 1, \ldots, \mathcal{I} \end{aligned} \tag{5.29}$$

in which the steady-state gain matrix G_i is defined as

$$G_i = C(I - A_i)^{-1} B_i$$

and the steady-state performance objective Φ_s is defined as

$$\begin{aligned} \Phi_s = \quad & \sum_{i=1}^{\mathcal{I}} (G_i u_s + C(I - A_i)^{-1} p_{i,s} - y_t)^T Q_s (G_i u_s + C(I - A_i)^{-1} p_{i,s} - y_t) \\ & + \sum_{i=1}^{\mathcal{I}} (p_{i,s} - p_i(t))^T P_i (p_{i,s} - p_i(t)) \end{aligned}$$

The positive definite matrix $Q_s \in \mathbb{R}^{p \times p}$ is the penalty matrix for the difference between the predicted steady-state output $y_s = G_i u_s + C(I - A_i)^{-1} p_{i,s}$ and the set point $y_t \in \mathbb{R}^p$. The symmetric positive definite matrices $P_i \in \mathbb{R}^{n \times n}$ penalize the difference between the steady-state biases $p_{i,s}$ and the model disturbance estimates $p_i(t)$. The minimization of the difference between $p_{i,s}$ and $p_i(t)$ allows Equation 5.28 to determine the unique steady-state biases so that Equation 5.27 holds for all i.

Equation 5.28 minimizes the deviation of the predicted y_s from y_t for all models in Π and the difference between the steady-state bias $p_{i,s}$ and the model disturbance estimates $p_i(t)$. The purpose of minimizing $y_s - y_t$ is to find the y_s value that is as close to the set point as possible. The minimization of $p_{i,s} - p_i(t)$ allows us to find the unique steady-state biases $p_{i,s}$ that are as close as possible to the model disturbance estimates.

Equation 5.28 is equivalent to the following optimization problem.

$$\min_{X_s} \; X_s^T H_1 X_s - 2(H_2 y_t + H_3 X(t))^T X_s \tag{5.30}$$

subject to

$$GX_s = 0$$

$$L_u X_s \in \mathbb{U}, \; L_x X_s \in \mathbb{X}, \; L_y X_s \in \mathbb{Y}$$

in which

$$X_s = \begin{bmatrix} u_s \\ p_{1,s} \\ \vdots \\ p_{\mathcal{I},s} \end{bmatrix}, \quad X(t) = \begin{bmatrix} x(t) \\ p_1(t) \\ \vdots \\ p_{\mathcal{I}}(t) \end{bmatrix}, \quad H_2 = \begin{bmatrix} -\sum_{i=1}^{\mathcal{I}} G_i^T Q_s \\ (I - A_1)^{-T} C^T Q_s \\ \vdots \\ (I - A_{\mathcal{I}})^{-T} C^T Q_s \end{bmatrix}, \quad H_3 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & P_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{\mathcal{I}} \end{bmatrix},$$

$$H_1 = \begin{bmatrix} \sum_{i=1}^{\mathcal{I}} G_i^T Q_s G_i & G_1^T Q_s C(I - A_1)^{-1} & \cdots & G_{\mathcal{I}}^T Q_s C(I - A_{\mathcal{I}})^{-1} \\ (I - A_1)^{-T} C^T Q_s G_1 & (I - A_1)^{-T} C^T Q_s C(I - A_1)^{-1} + P_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (I - A_{\mathcal{I}})^{-T} C^T Q_s G_{\mathcal{I}} & 0 & \cdots & (I - A_{\mathcal{I}})^{-T} C^T Q_s C(I - A_{\mathcal{I}})^{-1} + P_{\mathcal{I}} \end{bmatrix},$$

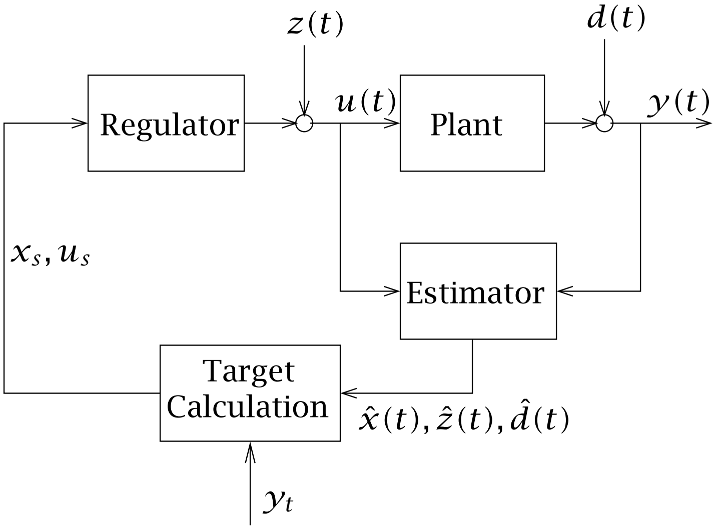

Figure 5.6: Block diagram for the RMPC method

$$G = \begin{bmatrix} (I-A_1)^{-1}B_1 - (I-A_2)^{-1}B_2 & (I-A_1)^{-1} & -(I-A_2)^{-1} & 0 & \cdots & 0 \\ (I-A_1)^{-1}B_1 - (I-A_3)^{-1}B_3 & (I-A_1)^{-1} & 0 & -(I-A_3)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (I-A_1)^{-1}B_1 - (I-A_{\mathcal{J}})^{-1}B_{\mathcal{J}} & (I-A_1)^{-1} & 0 & 0 & \cdots & -(I-A_{\mathcal{J}})^{-1} \end{bmatrix},$$

$$L_u = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}, \quad L_x = \begin{bmatrix} (I-A_1)^{-1}B_1 & (I-A_1)^{-1} & 0 & \cdots & 0 \\ (I-A_2)^{-1}B_2 & 0 & (I-A_2)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (I-A_{\mathcal{J}})^{-1}B_{\mathcal{J}} & 0 & 0 & \cdots & (I-A_{\mathcal{J}})^{-1} \end{bmatrix},$$

$$L_y = \begin{bmatrix} G_1 & C(I-A_1)^{-1} & 0 & \cdots & 0 \\ G_2 & 0 & C(I-A_2)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{\mathcal{J}} & 0 & 0 & \cdots & C(I-A_{\mathcal{J}})^{-1} \end{bmatrix}$$

5.8 Closed-loop Stability

Figure 5.6 is the closed-loop block diagram of the robust MPC method we are proposing, which includes all the elements that affect closed-loop stability. The target calculation computes the steady-state x_s and u_s values before the regulator computes the optimal manipulated input trajectory. If there is no model uncertainty or unmodeled disturbances, the steady-state values do not change over time, and the target calculation does not affect the regulator stability. However, when model uncertainties are present, the model disturbances $p_i(t)$ do not remain constant. They change whenever the estimated state is not equal to the predicted state. When $p_i(t)$ changes, the target calculation needs to be recomputed, which changes the u_s, x_s, y_s, and $p_{i,s}$ values. Because robust MPC considers model uncertainty explicitly in the controller design procedure, the target calculation directly affects the closed-loop stability.

For an unconstrained system, Equation 5.30 is equal to

$$\min_{X_s} \Phi_s = X_s^T H_1 X_s - 2(H_2 y_t + H_3 X(t))^T X_s \tag{5.31}$$

subject to

$$GX_s = 0.$$

The analytical solution for X_s is

$$X_s = H_1^{-1}(H_2 y_t + H_3 X(t)) - H_1^{-1} G^T (G H_1^{-1} G^T)^{-1} G H_1^{-1} (H_2 y_t + H_3 X(t)). \tag{5.32}$$

Since the system is unconstrained, the control law

$$u(t) = K(x(t) - x_s) + u_s \tag{5.33}$$

is a feasible control action. Once $u(t)$ has been implemented, the plant is

$$x(t+1) = A(t)x(t) + B(t)u(t) \tag{5.34}$$

for any $(A(t), B(t))$ in Ω. The disturbance updates are

$$p_i(t+1) = p_{i,s} + L_i(x(t+1) - x_i(t))$$

In terms of the augmented state $X(t+1)$, the plant and disturbance updates are equal to

$$X(t+1) = (\bar{A} + \tilde{L}(\bar{M} + \bar{B}\tilde{K}))X(t) + (\bar{N} + \tilde{L}(\bar{E} + \bar{B}\tilde{K}\bar{J}))X_s \tag{5.35}$$

in which

$$\tilde{L} = \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & L_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_q \end{bmatrix} \quad \text{and} \quad \tilde{K} = \begin{bmatrix} K & 0 & \cdots & 0 \end{bmatrix} \tag{5.36}$$

are the augmented disturbance filter gain and the state feedback gain, respectively. The augmented model matrices are

$$\bar{A} = \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ B - B_1 \\ \vdots \\ B - B_q \end{bmatrix}, \quad \bar{M} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ A - A_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A - A_q & 0 & \cdots & 0 \end{bmatrix},$$

$$\bar{N} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}, \quad \bar{E} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ B - B_1 & -I & 0 & \cdots & 0 \\ B - B_2 & 0 & -I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B - B_q & 0 & 0 & \cdots & -I \end{bmatrix},$$

$$\bar{J} = \begin{bmatrix} -(I - A_1)^{-1}B_1 & -(I - A_1)^{-1} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Equation 5.35 expresses $X(t+1)$ as a function of $X(t)$ and X_s. We substitute Equation 5.32 into Equation 5.35 to express $X(t+1)$ as a function of $X(t)$ and y_t.

$$X(t+1) = (\tilde{A} + \tilde{L}(\tilde{E} + \tilde{B}\tilde{K}\tilde{J}))X(t) + \tilde{G}y_t \tag{5.37}$$

in which

$$\tilde{A} = \bar{A} + \bar{N}\{H_1^{-1}H_3 - H_1^{-1}G^T(GH_1^{-1}G^T)^{-1}GH_1^{-1}H_3\}$$

$$\tilde{B} = \bar{B}$$

$$\tilde{E} = \bar{M} + \bar{E}\{H_1^{-1}H_3 - H_1^{-1}G^T(GH_1^{-1}G^T)^{-1}GH_1^{-1}H_3\}$$

$$\tilde{J} = I + \bar{J}\{H_1^{-1}H_3 - H_1^{-1}G^T(GH_1^{-1}G^T)^{-1}GH_1^{-1}H_3\}$$

$$\tilde{G} = (\bar{N} + \tilde{L}(\bar{E} + \bar{B}\tilde{K}\bar{J}))\{H_1^{-1}H_2 - H_1^{-1}G^T(GH_1^{-1}G^T)^{-1}GH_1^{-1}H_2\}$$

For the vertices of Ω, $A = A_i$ and $B = B_i$. The corresponding augmented matrices are $\tilde{A}_i$, $\tilde{B}_i$, $\tilde{E}_i$, and $\tilde{G}_i$.

Lemma 5.8.1 *If there exist $\tilde{L}$ such that the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle, then the eigenvalues of $\tilde{A} + \tilde{L}(\tilde{E} + \tilde{B}\tilde{K}\tilde{J})$ are within the unit circle for all (A, B) in Ω.*

Proof. $\tilde{A}_i$, $\tilde{E}_i$, and $\tilde{B}_i$ denote the $\tilde{A}$, $\tilde{B}$, and $\tilde{E}$ matrices with $A = A_i$ and $B = B_i$. The state feedback gain K in $\tilde{K}$ is the solution from Equation 5.20. Let $\tilde{C}_i = \tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J}$. If the eigenvalues of $\tilde{A}_i + \tilde{L}\tilde{C}_i$ are within the unit circle, then there exists $\tilde{Z}$ such that

$$\tilde{Z} - (\tilde{A}_i + \tilde{L}\tilde{C}_i)^T\tilde{Z}(\tilde{A}_i + \tilde{L}\tilde{C}_i) > 0 \quad \forall i = 1, \ldots \mathcal{I} \tag{5.38}$$

in which $\tilde{Z} = \tilde{Z}^T > 0$. Let $\tilde{L} = \tilde{Z}^{-1}\tilde{S}$. Equation 5.38 is equal to

$$\tilde{Z} - (\tilde{Z}\tilde{A}_i + \tilde{S}\tilde{C}_i)^T \tilde{Z}^{-1} (\tilde{Z}\tilde{A}_i + \tilde{S}\tilde{C}_i) > 0$$

which implies the following by the Schur complement:

$$\begin{bmatrix} \tilde{Z} & (\tilde{Z}\tilde{A}_i + \tilde{S}\tilde{C}_i)^T \\ (\tilde{Z}\tilde{A}_i + \tilde{S}\tilde{C}_i) & \tilde{Z} \end{bmatrix} > 0 \quad \forall i = 1, \ldots \mathcal{I}. \tag{5.39}$$

Since

$$\tilde{A} = \sum_{i=1}^{\mathcal{I}} \mu_i \tilde{A}_i \qquad \text{and} \qquad \tilde{C} = \sum_{i=1}^{\mathcal{I}} \mu_i \tilde{C}_i,$$

the Schur complement of equation 5.38 for $\tilde{A}$ and $\tilde{C}$ is

$$\sum_{i=1}^{\mathcal{I}} \mu_i \begin{bmatrix} \tilde{Z} & (\tilde{Z}\tilde{A}_i + \tilde{S}\tilde{C}_i)^T \\ (\tilde{Z}\tilde{A}_i + \tilde{S}\tilde{C}_i) & \tilde{Z} \end{bmatrix} > 0$$

which implies

$$\tilde{Z} - (\tilde{A} + \tilde{L}\tilde{C})^T \tilde{Z} (\tilde{A} + \tilde{L}\tilde{C}) > 0 \quad \forall (A, B) \in \Omega.$$

Therefore, there exist $\tilde{L}$ and $\tilde{K}$ such that the eigenvalues of $\tilde{A} + \tilde{L}(\tilde{E} + \tilde{B}\tilde{K}\tilde{J})$ are within the unit circle for all (A, B) in Ω. □

Theorem 5.8.2 *If there exist $\tilde{L}$ and $\tilde{K}$ such that the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle for all (A, B) in Ω, then there exists a control policy that robustly stabilizes the time-varying uncertain system in Equation 5.34. If the plant converges to the time-invariant model (A_∞, B_∞), then the control policy achieves the steady state (x_s, u_s) that minimizes or eliminates the difference between the predicted steady-state output y_s and the output set point y_t.*

Proof. The existence of $\tilde{L}$ such that the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle implies there are finite steady-state x_s and u_s values for every (A, B) in Ω. The existence of F and K such that Equation 5.20 is satisfied for all (A, B) in Ω guarantees convergence of the control policy to x_s and u_s, which are solutions to Equation 5.28. If y_t is reachable, then the addition of the model disturbances $p_i(t)$ provides integral control by eliminating the difference between Cx_s and y_t. At steady state,

$$
\begin{aligned}
x_s &= A_i x_s + B_i u_s + p_i \\
&= A x_s + B u_s + \sum_{i=1}^{\mathcal{I}} \mu_i p_i \quad \forall (A, B) \in \Omega.
\end{aligned}
$$

If the plant converges to (A_∞, B_∞) in Ω, the steady-state equality constraints are

$$
\begin{aligned}
x_t &= A_\infty x_t + B_\infty u_t + p_\infty \\
y_t &= C x_t
\end{aligned}
$$

in which x_t and u_t are the state and input values that correspond to y_t for (A_∞, B_∞). At steady state,

$$
\begin{aligned}
u_s &= K(x_s - x_t) + u_t \\
x_s - x_t &= A_\infty x_s + B_\infty u_s + p_\infty - A_\infty x_t - B_\infty u_t - p_\infty \\
&= A_\infty (x_s - x_t) + B_\infty (u_s - u_t) \\
&= A_\infty (x_s - x_t) + B_\infty K (x_s - x_t) \\
&= (A_\infty + B_\infty K)(x_s - x_t),
\end{aligned}
$$

which implies

$$(A_\infty + B_\infty K - I)(x_s - x_t) = 0$$

$$\text{if and only if } x_s - x_t = 0.$$

If $x_s = x_t$, then $y_s = Cx_s = Cx_t = y_t$. □

5.9 Existence of $\tilde{L}$

Section 5.5 shows the robustly stabilizing K and F can be found by solving an LMI-based optimization problem. Numerical algorithms are available that find the optimal solution to Equation 5.20. The optimization in Theorem 5.8.2 cannot be transformed into an LMI-based optimization problem because of the specific structures of $\tilde{L}$ and $\tilde{K}$. The following numerical algorithm determines if closed-loop stability can be achieve for all (A, B) in Ω. We only need a feasible solution to Equation 5.40 to guarantee closed-loop stability and offset-free non-zero set point tracking.

Algorithm 5.9.1

1. *Find the robustly stabilizing K and F by solving Equation 5.20 and forming $\tilde{K}$ as in Equation 5.36.*

2. *Solve*

$$\min_{L_i} \sum_{i=1}^{\mathcal{I}} \parallel \tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J}) \parallel^2 \tag{5.40}$$

subject to the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ *are within the unit circle for all* $i = 1, \ldots, \mathcal{I}$*.*

The optimization in Step 2 of Algorithm 5.9.1 is a difficult non-convex problem. Nonlinear solvers may fail to find a feasible $\tilde{L}$ even though a feasible solution exists. In order to devise a solution strategy, we note that $L_i = 0$ for all $i = 1, \ldots, \mathcal{I}$ is a feasible solution if the integrators are pre-multiplied by a factor less than one.

$$p_i(t+1) = d_i p_{i,s} + L_i(x(t+1) - x_i(t)) \tag{5.41}$$

in which $d_i \in \mathbb{R}$ and $0 \le d_i < 1$ for all i. The steady-state equality constraints for the target calculation are

$$x_s = A_i x_s + B_i u_s + d_i p_{i,s} \quad \text{for } i = 1, \ldots, \mathcal{I},$$

and the state forecast is

$$x_{\tau,i}(t) = A_i x_\tau(t) + B_i u_\tau(t) + d_i p_{i,s}$$

With the above modifications, if $L_i = 0$ for all i, the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ that correspond to the integrating modes are equal to d_i. If d_i is less than one, $L_i = 0$ is a feasible solution. The nonlinear optimization solvers are more likely to find a solution to Equation 5.40 if the initial guess is feasible. We propose, therefore, the following algorithm that involves an iterative procedure beginning with $0 \le d_i < 1$. The $\tilde{L}$ that satisfies Theorem 5.8.2 is found when $d_i = 1$ for all $i = 1, \ldots, \mathcal{I}$.

Algorithm 5.9.2

1. *Find the robustly stabilizing K and F by solving Equation 5.20 and forming $\tilde{K}$ as in Equation 5.36.*

2. *Let $d_i < 1$ and $L_i = 0$ for all $i = 1, \ldots, \mathcal{I}$. Let*
$$\tilde{L}_0 = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & L_1 & 0 & \cdots & 0 \\ 0 & 0 & L_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & L_{\mathcal{I}} \end{bmatrix}$$
be the initial guess.

3. *Solve*
$$\min_{L_i} \sum_{i=1}^{\mathcal{I}} \parallel \tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J}) \parallel^2 + \rho \parallel \tilde{L} - \tilde{L}_0 \parallel^2 \tag{5.42}$$
subject to the eigenvalues of $\tilde{A}_i + \tilde{L}(\tilde{E}_i + \tilde{B}_i\tilde{K}\tilde{J})$ are within the unit circle. The weighting factor $\rho \in \mathbb{R}$ is a user defined variable that decides how close the solution $\tilde{L}$ is to the initial guess $\tilde{L}_0$.

4. *If the nonlinear optimizer returns an optimal solution, increase d_i and let $\tilde{L}_0$ be equal to the solution of Equation 5.42. Go to Step 5. If the nonlinear optimizer*

cannot find a solution, decrease d_i and repeat Step 3.

5. *Solve Equation 5.42. If the solution is optimal and $d_i = 1$, STOP. If $d_i < 1$, go to Step 3 and repeat. If no solution exists or the solution is not optimal, decrease d_i and repeat Step 3.*

Algorithm 5.9.2 successfully found solutions for systems with various numbers of states and models. The largest problem we solved had 12 states and three models. As the numbers of states and models increase, the time for Algorithm 5.9.2 to find a solution increases, with the number of states having a larger effect. The solution to Algorithm 5.9.2 is determined off-line. The computation time does not affect the RMPC method's on-line implementation feasibility. The failure of Algorithm 5.9.2 to find a solution to Equation 5.42 for $d_i = 1$, and $i = 1, \ldots, \mathcal{I}$ does *not* imply no $\tilde{L}$ exists such that Theorem 5.8.2 holds. However, Lemma 5.9.3 discusses a case in which no $\tilde{L}$ exists such that Theorem 5.8.2 holds.

Lemma 5.9.3 *Suppose $p = m$, Equation 5.29, and $y_s = y_t$. The controlled output set point is reachable if the rank of the gain matrix is equal to p for all (A, B) in Ω, and there exist u_s and x_s such that Cx_s is equal to y_t.*

Proof. The gain of (A, B) is $G \in \mathbb{R}^{p \times m}$.

$$G = C(I - A)^{-1}B$$

At steady state,

$$y_t = Gu_s.$$

For every y_t, a solution exists if and only if the rank of G is equal to p. A unique solution exists if and only if the rank of G is equal to p and $p = m$. □

If Ω contains a model that is rank deficient, no $\tilde{L}$ exists such that Equation 5.42 holds for $d_i = 1$ because there is no x_s and u_s values such that Cx_s is equal to y_t.

5.10 Summary of the RMPC Method

In this section, we provide a summary of the steps needed in the new RMPC method.

1. Find the robustly stabilizing K and F by solving Equation 5.20, an LMI-based optimization.

2. Use Algorithm 5.9.2 to find the augmented disturbance filter gain $\tilde{L}$, which requires satisfaction of a set of inequalities.

3. Find the steady-state u_s, x_s, and $p_{i,s}$ values by solving Equation 5.28, a quadratic program.

4. Find the sequence of control actions $U^*(t)$ by solving Equation 5.15, a *min-max* optimization.

5. Inject the control action $u(t) = \overline{u}^*(t) + u_s$ from $U^*(t)$ into the plant.

6. Measure $y(t+1)$ and construct $x(t+1)$ as shown in Equation 5.24.

7. Use Equation 5.26 to update the model disturbances $p_i(t)$.

8. Go to Step 3 and repeat.

Steps 1 and 2 are performed off-line to establish closed-loop stability for offset-free control of the time-varying system described by Equations 5.1 and 5.2. Steps 3-7 are computed on-line at each time step.

5.11 Conclusion

In this chapter, we propose a new RMPC method that includes a closed-loop stability condition that determines if offset-free control is possible given the model uncertainty description. The stability condition considers the effects of the robust regulator, target calculation, and state estimation on the closed-loop stability. Offset-free control in the presence of model uncertainty is accomplished by adding steady-state biases to the steady-state equality constraints so that for a single u_s value, all models in the model uncertainty description have the same steady-state x_s value. The target calculation determines the x_s and u_s values necessary to achieve offset-free control. The tree trajectory in the robust regulator is designed to forecast the state for a system with time-varying uncertainty. Offset-free control is achieved when all branches in the tree trajectory converge to the x_s and u_s values.

5.12 Appendix

5.12.1 Additional Definitions Needed for Theorem 5.4.1 and 5.4.2 Proofs

D19: If $a \in \mathbb{I}$ and $b \in \mathbb{I}$, then the operation $\Re(a, b)$ finds the remainder of a divided by b.

D20: Let $\overline{\overline{T}}_l$ be a subset of $\overline{T}_l$ and be defined as

$$\overline{\overline{T}}_l = \overline{T}_l \setminus \mathbf{P}(\tau_l) \quad \text{for} \quad l = 1, \ldots, \mathcal{L} \tag{5.43}$$

The set $\overline{\overline{T}}_l$ contains all the nodes in T_l except for the terminal node τ_l and the second to last node $\mathbf{P}(\tau_l)$.

D21: The set Γ contains the nodes on all branches $l = 1, \ldots, \mathcal{L}$ and is defined as

$$\Gamma = \bigcup_{l=1}^{\mathcal{L}} T_l \tag{5.44}$$

D22: The subset Γ_N contains all the terminal nodes τ_l, the nodes on level N, and is defined as

$$\Gamma_N = \{\tau_l | l = 1, \ldots, \mathcal{L}\} \tag{5.45}$$

D23: The subset Γ_{N-1} contains the nodes on level $N - 1$, the second to last level, and is defined as

$$\Gamma_{N-1} = \{\mathbf{P}(\tau_l) | l = 1, \ldots, \mathcal{L}\} \tag{5.46}$$

D24: The subset $\overline{\Gamma}$ contains all nodes on the tree trajectory except for the terminal nodes and is defined as

$$\overline{\Gamma} = \Gamma \setminus \Gamma_N \tag{5.47}$$

D25: The subset $\overline{\overline{\Gamma}}$ contains all nodes on the tree trajectory except for the nodes on levels N and $N-1$. The subset is defined as

$$\overline{\overline{\Gamma}} = \overline{\Gamma} \setminus \Gamma_{N-1} \tag{5.48}$$

D26: If $i \in \mathbb{I}$ is the model index, then S_i denotes the set of branch indices such that

$$S_i = \{l | \mathbf{P}^{N-1}(\tau_l) = i, l = 1, \ldots, \mathcal{L}\} \tag{5.49}$$

D27: If $\sigma \in \mathbb{I}$, $l \in \mathbb{I}$, $i = 1, \ldots, \mathcal{I}$, and $c = \Re(l, \mathcal{I}^{N-1})$ for $l \in S_i$, then the set of branch indices $\overline{S}_l$ is defined as

$$\overline{S}_l = \left\{ \sigma \;\middle|\; \begin{array}{lll} \mathcal{I}^N - \mathcal{I} + 1 \leq \sigma \leq \mathcal{I}^N & \text{if} & c = 0 \\ \mathcal{I}(c-1) + 1 \leq \sigma \leq \mathcal{I}c & \text{if} & c \neq 0 \end{array} \right\} \tag{5.50}$$

D28: If $l \in \mathbb{I}$ and $i = 1, \ldots, \mathcal{I}$, the set S is defined as

$$S = \bigcup_{l \in S_i} \overline{S}_l = 1, \ldots, \mathcal{L} \tag{5.51}$$

5.12.2 Proof for Theorem 5.4.1

At time t and state $\overline{x}^*(t)$, the solution to Equation 5.15 is $\Phi^*(t)$. The optimal control sequence is $U^*(t)$. The optimal control action $\overline{u}(t) = \overline{u}^*(t)$ is implemented, and the state at time $t+1$ is

$$\begin{aligned}\overline{x}(t+1) &= A_{i*}\overline{x}^*(t) + B_{i*}\overline{u}^*(t) \\ &= \overline{x}_{i*}(t)\end{aligned} \tag{5.52}$$

when model i^* is the plant at time t. Consider the following feasible but non-optimal control actions at time $t+1$:

$$\overline{u}(t+1) = \overline{u}_{i*}(t) \tag{5.53}$$

$$\overline{u}_\tau(t+1) = \overline{u}_{i*,\tau}(t) \quad \text{for} \quad \tau \in \overline{\overline{\Gamma}} \tag{5.54}$$

$$\overline{u}_\tau(t+1) = K\overline{x}_{i*,\tau}(t) \quad \text{for} \quad \tau \in \Gamma_{N-1} \tag{5.55}$$

The control sequence at time $t+1$ is

$$U_\sigma(t+1) = \left\{\overline{u}(t+1), \{\overline{u}_\tau(t+1)|\tau \in \overline{T}_\sigma\}\right\} \text{ for } \sigma = 1,\ldots,\mathcal{L} \tag{5.56}$$

Equation 5.10 shows the state forecasts for the tree trajectory. The state at node $\tau = i$ and time $t+1$ is

$$\begin{aligned}\overline{x}_i(t+1) &= A_i\overline{x}(t+1) + B_i\overline{u}(t+1) \\ &= A_i\overline{x}_{i*}(t) + B_i\overline{u}_{i*}(t) \\ &= \overline{x}_{i*,i}(t)\end{aligned} \tag{5.57}$$

The state at node τ and time $t+1$ is

$$\overline{x}_\tau(t+1) = \overline{x}_{i*,\tau}(t) \quad \text{for} \quad \tau \in \overline{\Gamma} \tag{5.58}$$

The performance objective for branch l at time t with $\overline{u}(t) = \overline{u}^*(t)$ is

$$\Phi_l^*(t) = \phi(\overline{x}^*(t), \overline{u}^*(t)) + \sum_{\tau \in \overline{T}_l} \phi\left(\overline{x}_\tau(t), \overline{u}_\tau(t)\right) + \phi_N(\overline{x}_{\tau_l}(t)) \quad \text{for} \quad l = 1, \ldots, \mathcal{L} \tag{5.59}$$

For the branches in S_{i^*}, Equation 5.59 is equivalent to

$$\begin{aligned}\Phi_l^*(t) &= \phi(\overline{x}^*(t), \overline{u}^*(t)) + \phi(\overline{x}_{i^*}(t), \overline{u}_{i^*}(t)) + \sum_{\tau \in \overline{\overline{T}}_\sigma} \phi(\overline{x}_{i^*,\tau}(t), \overline{u}_{i^*,\tau}(t)) \\ &+ \phi_N(\overline{x}_{i^*,\mathbf{P}(\tau_\sigma)}(t)) \quad \text{for} \;\; \sigma \in \overline{S}_l \;\; \text{and} \;\; l \in S_{i^*}\end{aligned} \tag{5.60}$$

After substituting Equations 5.53, 5.54, and 5.58 into Equation 5.60,

$$\begin{aligned}\Phi_l^*(t) &= \phi(\overline{x}^*(t), \overline{u}^*(t)) + \phi(\overline{x}(t+1), \overline{u}(t+1)) + \sum_{\tau \in \overline{\overline{T}}_\sigma} \phi(\overline{x}_\tau(t+1), \overline{u}_\tau(t+1)) \\ &+ \phi_N(\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1)) \quad \text{for} \;\; \sigma \in \overline{S}_l \text{and} \;\; l \in S_{i^*}\end{aligned} \tag{5.61}$$

The non-optimal performance objective for branch σ at time $t+1$ is

$$\begin{aligned}\Phi_\sigma(t+1) &= \phi(\overline{x}(t+1), \overline{u}(t+1)) + \sum_{\tau \in \overline{\overline{T}}_\sigma} \phi(\overline{x}_\tau(t+1), \overline{u}_\tau(t+1)) \\ &+ \phi(\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1), \overline{u}_{\mathbf{P}(\tau_\sigma)}(t+1)) + \phi_N(\overline{x}_{\tau_\sigma}(t+1)) \;\; \text{for} \;\; \sigma = 1, \ldots, \mathcal{L}\end{aligned} \tag{5.62}$$

For node $\tau \in \Gamma_{N-1}$ at time $t+1$, Equation 5.55 is a feasible control action. The state at $\tau_\sigma \in \Gamma_N$ is

$$\begin{aligned}\overline{x}_{\tau_\sigma}(t+1) &= \overline{x}_{\mathbf{P}(\tau_\sigma),i}(t+1) \\ &= A_i \overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1) + B_i \overline{u}_{\mathbf{P}(\tau_\sigma)}(t+1) \\ &= A_i \overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1) + B_i K \overline{x}_{i^*,\mathbf{P}(\tau_\sigma)}(t) \\ &= (A_i + B_i K)\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1) \quad \text{for} \;\; i = 1, \ldots, \mathcal{I}\end{aligned} \tag{5.63}$$

After substituting Equation 5.63 into Equation 5.62,

$$\begin{aligned}\Phi_\sigma(t+1) \quad &= \phi(\overline{x}(t+1), \overline{u}(t+1)) + \sum_{\tau \in \overline{\overline{T}}_\sigma} \phi(\overline{x}_\tau(t+1), \overline{u}_\tau(t+1)) \\ &+\phi(\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1), K\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1)) \\ &+\phi_N((A_i+B_iK)\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1))\end{aligned} \tag{5.64}$$

The difference between Equations 5.61 and 5.64 is

$$\begin{aligned}&\Phi_l^*(t) - \Phi_\sigma(t+1) \\ &= \phi(\overline{x}^*(t), \overline{u}^*(t)) \\ &+x_{\mathbf{P}(\tau_\sigma)}(t+1)^T(F - Q - K^TRK - (A_i+B_iK)^TF(A_i+B_iK))x_{\mathbf{P}(\tau_\sigma)}(t+1) \\ &\text{for } \ \sigma \in \overline{S}_l \ \text{ and} l \in S_{i^*}\end{aligned}$$

If there exist F and K such that

$$F - Q - K^TRK - (A_i+B_iK)^TF(A_i+B_iK) \geq 0 \quad \text{for } \ i = 1, \ldots, \mathcal{I}$$

then

$$\Phi_l^*(t) - \Phi_\sigma(t+1) \geq 0 \quad \text{for } \ \sigma \in \overline{S}_l \ \text{ and } \ l \in S_{i^*} \tag{5.65}$$

The optimal performance objective at time t is the maximum of the $\Phi_l^*(t)$, so

$$\Phi_\sigma(t+1) \leq \Phi^*(t) \quad \text{for } \ \sigma \in S \tag{5.66}$$

Equation 5.56 is a feasible but non-optimal control sequence. Recall Equation 5.17. $\Phi_\sigma(t+1; U_\sigma(t+1))$ denotes the performance objective at time $t+1$ using the control sequence $U_\sigma(t+1)$ in Equation 5.56. Let $\Phi_\sigma(t+1; U'_\sigma(t+1))$ denote the performance objective at time $t+1$ using some other control sequence $U'_\sigma(t+1)$. We define the function

$$\Psi(U'_\sigma(t+1)) = \max_{\sigma \in S} \Phi_\sigma(t+1; U'_\sigma(t+1)) \tag{5.67}$$

The value of $\Psi(U_\sigma(t+1))$ is

$$\Psi(U_\sigma(t+1)) = \max_{\sigma \in S} \Phi_\sigma(t+1; U_\sigma(t+1)) \tag{5.68}$$

The *min-max* optimization

$$\min_{U'_\sigma(t+1)} \max_{\sigma \in S} \Phi_\sigma(t+1; U'_\sigma(t+1)) = \min_{U'_\sigma(t+1)} \Psi(U'_\sigma(t+1)) \tag{5.69}$$

has a solution $U^*(t+1)$ which implies

$$\Psi(U^*(t+1)) \leq \Psi(U_\sigma(t+1)) \tag{5.70}$$

Therefore,

$$\min_{U'_\sigma(t+1)} \max_{\sigma \in S} \Phi_\sigma(t+1; U'_\sigma(t+1)) \leq \max_{\sigma \in S} \Phi_\sigma(t+1; U_\sigma(t+1)) \tag{5.71}$$

$$\Longrightarrow \quad \Phi^*(t+1; U^*(t+1)) \leq \Phi^*(t; U^*(t)) \tag{5.72}$$

in which $\Phi^*(t+1; U^*(t+1))$ is the optimal performance objective at time $t+1$. Equation 5.72 indicates $\Phi^*(t; U^*(t))$ is a non-increasing sequence bounded below by zero and therefore converges.

5.12.3 Proof for Theorem 5.4.2

At time $t+1$, the state is

$$\begin{aligned}\overline{x}(t+1) &= \sum_{i=1}^{\mathcal{I}} \mu_i(t) A_i \overline{x}^*(t) + \sum_{i=1}^{\mathcal{I}} \mu_i(t) B_i \overline{u}^*(t) \\ &= \sum_{i=1}^{\mathcal{I}} \mu_i(t)(A_i \overline{x}(t) + B_i \overline{u}^*(t)) \\ &= \sum_{i=1}^{\mathcal{I}} \mu_i(t) \overline{x}_i(t) \qquad (5.73)\end{aligned}$$

if the convex combination $(\sum_{i=1}^{\mathcal{I}} \mu_i(t) A_i, \sum_{i=1}^{\mathcal{I}} \mu_i(t) B_i)$ is the plant. The control action $\overline{u}(t) = \overline{u}^*(t)$ implemented is from the optimal control sequence $U^*(t)$. $\overline{x}(t+1)$ is a linear convex combination of the states at node $\tau = i$ and time t. Consider the following feasible but non-optimal candidate control actions:

$$\overline{u}(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) \overline{u}_i(t) \qquad (5.74)$$

$$\overline{u}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) \overline{u}_{i,\tau}(t) \quad \text{for} \quad \tau \in \overline{\overline{\Gamma}} \qquad (5.75)$$

$$\overline{u}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) K \overline{x}_{i,\tau}(t) \quad \text{for} \quad \tau \in \Gamma_{N-1} \qquad (5.76)$$

A control sequence at time $t+1$ is

$$U_\sigma(t+1) = \left\{\overline{u}(t+1), \{\overline{u}_\tau(t+1) | \tau \in \overline{T}_\sigma\}\right\} \quad \text{for } \sigma = 1, \ldots, \mathcal{L} \qquad (5.77)$$

We claim the state at node τ and time $t+1$ is

$$\overline{x}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t) \overline{x}_{i,\tau}(t) \qquad (5.78)$$

Proof for the above claim is in Section 5.12.4. The performance objective for branch σ is

$$\begin{aligned}\Phi_\sigma(t+1) &= \phi(\overline{x}(t+1),\overline{u}(t+1)) + \sum_{\tau\in\overline{\overline{T}}_\sigma} \phi(\overline{x}_\tau(t+1),\overline{u}_\tau(t+1)) \\ &+\phi(\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1),\overline{u}_{\mathbf{P}(\tau_\sigma)}(t+1)) \\ &+\phi_N(\overline{x}_{\tau_\sigma}(t+1)) \end{aligned} \tag{5.79}$$

in which

$$\begin{aligned}\overline{u}_{\mathbf{P}(\tau_\sigma)}(t+1) &= K\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1) \\ \overline{x}_{\tau_\sigma}(t+1) &= (A_k+B_kK)\overline{x}_{\mathbf{P}(\tau_\sigma)}(t+1) \quad \text{for } k=1,\ldots,\mathcal{I}\end{aligned}$$

After substituting Equations 5.74-5.76 and 5.78 into Equation 5.79,

$$\begin{aligned}\Phi_\sigma(t+1) &= \phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{x}_i(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{u}_i(t)\right) \\ &+ \sum_{\tau\in\overline{\overline{T}}_\sigma}\phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{x}_{i,\tau}(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{u}_{i,\tau}(t)\right) \\ &+\phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)K\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\right) \\ &+\phi_N\left((A_k+B_kK)\sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t)\right)\end{aligned} \tag{5.80}$$

By the convexity of the function $\phi(x,u)$ and $0 \le \mu_i(t) \le 1$,

$$\begin{aligned}&\phi\left(\sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{x}_{i,\tau}(t), \sum_{i=1}^{\mathcal{I}}\mu_i(t)\overline{u}_{i,\tau}(t)\right) \\ &\le \sum_{i=1}^{\mathcal{I}}\phi\left(\mu_i(t)\overline{x}_{i,\tau}(t),\mu_i(t)\overline{u}_{i,\tau}(t)\right) = \sum_{i=1}^{\mathcal{I}}\mu_i(t)^2\phi\left(\overline{x}_{i,\tau}(t),\overline{u}_{i,\tau}(t)\right) \\ &\le \sum_{i=1}^{\mathcal{I}}\mu_i(t)\phi\left(\overline{x}_{i,\tau}(t),\overline{u}_{i,\tau}(t)\right) \quad \text{for } \tau\in\overline{T}_\sigma \text{ and } \sigma=1,\ldots,\mathcal{L}\end{aligned} \tag{5.81}$$

Let

$$\begin{aligned}\Phi_{\sigma,i}(t+1) \quad &= \phi(\overline{x}_i(t),\overline{u}_i(t)) + \sum_{\tau\in\overline{\overline{T}}_\sigma} \phi(\overline{x}_{i,\tau}(t),\overline{u}_{i,\tau}(t)) \\ &+\phi(\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t),K\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t)) + \phi_N((A_k+B_kK)\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t)) \quad (5.82) \\ &\text{for } \sigma\in\overline{S}_l,\ \ l\in S_i,\ \ i=1,\ldots,\mathcal{I}\end{aligned}$$

in which $\Phi_{\sigma,i}(t+1)$ is the performance objective at $t+1$ for branch σ with $\mathbf{P}^{N-1}(\tau_\sigma) = i$. After substituting Equation 5.81 into Equation 5.80, the performance objective becomes

$$\begin{aligned}&\Phi_\sigma(t+1) \\ &\leq \sum_{i=1}^{\mathcal{I}} \mu_i(t) \Bigg[\phi(\overline{x}_i(t),\overline{u}_i(t)) + \sum_{\tau\in\overline{\overline{T}}_\sigma} \phi(\overline{x}_{i,\tau}(t),\overline{u}_{i,\tau}(t)) \\ &+\phi(\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t),K\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t)) + \phi_N((A_k+B_kK)\overline{x}_{i,\mathbf{P}(\tau_\sigma)}(t))\Big] \\ &= \sum_{i=1}^{\mathcal{I}} \mu_i(t)\Phi_{\sigma,i}(t+1) \qquad (5.83)\end{aligned}$$

If there exists K and F such that Theorem 5.4.1 holds, then

$$\Phi_{\sigma,i}(t+1) \leq \Phi^*(t) \quad \text{for } \sigma\in\overline{S}_l,\ \ l\in S_i \ \text{ and } \ i=1,\ldots,\mathcal{I} \qquad (5.84)$$

$$\Longrightarrow \quad \sum_{i=1}^{\mathcal{I}} \mu_i(t)\Phi_{\sigma,i}(t+1) \leq \sum_{i=1}^{\mathcal{I}} \mu_i(t)\Phi^*(t) \qquad (5.85)$$

By combining Equations 5.83, 5.82 and 5.85,

$$\Phi_\sigma(t+1) \leq \sum_{i=1}^{\mathcal{I}} \mu_i(t)\Phi_{\sigma,i}(t+1) \leq \Phi^*(t) \quad \text{for } \sigma\in S \qquad (5.86)$$

Equation 5.77 is a feasible but non-optimal control sequence. Recall Equation 5.17. $\Phi_\sigma(t+1;U_\sigma(t+1))$ denotes the performance objective at time $t+1$ using the control sequence $U_\sigma(t+1)$ in Equation 5.77. Let $\Phi_\sigma(t+1; U'_\sigma(t+1))$ denote the performance objective at time $t+1$ using some other control sequence $U'_\sigma(t+1)$. We define the function

$$\Psi(U'_\sigma(t+1)) = \max_{\sigma\in S} \Phi_\sigma(t+1; U'_\sigma(t+1)) \tag{5.87}$$

The value of $\Psi(U_\sigma(t+1))$ is

$$\Psi(U_\sigma(t+1)) = \max_{\sigma\in S} \Phi_\sigma(t+1; U_\sigma(t+1)) \tag{5.88}$$

The *min-max* optimization

$$\min_{U'_\sigma(t+1)} \max_{\sigma\in S} \Phi_\sigma(t+1; U'_\sigma(t+1)) = \min_{U'_\sigma(t+1)} \Psi(U'_\sigma(t+1)) \tag{5.89}$$

has a solution $U^*(t+1)$ which implies

$$\Psi(U^*(t+1)) \leq \Psi(U_\sigma(t+1)) \tag{5.90}$$

Therefore,

$$\min_{U'_\sigma(t+1)} \max_{\sigma\in S} \Phi_\sigma(t+1; U'_\sigma(t+1)) \leq \max_{\sigma\in S} \Phi_\sigma(t+1; U_\sigma(t+1)) \tag{5.91}$$

$$\Longrightarrow \quad \Phi^*(t+1; U^*(t+1)) \leq \Phi^*(t; U^*(t)) \tag{5.92}$$

which indicates $\Phi^*(t; U^*(t))$ is a non-increasing sequence bounded below by zero and therefore converges. In other words,

$$\begin{aligned} & \Phi^*(t; U^*(t)) \longrightarrow 0 \quad \text{as} \quad t \longrightarrow \infty \\ \Longrightarrow \quad & \overline{x}_\tau(t) \longrightarrow 0 \quad \text{and} \quad \overline{u}_\tau(t) \longrightarrow 0 \\ \Longrightarrow \quad & x_\tau(t) \longrightarrow x_s \quad \text{and} \quad u_\tau(t) \longrightarrow u_s. \end{aligned}$$

5.12.4 State at $t + 1$ as Function of State at t

Equation 5.73 shows the state at $t + 1$ is

$$\overline{x}(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t)\overline{x}_i(t) \tag{5.93}$$

The state at node $\tau = k$ after applying the candidate control action in Equation 5.74 is

$$\begin{aligned} \overline{x}_k(t+1) \quad &= A_k\overline{x}(t+1) + B_k\overline{u}(t+1) \\ &= \sum_{i=1}^{\mathcal{I}} \mu_i(t)(A_k\overline{x}_i(t) + B_k\overline{u}_i(t)) \\ &= \sum_{i=1}^{\mathcal{I}} \mu_i(t)\overline{x}_{i,k}(t) \end{aligned} \tag{5.94}$$

After carrying the state evolution further into the tree trajectory, we show

$$\overline{x}_\tau(t+1) = \sum_{i=1}^{\mathcal{I}} \mu_i(t)\overline{x}_{i,\tau}(t) \quad \text{for} \ \ \tau \in \overline{\Gamma} \tag{5.95}$$

Chapter 6

Results and Discussion

6.1 Introduction

This chapter presents some closed-loop simulation results using the RMPC method proposed in Chapter 5. The simulations show robust offset-free integral control in the presence of model uncertainty and successful disturbance rejection without input wind-up. The RMPC control performance is compared to the single-model MPC (SMPC) control performance using various models in the uncertainty description to show the advantages and disadvantages of the new RMPC method. We compare the RMPC control performance to the SMPC performance because of the popularity of the SMPC controller in industry applications. SMPC's explicit use of process models to make process predictions is conceptually appealing to industrial practitioners. SMPC's ability to explicitly handle input and output constraints in the optimal control problem increases its attractiveness even further.

SMPC also has the added advantage of the ability to achieve offset-free non-zero set point tracking without wind-up. Although tuning can increase SMPC's robustness to some modeling errors, SMPC is not designed to find the control policy that can control a large class of models. On the other hand, RMPC has all the same features as SMPC, except that RMPC explicitly accounts for model uncertainty in the design procedure. RMPC achieves offset-free non-zero set point tracking by adding a state disturbance to every model in Π. Except when noted, the SMPC controller uses the output disturbance model to achieve integral control (see Muske and Rawlings [110] or Section 2.5 for more detail).

We can try to increase the robustness of the nominal SMPC controller to model uncertainty by detuning and checking closed-loop stability for all models in the model uncertainty description. This method is rather ad hoc. It does not use the additional information provided by the model uncertainty description to get more accurate model forecasts. We know the control performance is not optimal because the model forecasts are not equal to the system behavior. We do not claim the new RMPC method computes the optimal control actions, but the new RMPC controller will compute the best control actions for the sequence of time-varying models with the largest deviations from x_s and u_s. The tree trajectory provides a fuller description of the uncertain system's behavior by forecasting the time-varying uncertainty. Even though the system behavior may not be equal to the forecast of any specific branch, the system behavior can be described by a convex linear combination of the branches. The increase in robustness provided by the tree trajectory is caused by a more accurate model forecast and not a study in SMPC tuning rules. Except when noted, the examples presented use the same tuning parameters for both the RMPC and SMPC controllers to provide the most accurate comparison.

The Matlab LMI Toolbox is used to solve the LMI-based optimization problem that finds the robustly stabilizing K and F. The nonlinear optimization program NPSOL is used to find the optimal $\tilde{L}$ that guarantees closed-loop stability. Matlab's Optimization Toolbox is used to solve the *min-max* optimization in the dynamic regulator.

6.2 Single-Input-Single-Output Examples

6.2.1 Unconstrained single state examples

$\Pi_1 = \{(A_1, B_1), (A_2, B_2)\}$ is the set of uncertain models for the first series of simulations.

$$\text{Model 1}: \qquad x(t+1) = 0.4x(t) + 0.1u(t)$$

$$\text{Model 2}: \qquad x(t+1) = 0.4x(t) + 0.7u(t)$$

The LMI optimizer finds K and F that robustly stabilize Ω_1, the polytope of Π_1. Figures 6.1 and 6.2 compare the closed-loop performance of the RMPC to the SMPC with plant-model mismatch for a set point change from $x_t = 0$ to $x_t = 1$.

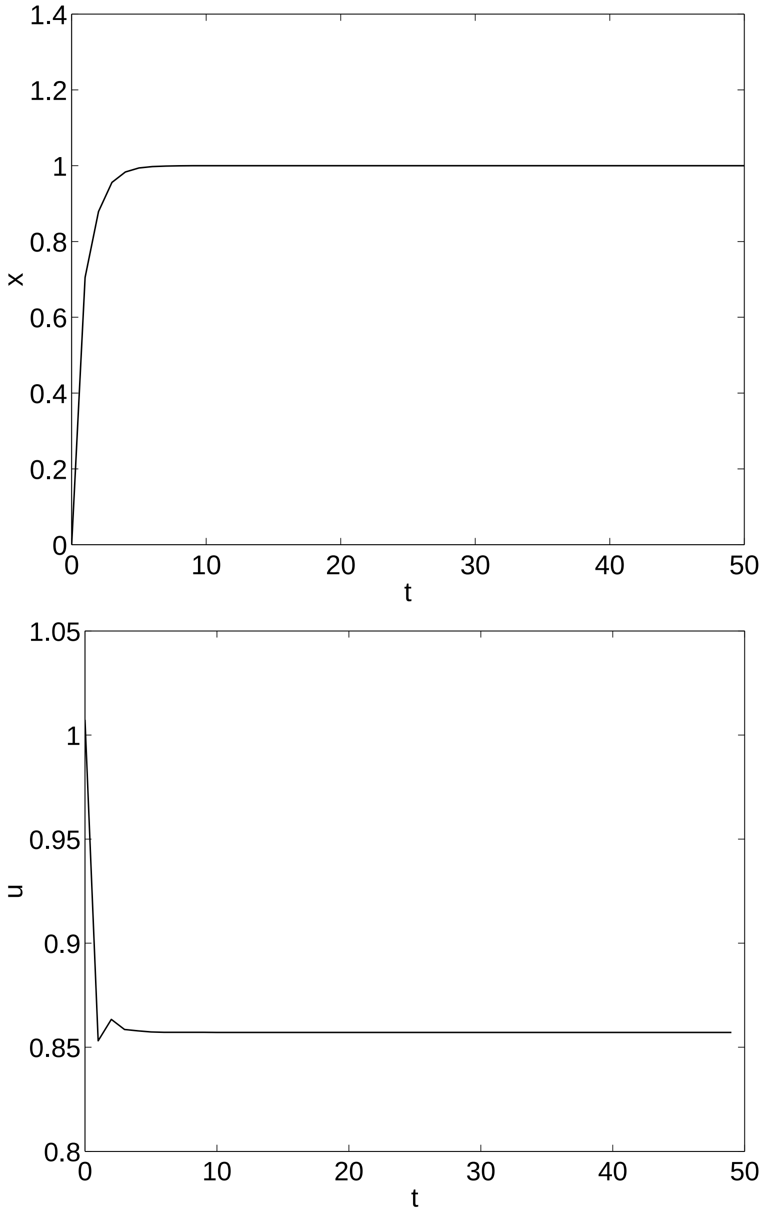

Figure 6.1: RMPC performance for the model uncertainty description Ω_1 with the plant $A_p = 0.4$, $B_p = 0.7$.

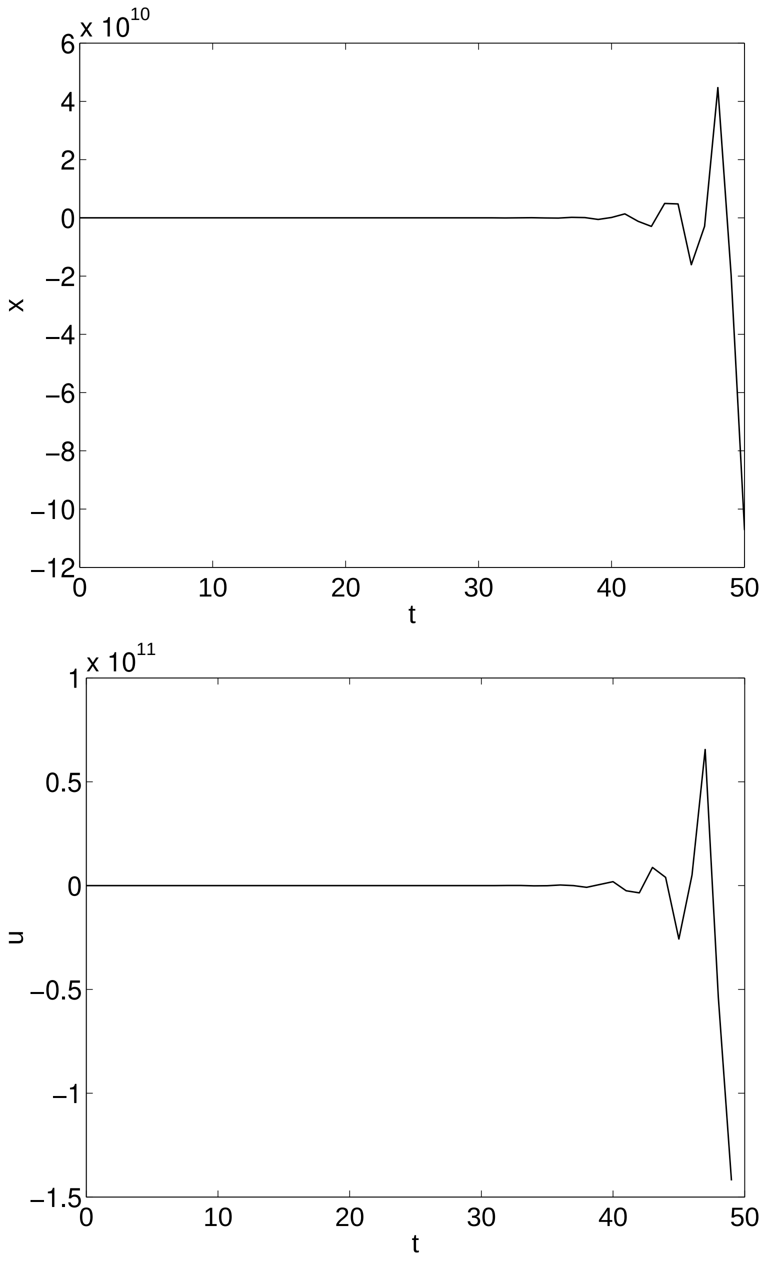

Figure 6.2: SMPC performance with plant model mismatch, $A_m = 0.4$, $B_m = 0.1$ and $A_p = 0.4$, $B_p = 0.7$.

The plant is model 2. The SMPC algorithm uses model 1 to make the open-loop predictions. Figure 6.1 shows the RMPC controller successfully controls the system for the set point change. However, the SMPC performance (Figure 6.2) is unstable.

The second set of uncertain models we examined are robustly stabilizable and closed-loop stable using the SMPC with plant-model mismatch. The models are $\Pi_2 = \{(0.4, 0.1), (0.4, 0.4)\}$ with model uncertainty description Ω_2. Figure 6.3 compares the robust and the single-model MPC controllers' closed-loop performances. The plant is model 2, but the SMPC algorithm model is model 1. Both controllers successfully control the system from the original set point $x_t = 0$ to the new set point $x_t = 1$, but the RMPC controller performance is more efficient. Plant-model mismatch causes oscillatory SMPC performance, which is undesirable.

We have shown the RMPC performance is better than the SMPC performance with plant-model mismatch. We now compare closed-loop performances for the robust and the single-model MPC controllers without plant-model mismatch. Figure 6.4 is the state and input closed-loop responses for model 2, the plant and the single-model MPC algorithm model. The robust MPC performance is almost identical to the nominal performance, which is the best possible control performance. Figure 6.5 compares the robust and the nominal performance for model 1, the plant. The robust controller performance is more conservative than

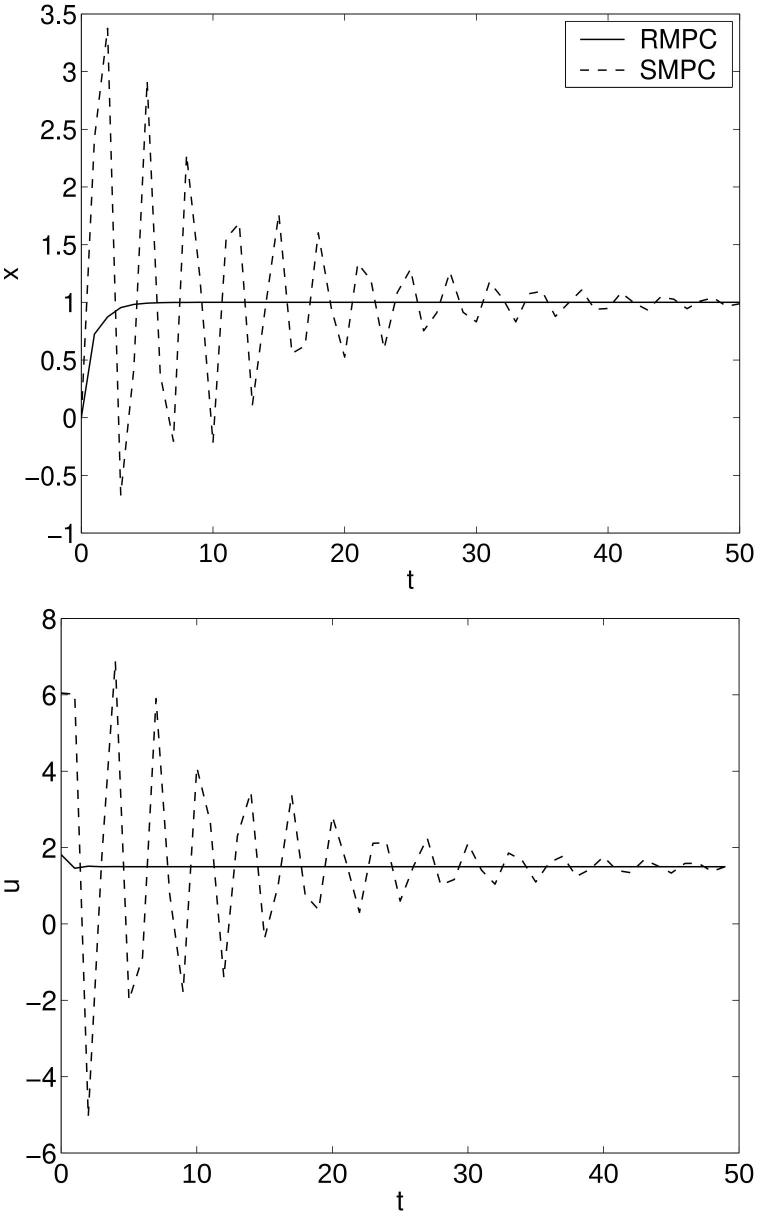

Figure 6.3: Comparison of RMPC and SMPC performance with plant model mismatch. $A_m = 0.4$, $B_m = 0.1$ for SMPC and $A_p = 0.4$, $B_p = 0.4$.

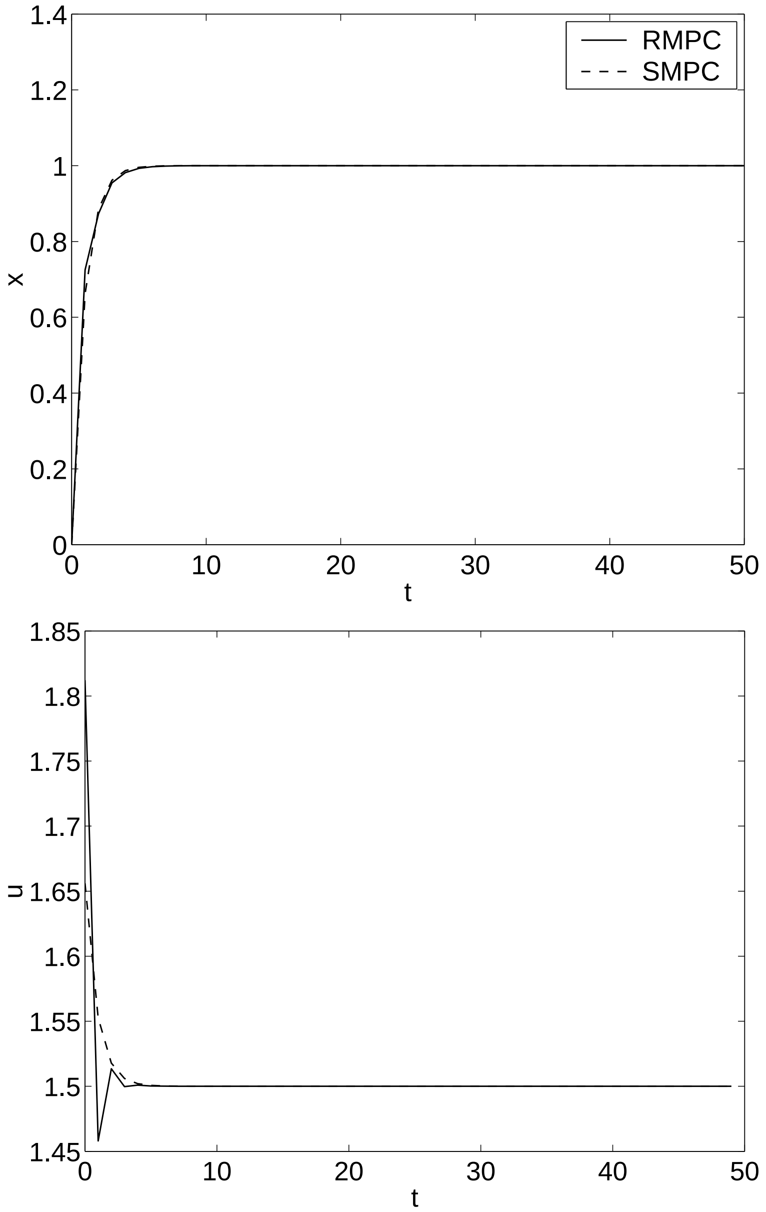

Figure 6.4: Comparison of the RMPC performance with uncertainty description Ω_2 and the SMPC performance with $A_m = A_p = 0.4$, $B_m = B_p = 0.4$.

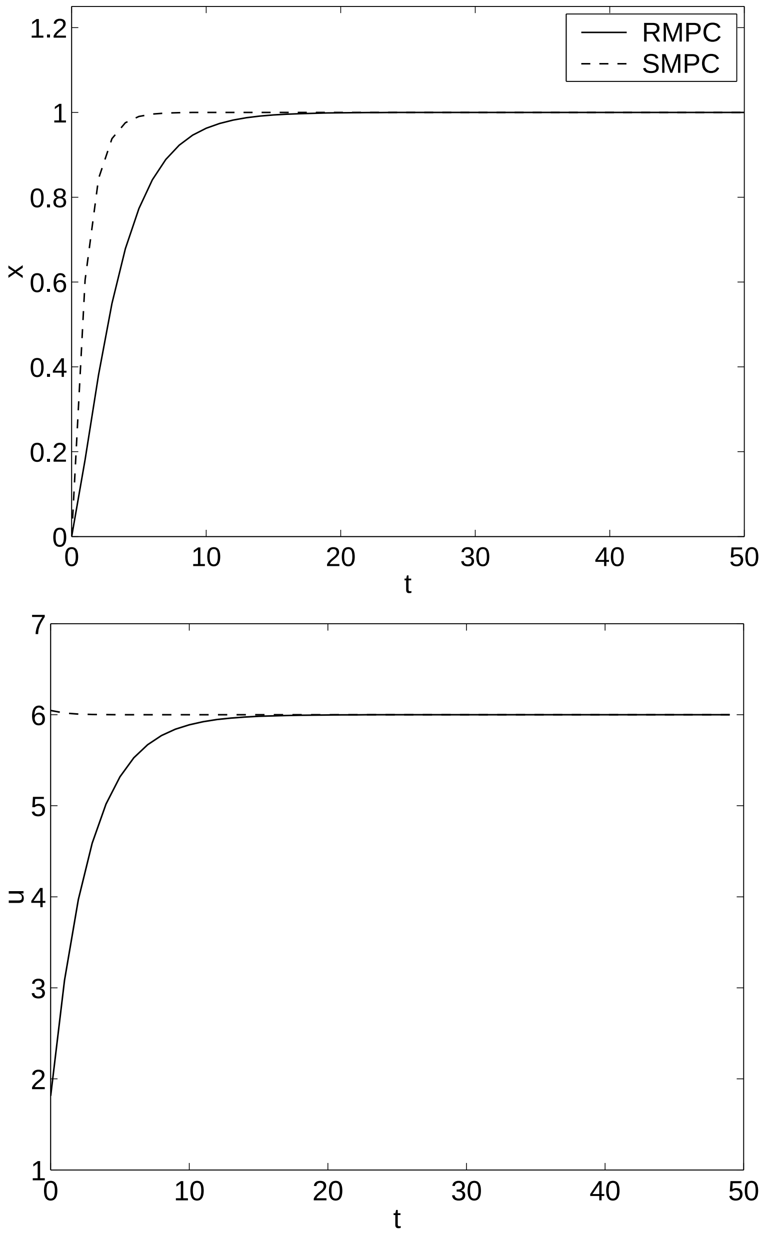

Figure 6.5: Comparison of the RMPC performance with uncertainty description Ω_2 and the SMPC performance with $A_m = A_p = 0.4$, $B_m = B_p = 0.1$.

the nominal controller performance. The best performance the RMPC controller can achieve is the nominal MPC performance without plant-model mismatch. Robustness is gained by decreasing the aggressiveness of the control actions

6.2.2 Open-loop unstable system

First, we compare the RMPC control performance to the SMPC control performance for an uncertainty description including both stable and unstable models. The example is designed to be a difficult test for the RMPC controller.

$$\text{Model 1:} \qquad y(t+1) = 0.5y(t) + u(t)$$

$$\text{Model 2:} \qquad y(t+1) = 1.6y(t) + u(t)$$

Using the tuning parameters $Q_s = 1$, $Q = C^T Q_s C$, $R = 1$, and $P_1 = P_2 = 1$, the robustly stabilizing control parameters are $K = -0.99$, $F = 5.007 \times 10^{10}$, $L_1 = 0.5109$, and $L_2 = 0.54957$. Figures 6.6 and 6.7 compare the RMPC control performance to the SMPC control performance with or without plant-model mismatch. The control objective is a set point change from 0 to -1. The uncertainty in the A matrix causes the SMPC controller using model 1 to be closed-loop unstable when the plant is model 2. On the other hand, the SMPC controller using model 2 is closed-loop stable when the plant is model 1, but its control performance is very oscillatory and aggressive compared to the nominal SMPC control performance. Even though the RMPC control performance is more oscillatory and not as efficient as the nominal SMPC control performance because of the

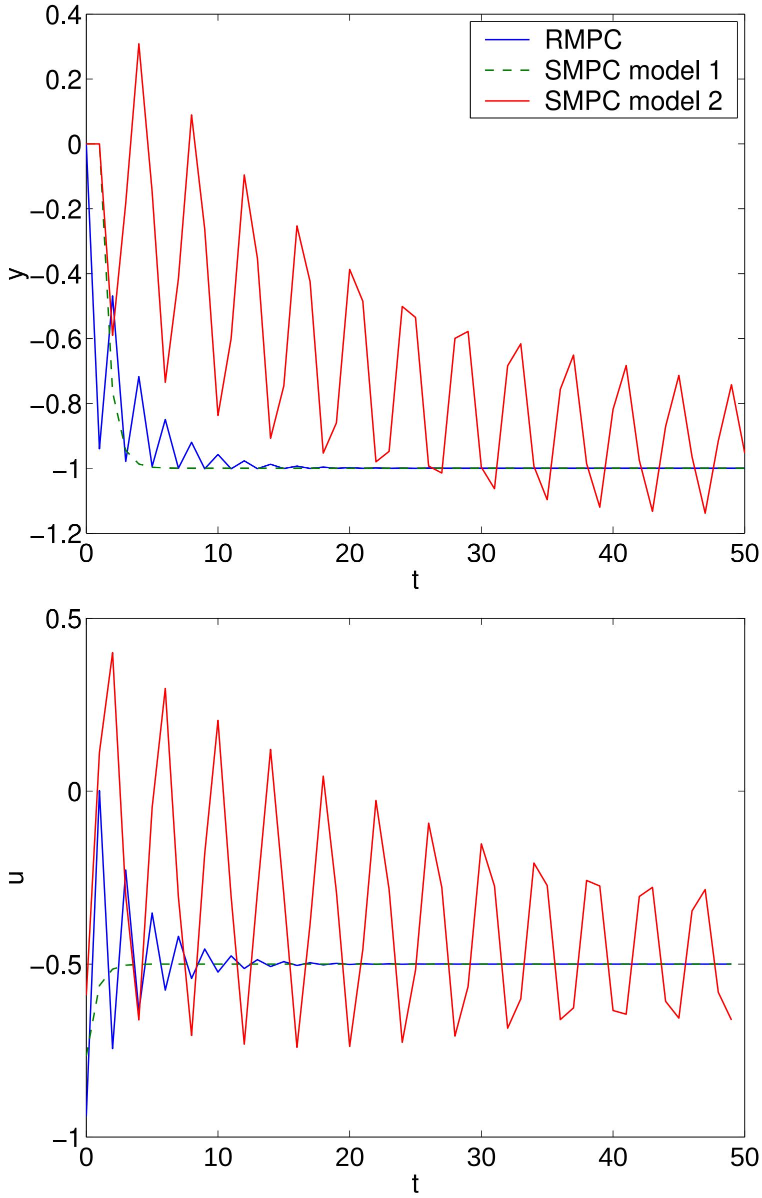

Figure 6.6: Closed-loop performance for the open-loop unstable system with model 1 as the plant.

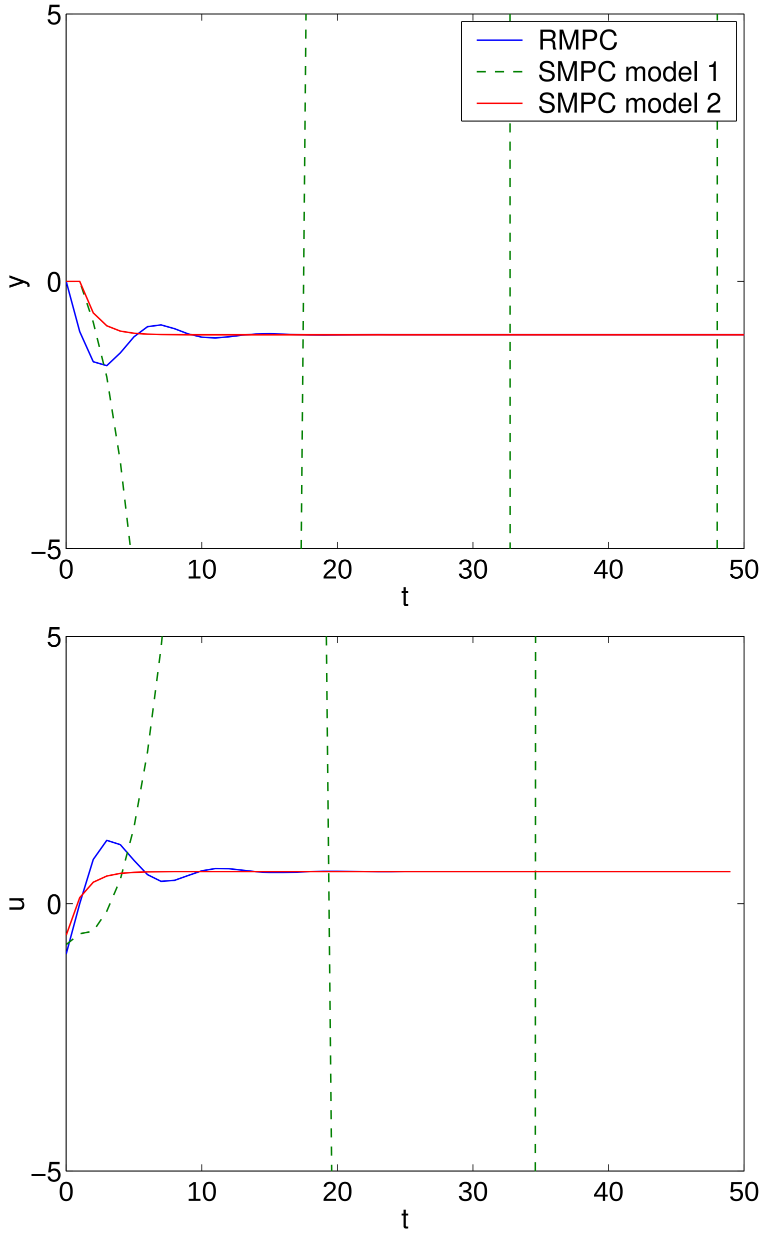

Figure 6.7: Closed-loop performance for the open-loop unstable system with model 2 as the plant.

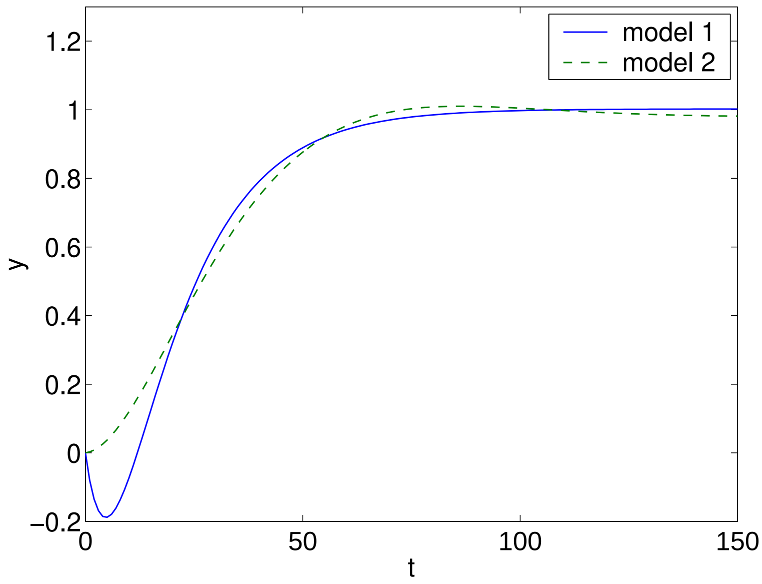

Figure 6.8: Open-loop response of the inverse response system

mismatch in A, the RMPC controller can successfully control model 1, model 2, and any time-varying linear combination of models 1 and 2, which include both open-loop stable and unstable models. But Figure 6.7 shows the SMPC controller is closed-loop unstable in the presence of this plant-model mismatch.

6.2.3 Inverse response

The following example has dynamic uncertainties described by the convex hull of two models. Figure 6.8 shows the open-loop response to a unit step change in the input. The ARMAX models are:

- Model 1:

$$\begin{bmatrix} y(t+1) \\ y(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} 1.79250 & -0.80145 & 0.08928 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} -0.08031 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t+1)$$

- Model 2:

$$\begin{bmatrix} y(t+1) \\ y(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} 1.91924 & -0.92208 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0.002786 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t+1).$$

Using the tuning parameters $Q_s = 1$, $Q = C^T Q_s C$, $R = 1$, and $P_1 = P_2 = I$, the SMPC control algorithm is closed-loop stable for both models 1 and 2. Figures 6.9 and 6.10 compare the closed-loop control performance between the SMPC and RMPC for the set point change from 0 to 1. Even though model 2 does not model the inverse response properly, the mismatch is not significant enough to cause closed-loop instability. When the plant is model 1 but the SMPC uses model 2, the inverse response in the plant causes the SMPC to take more aggressive, oscillatory control actions to properly control the plant for the set point change.

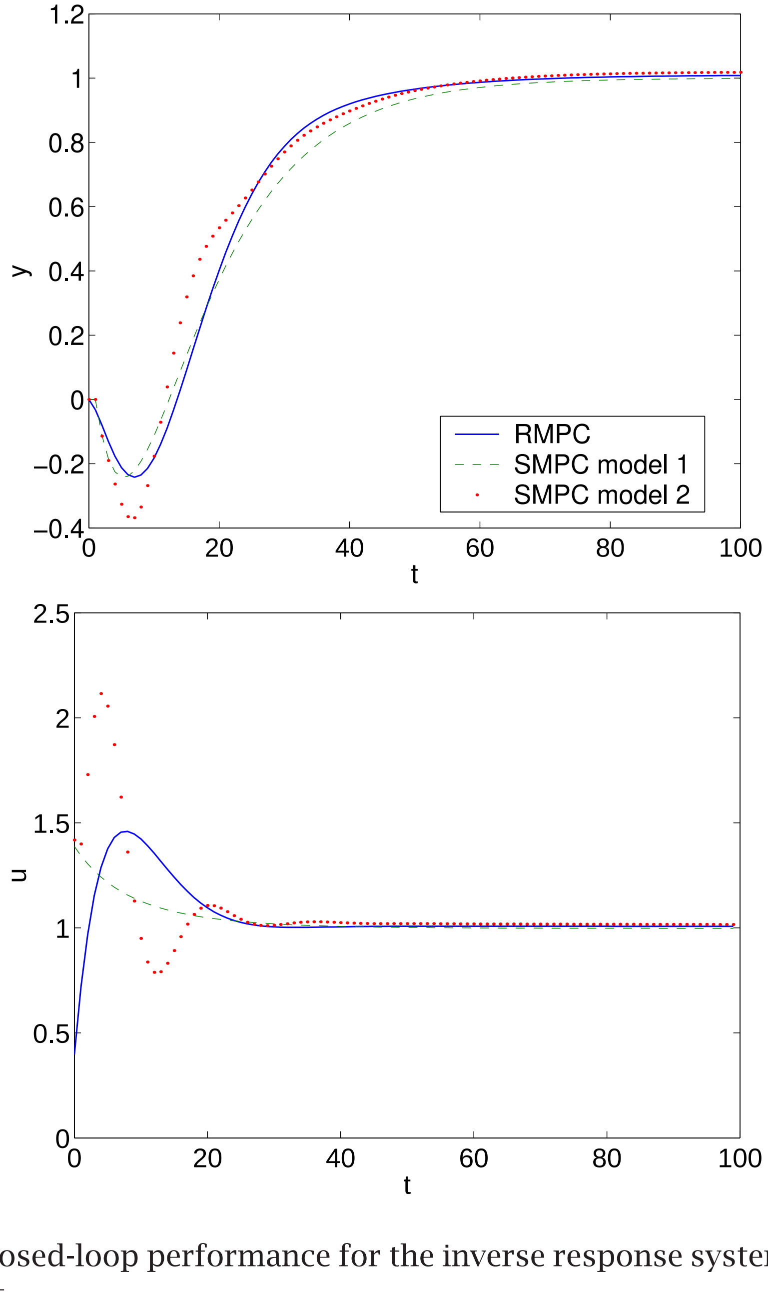

Figure 6.9: Closed-loop performance for the inverse response system with model 1 as the plant.

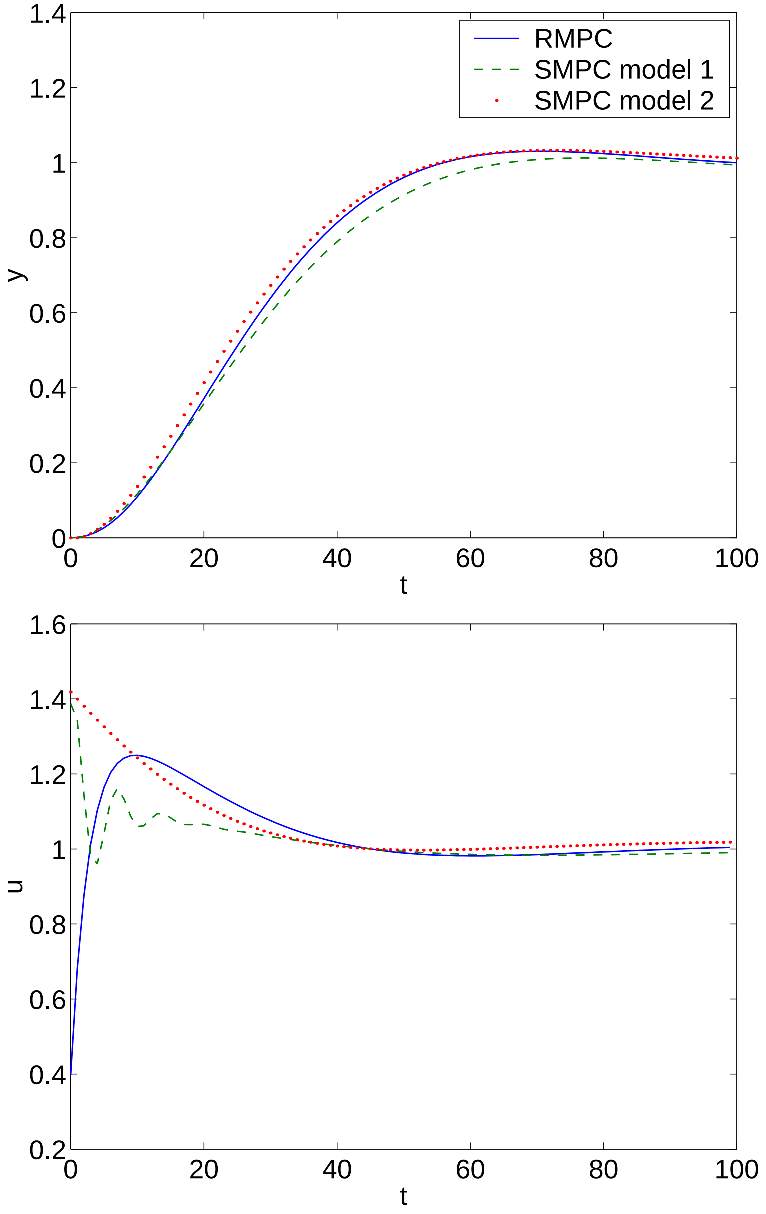

Figure 6.10: Closed-loop performance for the inverse response system with model 2 as the plant.

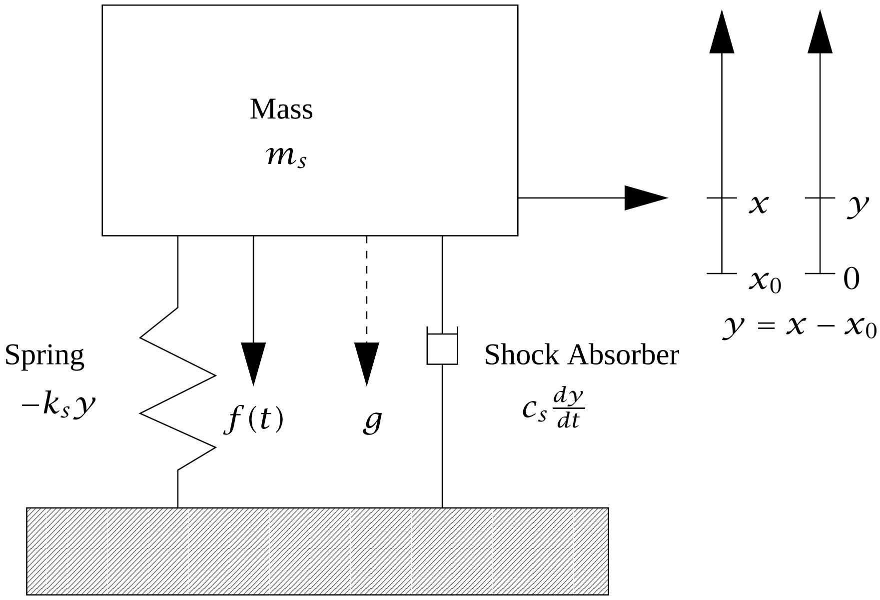

Figure 6.11: Suspension system exhibiting oscillatory open-loop response

6.2.4 Oscillatory system

Figure 6.11 shows a suspension system with a shock absorber that tends to wear out with use [112]. The parameters of the suspension system are:

- spring constant $k_s = 2000$ newtons/m,
- mass of the load $m_s = 500$ kg,
- amplitude of the test force $A_s = 50$ newtons.

The shock absorber constant c_s for various wear and tear levels are:

1. New heavy duty shocks: $c_s = 1600$ newtons/(m/s),
2. New regular shocks: $c_s = 1200$ newtons/(m/s),
3. Worn regular shocks: $c_s = 400$ newtons/(m/s).

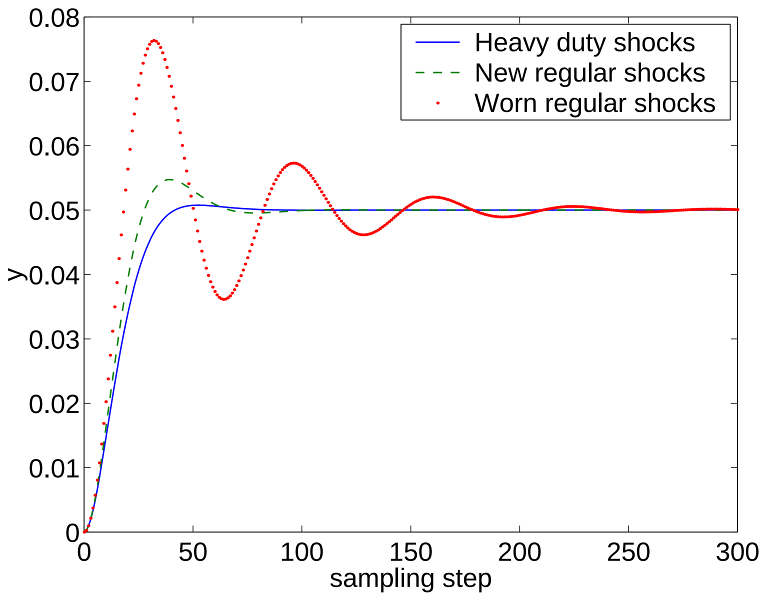

Figure 6.12: Open-loop response of the suspension system in figure 6.11 to a unit step change in the input

Equation 6.1 shows the dynamic relationship for the deviation from the desired position y as a function of k_s, m_s, A_s, and c_s.

$$m_s \frac{d^2y}{dt^2} = -k_s y - c_s \frac{dy}{dt} + f(t) \tag{6.1}$$

The continuous state-space model is

$$\begin{bmatrix} \frac{dy}{dt} \\ \frac{d^2y}{dt^2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m_s} & -\frac{c_s}{m_s} \end{bmatrix} \begin{bmatrix} y \\ \frac{dy}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_s} \end{bmatrix} u \tag{6.2}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ \frac{dy}{dt} \end{bmatrix}.$$

Figure 6.12 shows the open-loop response of the suspension system to a unit step change in the test force. The open-loop response for the heavy duty shocks is overdamped, and the response for the worn regular shocks is underdamped. The sampling time used to discretize the continuous state-space model (Equation 6.2) is one-tenth the time constant $\tau = \frac{m_s}{k_s}$. The B_i matrices are scaled up by a factor of 1000 to increase the accuracy of model identification. The discrete-time ARMAX models are:

- Model 1: Heavy duty shocks

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.85094 & -0.85953 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.0042927 \\ 0 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix}$$

- Model 2: Regular shocks

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.88163 & -0.89053 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.0044459 \\ 0 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix}$$

- Model 3: Worn regular shocks

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.95128 & -0.96087 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.0047938 \\ 0 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix}.$$

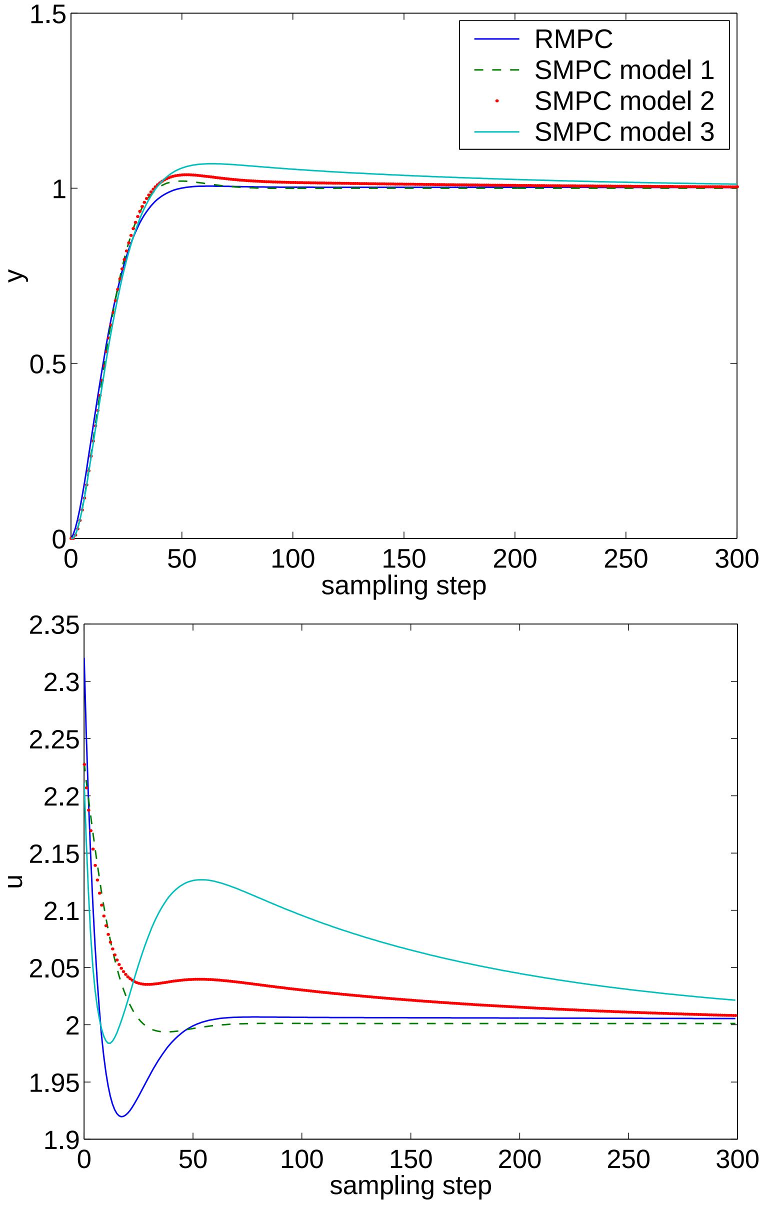

Figure 6.13: Closed-loop performance for the suspension system with model 1 as the plant.

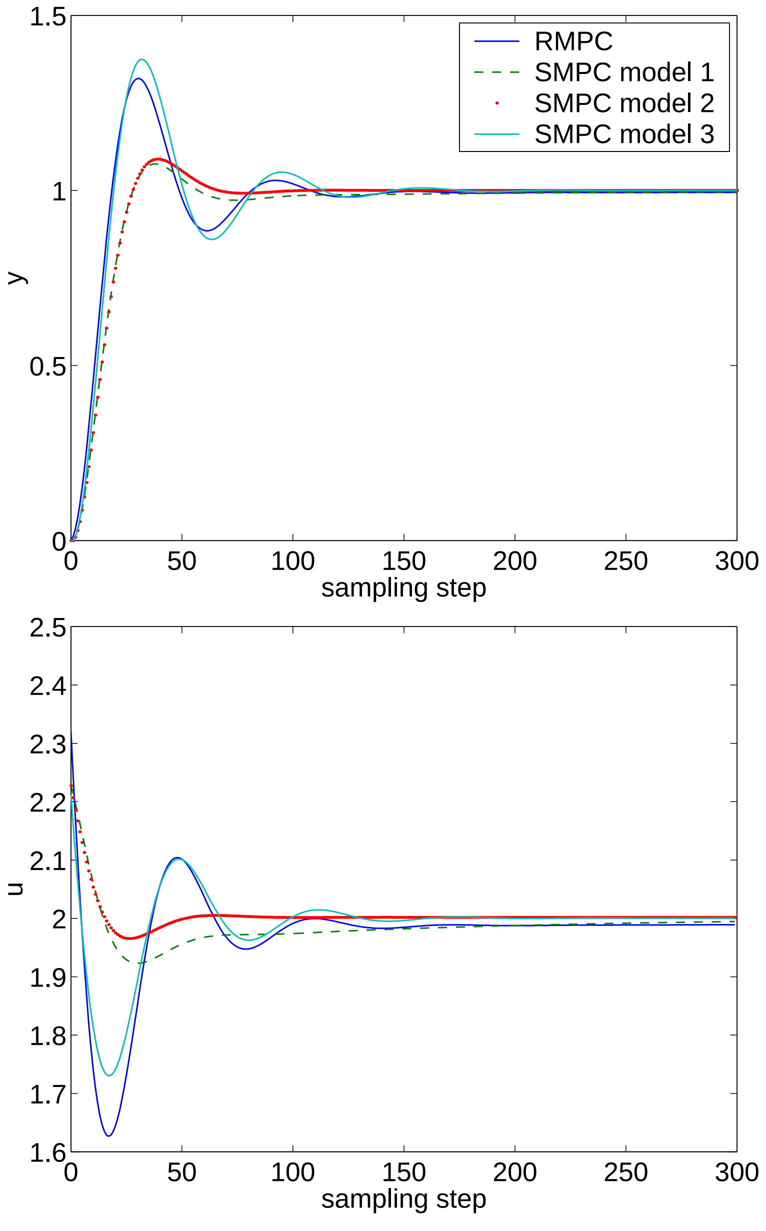

Figure 6.14: Closed-loop performance for the suspension system with model 2 as the plant.

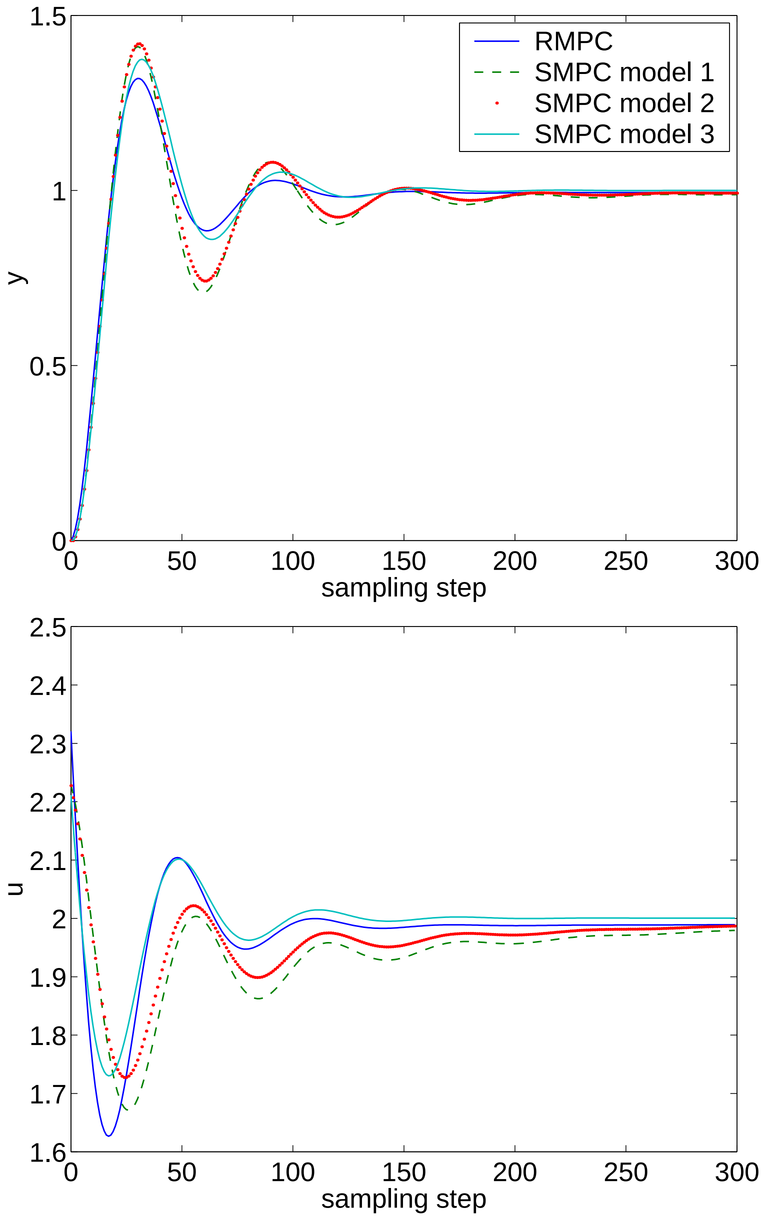

Figure 6.15: Closed-loop performance for the suspension system with model 3 as the plant.

The robustly stabilizing K, F, and L_i's are:

$$K = \begin{bmatrix} -6.6279 & 6.3084 \end{bmatrix}, \qquad F = \begin{bmatrix} 5.1409 & -4.8474 \\ -4.8474 & 4.6148 \end{bmatrix} \times 10^{13},$$

$$L_1 = \begin{bmatrix} 0.00072 & -1.08623 \\ 0.00059 & 0.99994 \end{bmatrix}, \qquad L_2 = \begin{bmatrix} 0.00077 & -1.12455 \\ 0.00059 & 1.00018 \end{bmatrix},$$

$$L_3 = \begin{bmatrix} 0.00365 & -1.20932 \\ -0.00310 & 0.99757 \end{bmatrix}$$

for the tuning parameters $Q_s = 1$, $Q = C^T Q_s C$, $R = 1$, and $P_1 = P_2 = P_3 = I$. Similar to the inverse response example, the dynamic uncertainties are not large enough to cause closed-loop instability for the SMPC algorithm. Figures 6.13-6.15 compare the closed-loop performance of SMPC to RMPC for a set point change from 0 to 1.

RMPC increases the robustness of the control algorithm to model uncertainties by decreasing the aggressiveness of the control actions. The RMPC performances in Figures 6.13-6.15 are similar to the SMPC performance without plant-model mismatch. For the suspension example, the RMPC control theory increased the robustness of the control algorithm by sacrificing little of the closed-loop control performance.

6.2.5 Underdamped system

The previous oscillatory system example examines the control performance of robust MPC for dynamic model uncertainties caused by uncertainties in the shock absorber constant c_s. The following example examines the SMPC and the RMPC control performance for uncertainties in the time constant $\tau = \sqrt{\frac{m_s}{k_s}}$. Using the worn regular shock absorber constant $c_s = 400$, the vertices of Ω are defined by three models with three different time constants.

- Model 1: $\tau = 0.5$

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.95128 & -0.96087 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.0047938 \\ 0 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix}$$

- Model 2: $\tau = 1.0$

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.88629 & -0.92314 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.018405 \\ 0 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix}$$

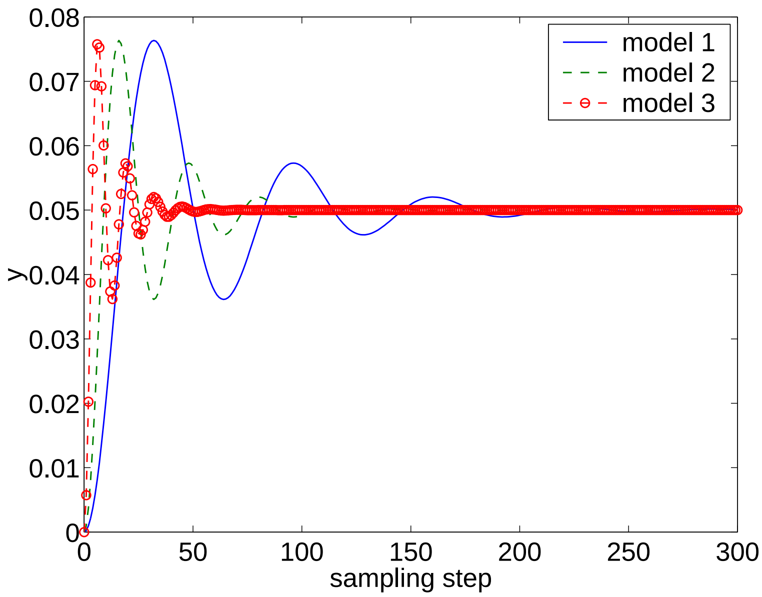

Figure 6.16: Open-loop response to a unit step change for the suspension system with $c_s = 400$.

- Model 3: $\tau = 2.5$

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.61377 & -0.81496 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.10046 \\ 0 \end{bmatrix} u(t)$$

$$y(t+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix}$$

Figure 6.16 shows the open-loop response for a unit step change in the input.

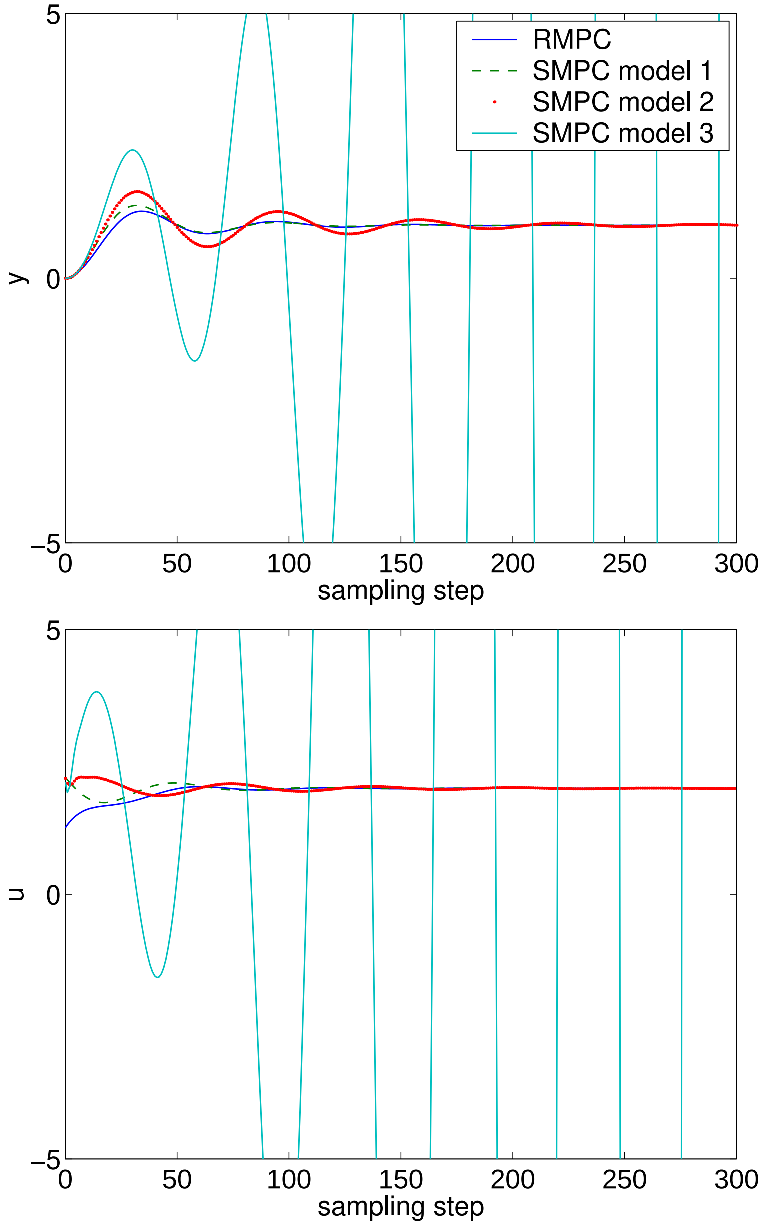

Figure 6.17: Closed-loop performance for the underdamped system with model 1 as the plant.

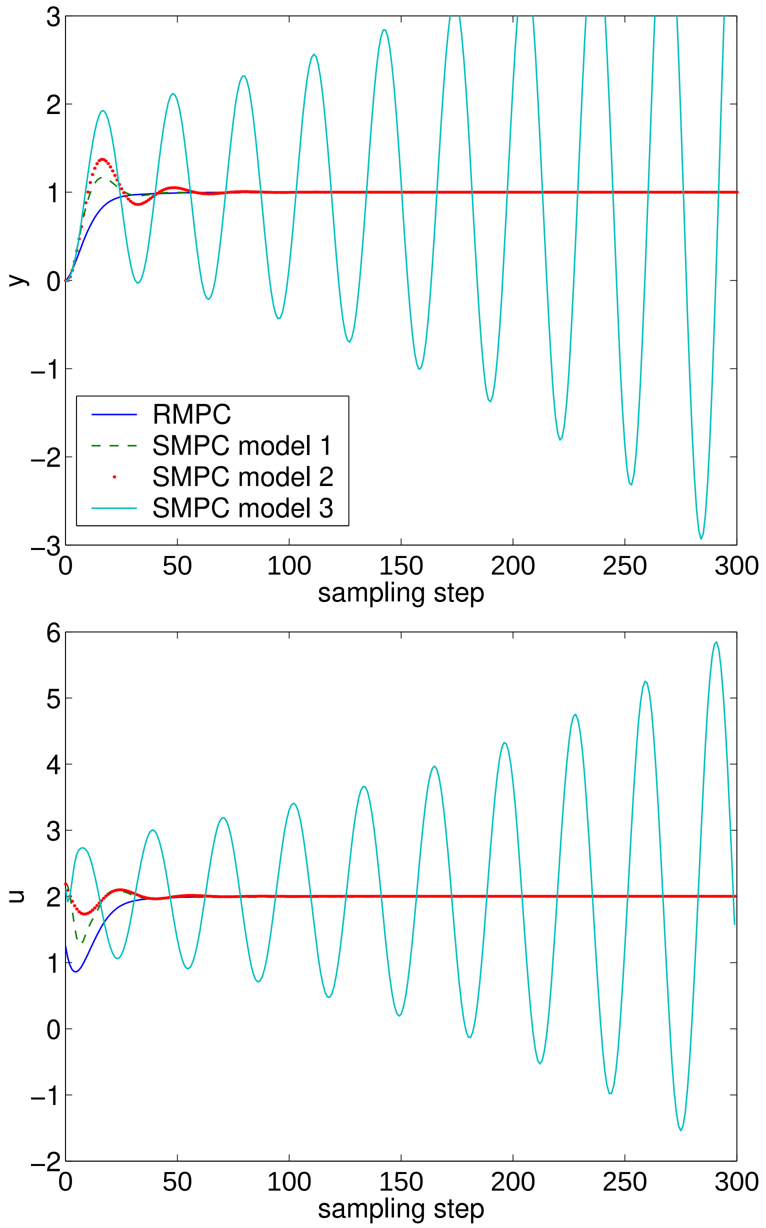

Figure 6.18: Closed-loop performance for the underdamped system with model 2 as the plant.

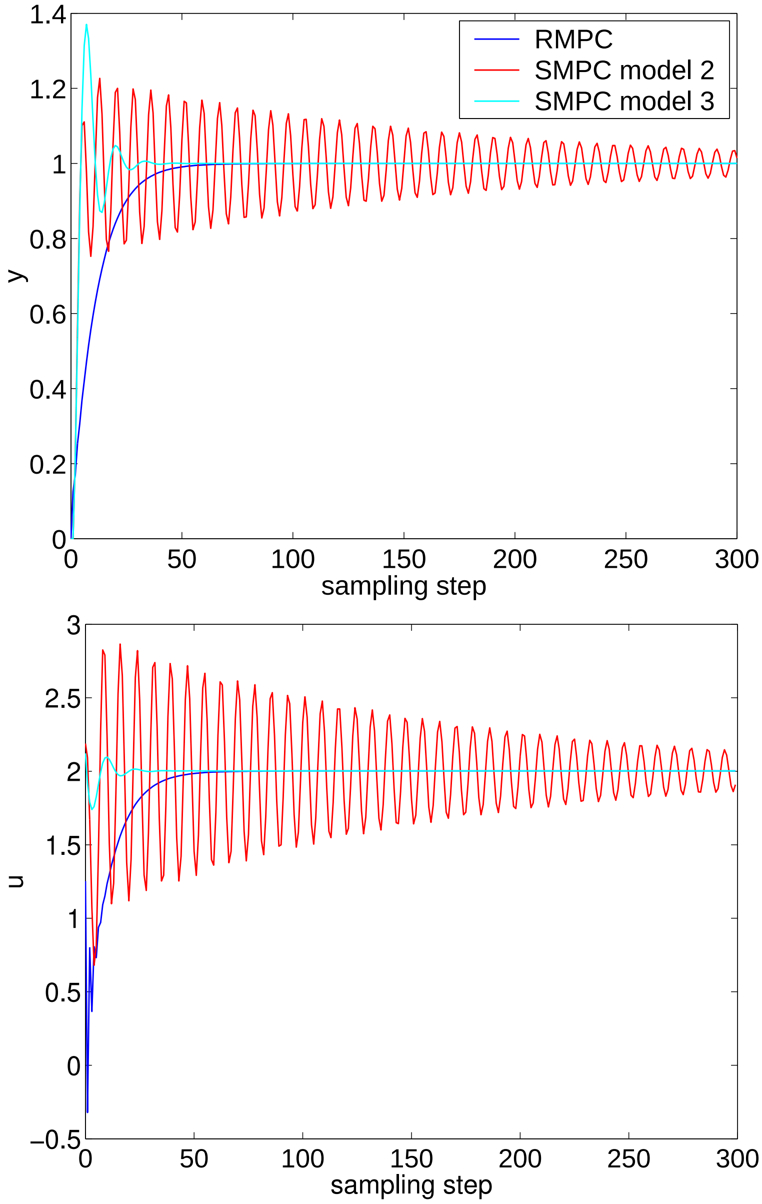

Figure 6.19: Closed-loop performance for the underdamped system with model 3 as the plant.

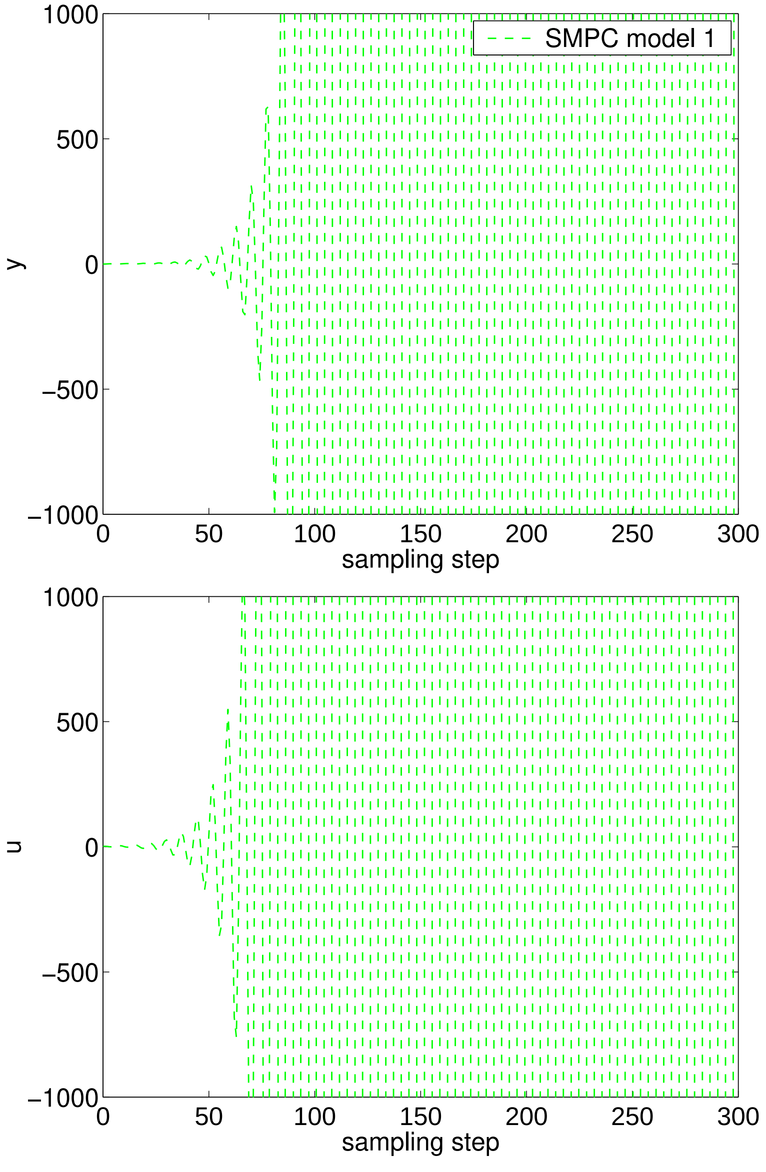

Figure 6.20: Closed-loop performance for the underdamped system with model 3 as the plant.

The robustly stabilizing K, F, and L_i's are:

$$K = \begin{bmatrix} -12.0595 & 12.8092 \end{bmatrix}, \qquad F = \begin{bmatrix} 2.6337 & -2.4266 \\ -2.4266 & 2.2608 \end{bmatrix} \times 10^{14},$$

$$L_1 = \begin{bmatrix} 0.05470 & -0.04937 \\ -0.00371 & -0.03896 \end{bmatrix}, \qquad L_2 = \begin{bmatrix} 0.21331 & -0.04152 \\ 0.00347 & 0.42407 \end{bmatrix},$$

$$L_3 = \begin{bmatrix} 0.10910 & 0.29405 \\ 0.00432 & -0.06387 \end{bmatrix}$$

for the tuning parameters $Q_s = 1$, $Q = C^T Q_s C$, $R = 1$, and $P_1 = P_2 = P_3 = I$. Figures 6.17-6.20 compare the SMPC control performance to the RMPC control performance for the vertices of Ω. Figure 6.17 shows that SMPC using model 3 is closed-loop unstable. SMPC using model 1 or 2 is closed-loop stable, but the closed-loop performance is slightly more oscillatory than the closed-loop performance of RMPC. When the plant is equal to model 2, SMPC using model 3 is still closed-loop unstable. The RMPC control performance is not only closed-loop stable but much less oscillatory than either SMPC with model 1 or SMPC with model 2, the nominal MPC performance (see Figure 6.18). Figures 6.19 and 6.20 show that if model 3 is the plant, SMPC using model 1 is closed-loop unstable while SMPC using models 2 and 3 are closed-loop stable. For the plant equal to the vertices of Ω, the RMPC control performance is similar to the nominal MPC performance. The RMPC method increased the robustness of the controller to modeling errors and sacrifices little in the controller performance.

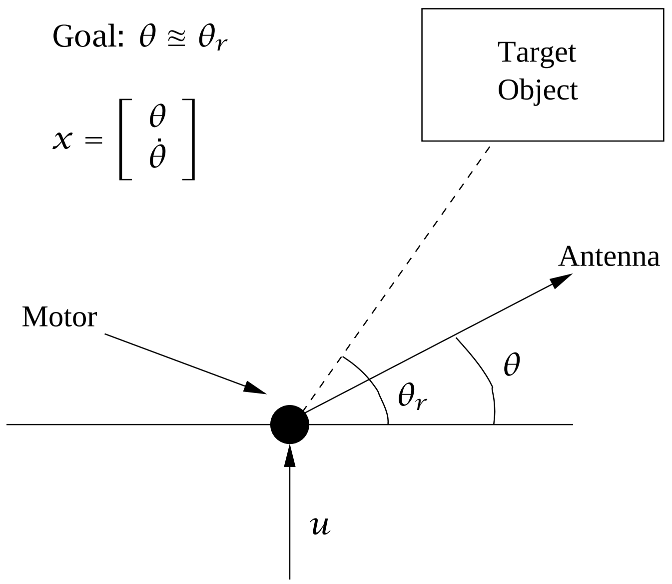

Figure 6.21: Angular positioning system

6.2.6 Angular positioning system

Figure 6.21 shows the angular positioning system example adapted from Kwakernaak and Sivan [75]. The rotating antenna at the origin of the plane is driven by an electric motor. The objective of the control problem "is to use the input voltage to the motor to rotate the antenna so that it always points in the direction of a moving object in the plane" [69]. Kothare et al. [69] assumes both the angular positions of the antenna and of the moving object (θ and θ_r rad respectively) and the angular velocity of the antenna ($\dot{\theta}$ rad s^{-1}) are measurable. The angular velocity is measurable assumption is not physically practical. The models are changed from the state-space models with

$$x(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}$$

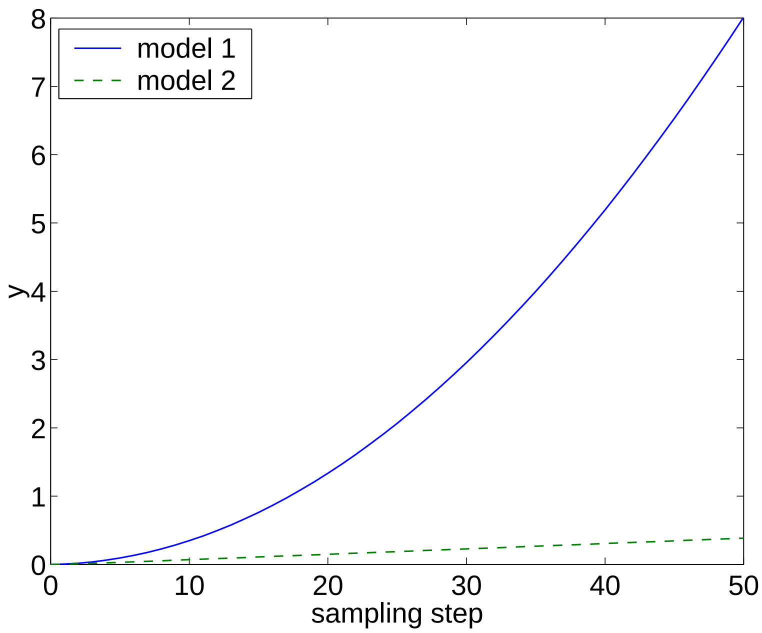

Figure 6.22: Open-loop response for the angular positioning system (Figure 6.21).

to the ARMAX model with $\theta(t)$ as the controlled output. The uncertain model description given by Kothare et al. [69] is

$$\begin{bmatrix} \theta(t+1) \\ \dot{\theta}(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1-0.1\alpha(t) \end{bmatrix} \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1\kappa \end{bmatrix} u(t)$$

in which $\kappa = 0.787$ rad^{-1} V^{-1} s^{-2} and $\alpha(t)$ is proportional to the coefficient of viscous friction in the rotating parts of the antenna and may be arbitrarily time-varying in the range $0.1\text{s}^{-1} \leq \alpha(t) \leq 10\text{s}^{-1}$. Using the minimum and maximum values of $\alpha(t)$, Ω is formed by the convex hull of

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

There is no uncertainty in B. The ARMAX models corresponding to A_1 and A_2 are:

- Model 1: $\alpha(t) = 0.1$

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 2.00253 & -1.00275 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.0062956 \\ 0 \end{bmatrix} u(t)$$

- Model 2: $\alpha(t) = 10$

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} 1.51068 & -0.51070 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix} + \begin{bmatrix} 0.0038572 \\ 0 \end{bmatrix} u(t)$$

in which $y(t)$ is the angular position $\theta(t)$. Figure 6.22 shows that model 1 is open-loop unstable for a unit step change in the input. The robustly stabilizing K, F, and L_i's for the tuning parameters $Q_s = 1$, $Q = C^T Q_s C$, $R = 1$, and $P_1 = P_2 = I$ are:

$$K = \begin{bmatrix} -39.5137 \\ 36.8505 \end{bmatrix}, \qquad F = \begin{bmatrix} 4.2312 & -3.9188 \\ -3.9188 & 3.7047 \end{bmatrix} \times 10^{14},$$

$$L_1 = \begin{bmatrix} -0.04562 & -1.70393 \\ 0.22219 & 1.04375 \end{bmatrix}, \qquad L_2 = \begin{bmatrix} 0.16822 & -0.77189 \\ -0.05363 & 0.53643 \end{bmatrix}.$$

When the plant is equal to model 1, SMPC using model 2 is closed-loop unstable (Figure 6.23). The RMPC performance is very similar to the SMPC performance without plant-model mismatch. The additional robustness of the control algorithm is accomplished by sacrificing little of the controller performance. Figure 6.24 shows the RMPC performance achieves the new set point of 1 much faster than the SMPC algorithm either with or without plant-model mismatch.

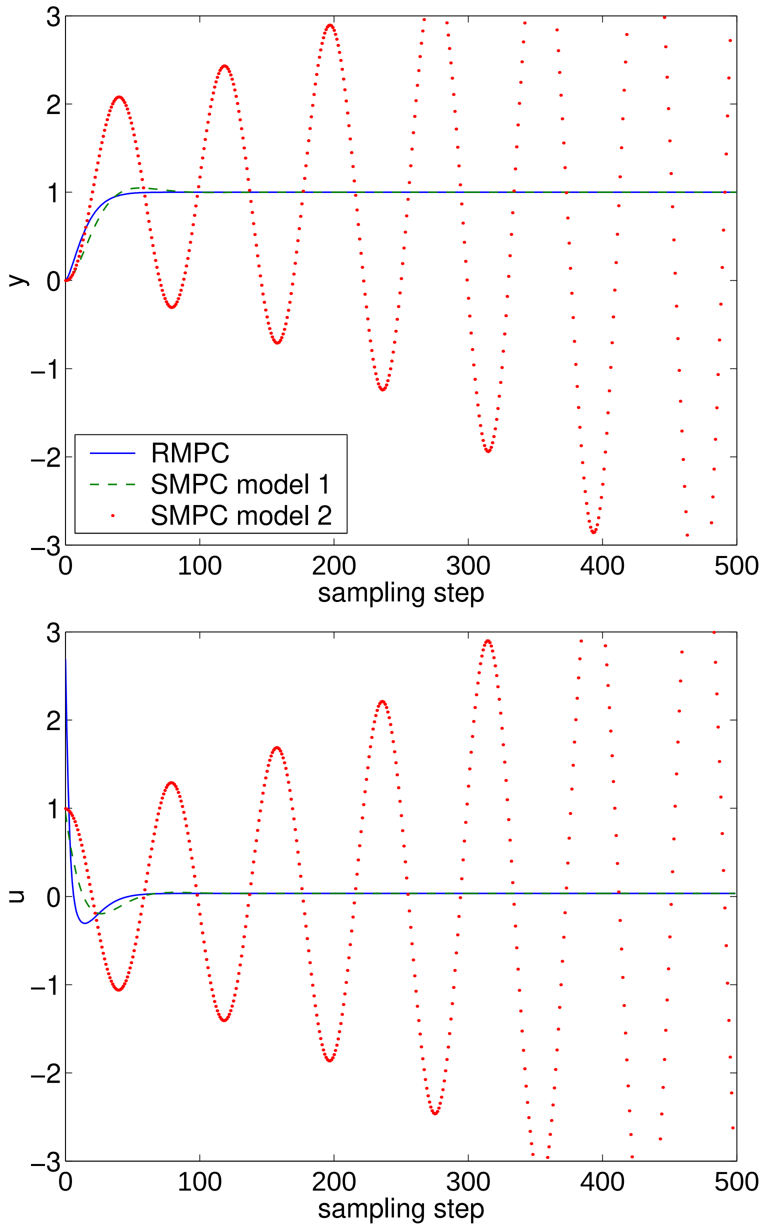

Figure 6.23: Closed-loop performance for the angular positioning system with model 1 as the plant.

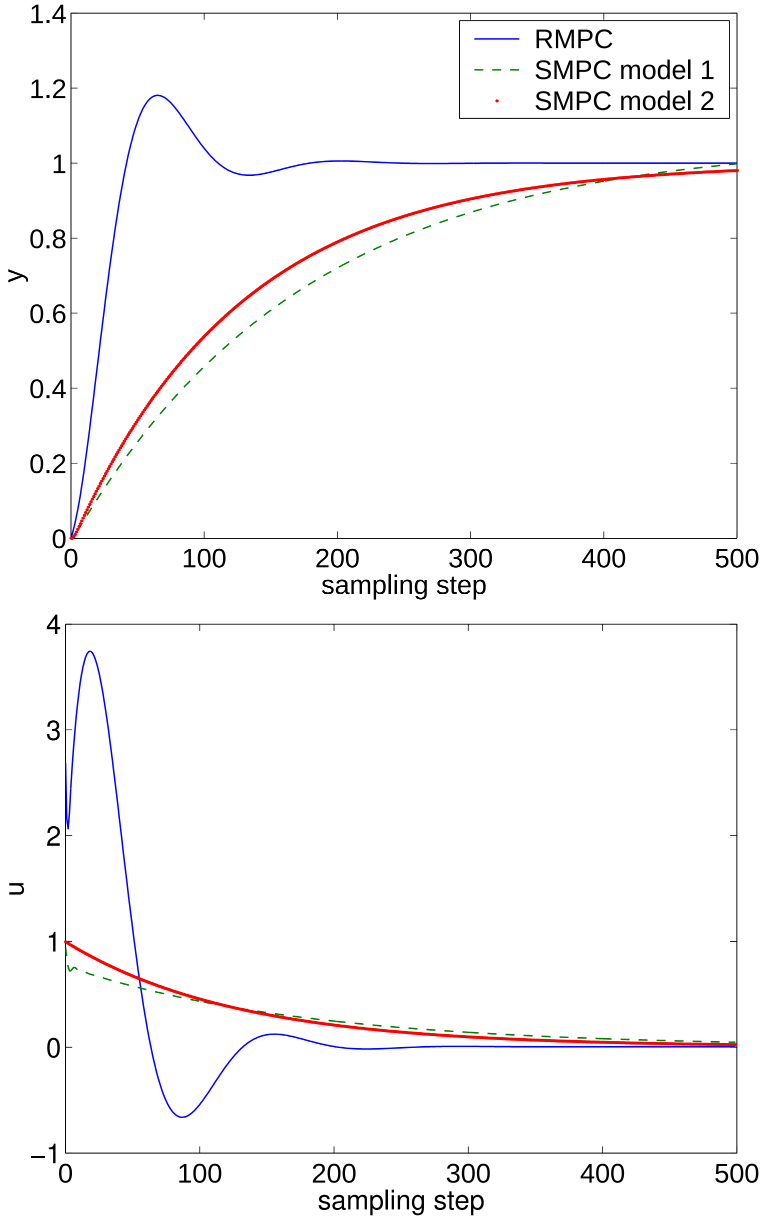

Figure 6.24: Closed-loop performance for the angular positioning system with model 2 as the plant.

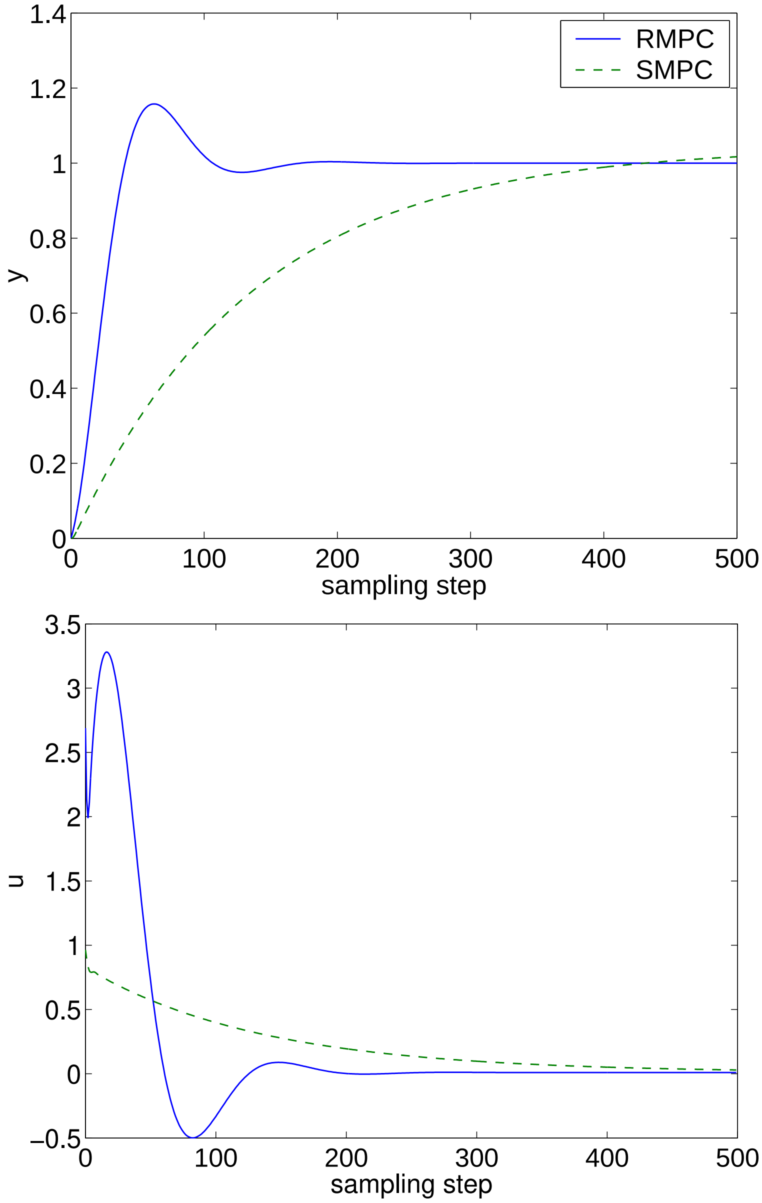

Figure 6.25: Closed-loop performance for the angular positioning system with $\mu_1 = 0.101$ and $\mu_2 = 0.899$ for the plant while the SMPC model is $\mu_1 = 0.9091$ and $\mu_2 = 0.0909$.

Figure 6.25 compares the RMPC and the SMPC control performance for a plant that is contained inside the convex hull Ω. The SMPC uses a model that is also inside Ω and not one of the vertices. Even though both the RMPC and the SMPC are closed-loop stable, RMPC takes less time than the SMPC to reach the new set point. All of the above simulations are performed for a set point change from 0 to 1.

6.2.7 Constraint saturation example

An example with constraint saturation is presented to show that the RMPC controller successfully rejects disturbances without wind-up. Figure 6.26 compares the closed-loop performance of the RMPC to the performance shown in Figure 4.3 using the ARMAX models with integrator $\xi(t)$ and the same process models and operating conditions. The same step disturbance enters the system at time $t = 50$, which moves the set point y_t into the feasible region. As soon as the disturbance enters the plant, RMPC takes control actions immediately to bring the state to its set point. There is no delay as shown by the performance using the ARMAX models with the integrator $\xi(t)$. The wind-up is removed because the external disturbance affects the values of $p_i(t)$ as soon as it enters the system. Once the disturbance model estimates $p_i(t)$ change, the u_s, x_s, and $p_{i,s}$ values change because of the target calculation. The change in u_s and x_s cause the RMPC controller to take control actions immediately and realize the set point

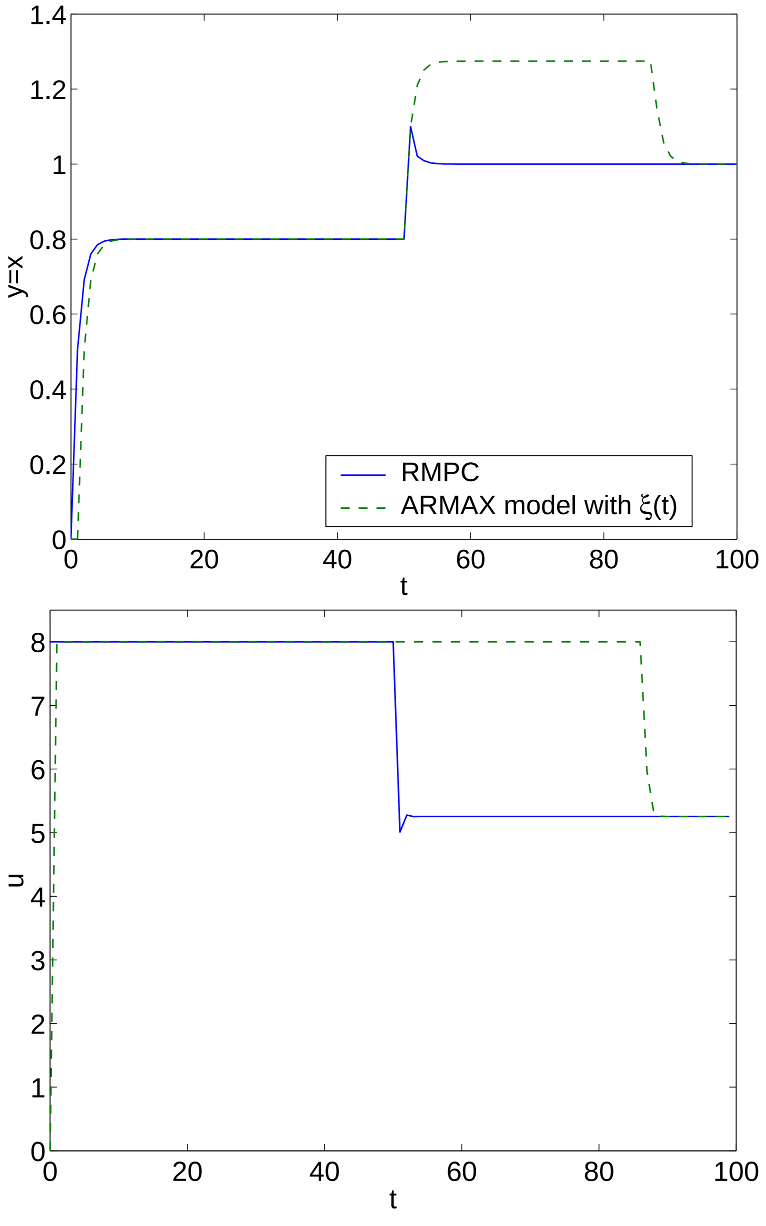

Figure 6.26: Comparison of closed-loop performance in the presence of constraint saturation for the uncertain models in equation 4.14.

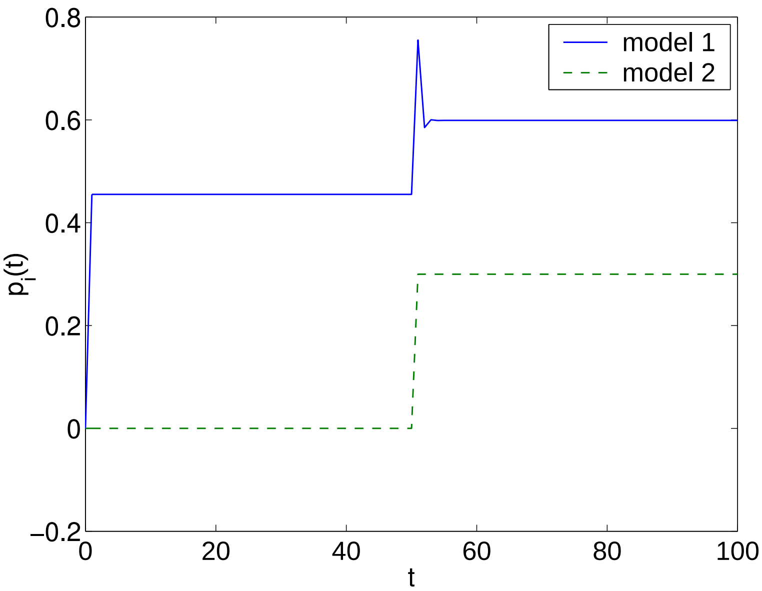

Figure 6.27: Graphical illustration of disturbance updating using the uncertain models in equation 4.14.

is now reachable. Astrom and Rundqwist [1], Edwards and Postlethwaite [45], Kapasouris and Athans [63], Teel and Kapoor [149], Mulder et al. [105, 106], and Kothare et al. [70] provide a more general discussion of wind-up behavior and methods to remove wind-up. The RMPC method used in this simulation is just one of the many control methods that avoid wind-up behavior in the presence of constraint saturation.

Figure 6.27 shows the closed-loop disturbance responses using robust MPC. From time $t = 0$ to $t = 50$, the disturbance for model 2 is zero because it is the correct model, and the disturbance for model 1 correctly biases model 1 state predictions so that they are equal to the estimated states. After the exogenous step disturbance enters the plant, both $p_1(t)$ and $p_2(t)$ estimate the

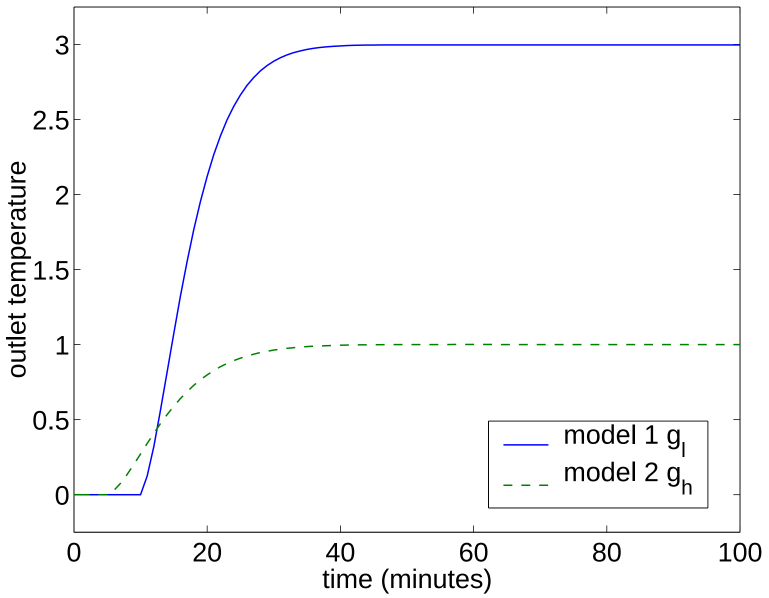

Figure 6.28: Open-loop response to a unit step change in the fuel gas feed rate for the fired heater example.

disturbance correctly.

6.2.8 Fired heater example

Besides uncertainties in the time constant and the steady-state gain matrices, model uncertainties may also occur in the time delay. A fired heater problem is presented that has uncertainties in the time constant, the steady-state gain, and the time delay [124]. The process is a single-input-single-output system with the fuel gas feed rate as the manipulated input and the heater outlet temperature as the controlled output. The dynamic response of the heater changes as the fuel gas feed rate changes. The system's open-loop dynamics is modeled by the transfer function g_l at low fuel gas feed rates and g_h at high fuel gas feed rates, respectively.

$$g_l = \frac{3e^{-10s}}{(4s+1)^2}$$
$$g_h = \frac{1e^{-5s}}{(5s+1)^2}$$

in which the time constants are in minutes. Figure 6.28 shows the open-loop response of the fired heater system to a unit step change in the fuel gas feed rate.

If there are no time-delays, the g_l and g_h transfer functions can be accurately transformed into the following ARMAX models.

$$\text{Model 1, } g_l: \ y(t+1) = a_1^l y(t) + a_2^l y(t-1) + b_1^l u(t)$$

$$\text{Model 2, } g_h: \ y(t+1) = a_1^h y(t) + a_2^h y(t-1) + b_1^h u(t)$$

Similar to the previous examples, the state is formed by the current $y(t)$ and the previous $y(t-1)$. With the time delays, the state is augmented to include all the past inputs that have been injected into the system but whose effects have not been seen due to the time delay. The sampling time used to get the discrete-time models is 1 minute.

- Model 1

$$
\begin{bmatrix} y(t+1) \\ y(t) \\ u(t) \\ u(t-1) \\ u(t-2) \\ u(t-3) \\ u(t-4) \\ u(t-5) \\ u(t-6) \\ u(t-7) \\ u(t-8) \\ u(t-9) \end{bmatrix}
= \begin{bmatrix}
a_1^l & a_2^l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_1^l \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} x(t) +
\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{6.3}
$$

$$
y(t+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t+1)
$$

- Model 2

$$
\begin{bmatrix} y(t+1) \\ y(t) \\ u(t) \\ u(t-1) \\ u(t-2) \\ u(t-3) \\ u(t-4) \\ u(t-5) \\ u(t-6) \\ u(t-7) \\ u(t-8) \\ u(t-9) \end{bmatrix}
= \begin{bmatrix}
a_1^h & a_2^h & 0 & 0 & 0 & 0 & b_1^h & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} x(t) +
\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{6.4}
$$

$$
y(t+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t+1)
$$

Model 2 has the same number of states as model 1 even though the time delay can be accurately modeled by a smaller number of states because the definition of the convex hull specifies that all models must have the same number of states. Algorithm 5.9.2 was successful at finding the robustly stabilizing K, F, L_1, and L_2, but it converged to the optimal solution very slowly.

Rao et al. [126] showed that for systems with time delays, the regulator is equivalent to systems without time delay by shifting the state forward by the amount of the time delay.

$$x(t) = x(t+\delta)$$

in which δ denotes the time delay. The same concept is applied to the fired heater's ARMAX models. Because the models must have the same number of states, the state is shifted forward by the time delay of model 2, $\delta_2 = 5$, which is less than the time delay of model 1, $\delta_1 = 10$.

- Model 1

$$\begin{aligned}
\begin{bmatrix} y(t+\delta_2+1) \\ y(t+\delta_2) \\ u(t+\delta_2) \\ u(t+\delta_2-1) \\ u(t+\delta_2-2) \\ u(t+\delta_2-3) \\ u(t+\delta_2-4) \end{bmatrix} &= \begin{bmatrix} a_1^l & a_2^l & 0 & 0 & 0 & 0 & b_1^l \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t+\delta_2) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \qquad (6.5) \\
y(t+\delta_2+1) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t+\delta_2+1)
\end{aligned}$$

- Model 2

$$\begin{aligned}
\begin{bmatrix} y(t+\delta_2+1) \\ y(t+\delta_2) \\ u(t+\delta_2) \\ u(t+\delta_2-1) \\ u(t+\delta_2-2) \\ u(t+\delta_2-3) \\ u(t+\delta_2-4) \end{bmatrix} &= \begin{bmatrix} a_1^h & a_2^h & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t+\delta_2) + \begin{bmatrix} b_1^h \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \qquad (6.6) \\
y(t+\delta_2+1) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t+\delta_2+1)
\end{aligned}$$

The robustly stabilizing K, F, L_1, and L_2 were found for the tuning parameters $Q_s = 1$, $Q = C^T Q_s C$, $R = 1$, and $P_1 = P_2 = I$. The objective is to control the

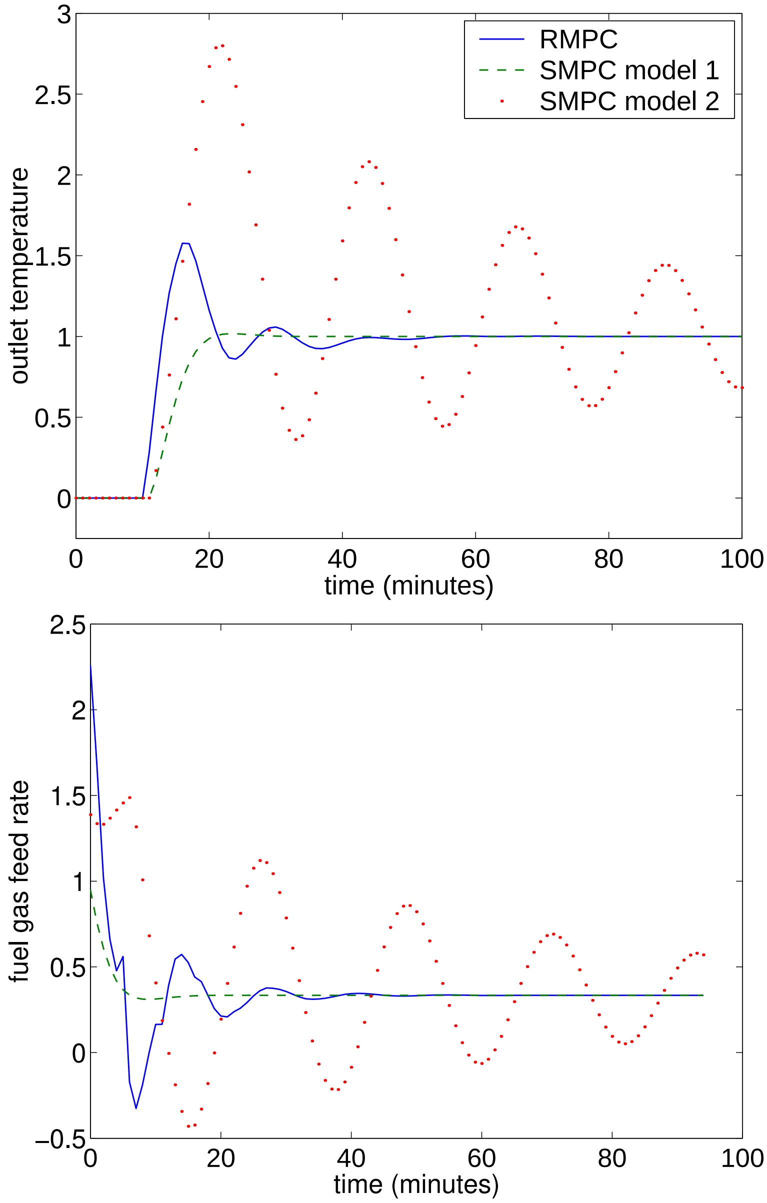

Figure 6.29: Closed-loop performance for the fired heater system with model 1 as the plant.

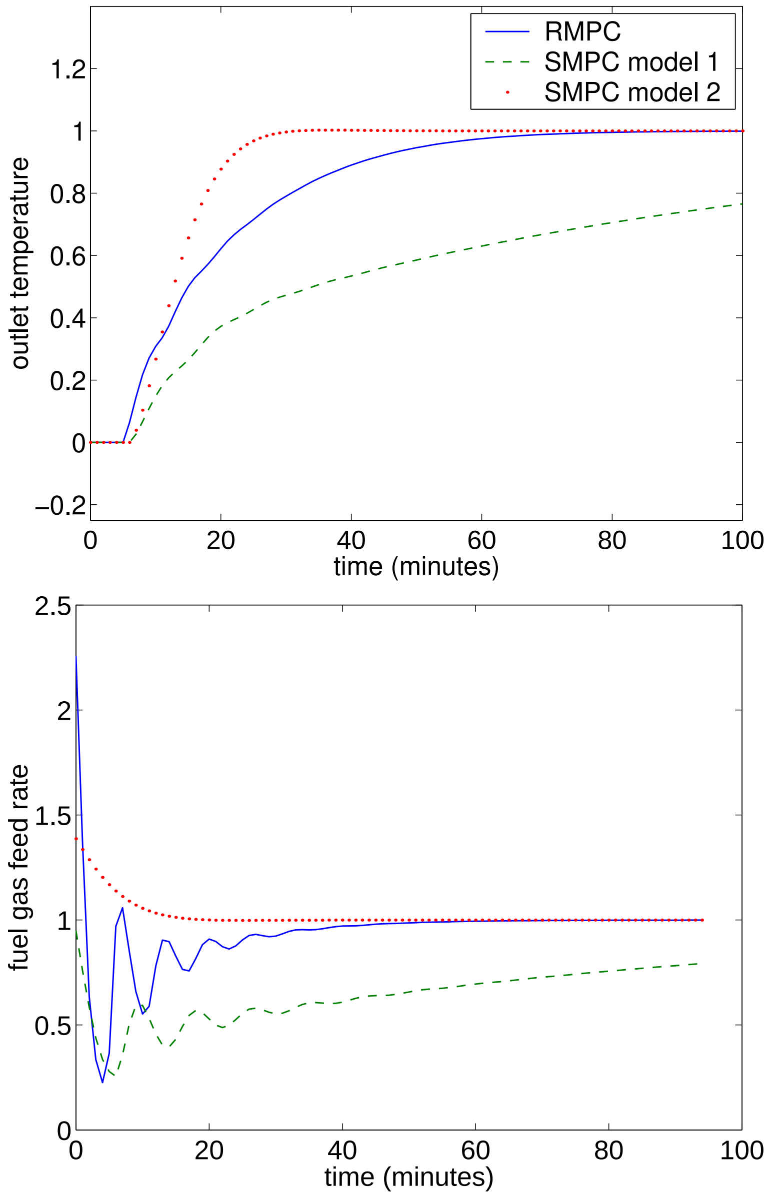

Figure 6.30: Closed-loop performance for the fired heater system with model 2 as the plant.

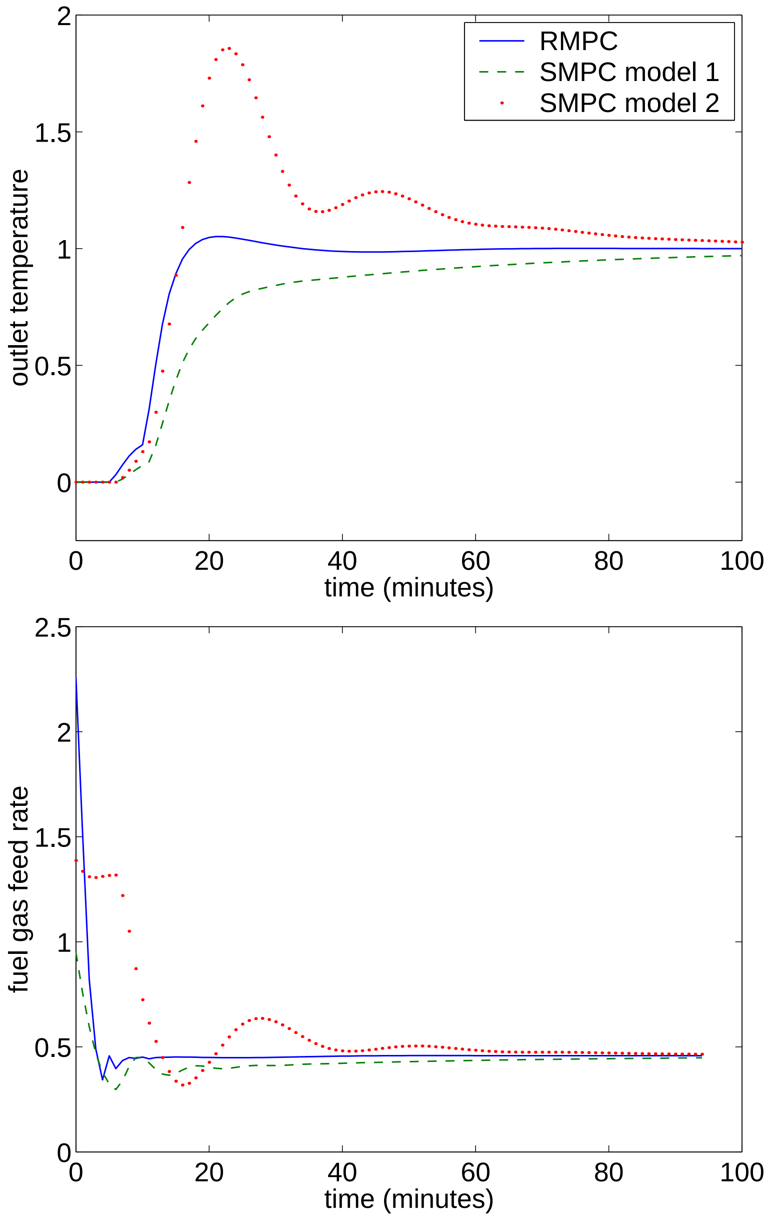

Figure 6.31: Closed-loop performance for the fired heater system with the plant equal to the average of models 1 and 2.

system for a set point change from 0 to 1. Figures 6.29-6.31 show that the SMPC is closed-loop stable in the presence of plant-model mismatch, but its performance is more sluggish than the RMPC performance for all the tested scenarios.

Figures 6.32 and 6.33 compare the closed-loop control performances of RMPC and SMPC with plant-model mismatch. The control objective is to reject a non-zero disturbance that enters the system at the second sampling step. Figure 6.32 shows that the RMPC performance is less oscillatory than the SMPC performance with plant-model mismatch because the plant has a higher gain and longer time delay than the SMPC model. Figure 6.29 shows similar cycling effect due to uncertainties in the gain and the time delay for a set point change. The SMPC controller performance cannot be improved with tunning. The uncertainty in time delay causes the cycling effect. Figure 6.33 shows that the longer time delay and lower gain in the SMPC model causes its performance to be more sluggish than the RMPC performance.

Figure 6.30 shows the SMPC control performance using model 1 is very sluggish compared to the RMPC control performance. The sluggish control performance is caused by the difference in the time delay between models 1 and 2. The SMPC algorithm uses the model with the longer time delay when the plant is actually responding to changes in the fuel gas feed rate much faster. SMPC achieves integral control by using the output disturbance model. Shinskey [143] showed that the output disturbance model is not very effective for systems with

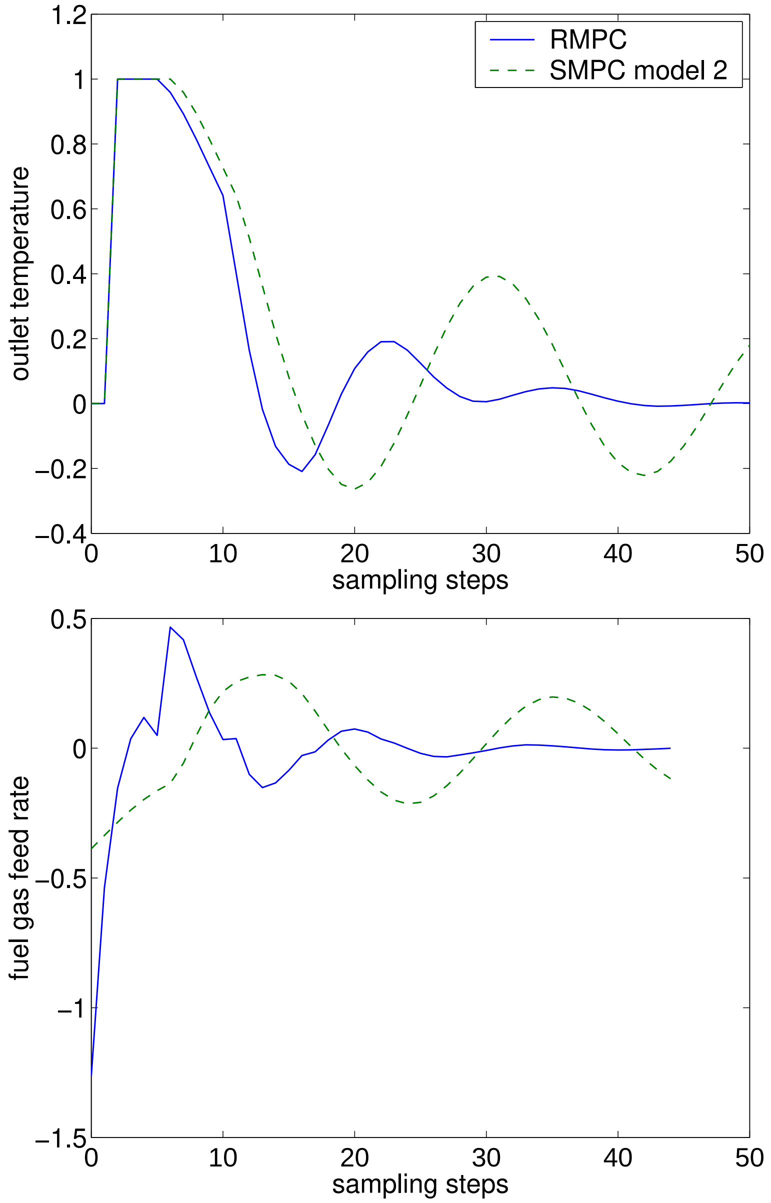

Figure 6.32: Comparison of disturbance rejection for the fired heater system with model 1 as the plant.

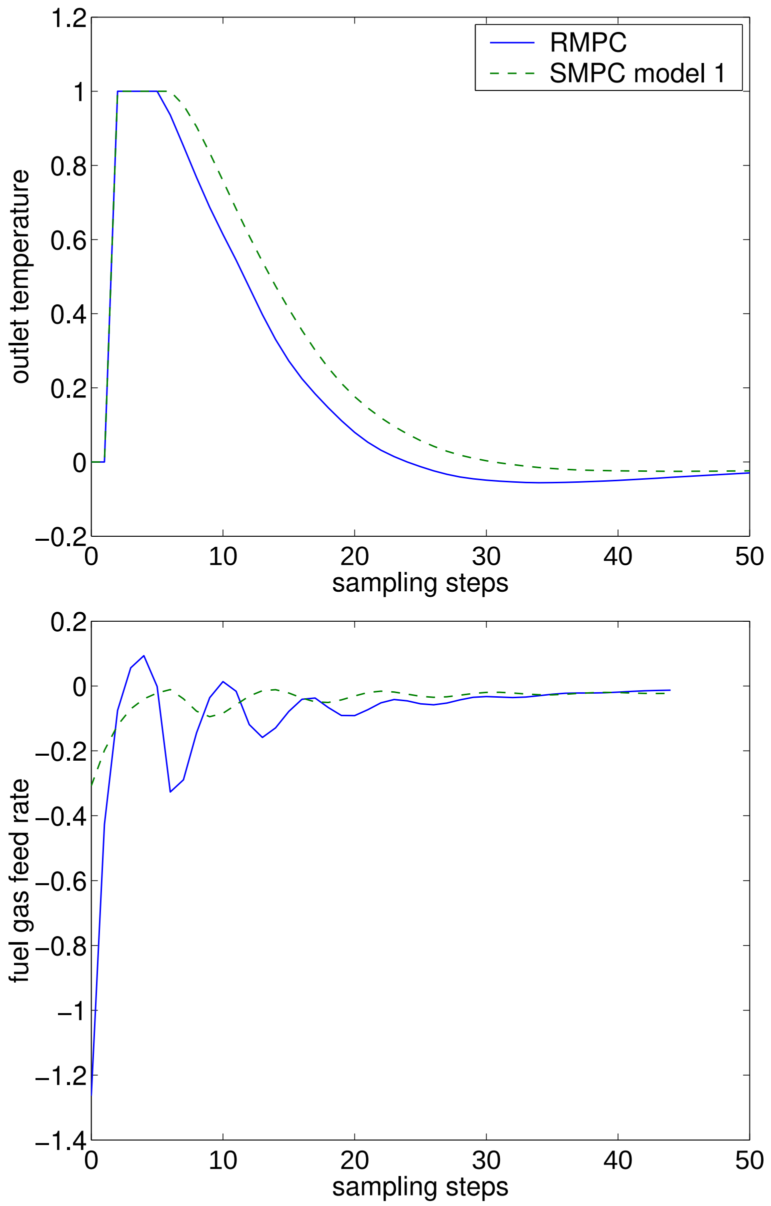

Figure 6.33: Comparison of disturbance rejection for the fired heater system with model 2 as the plant.

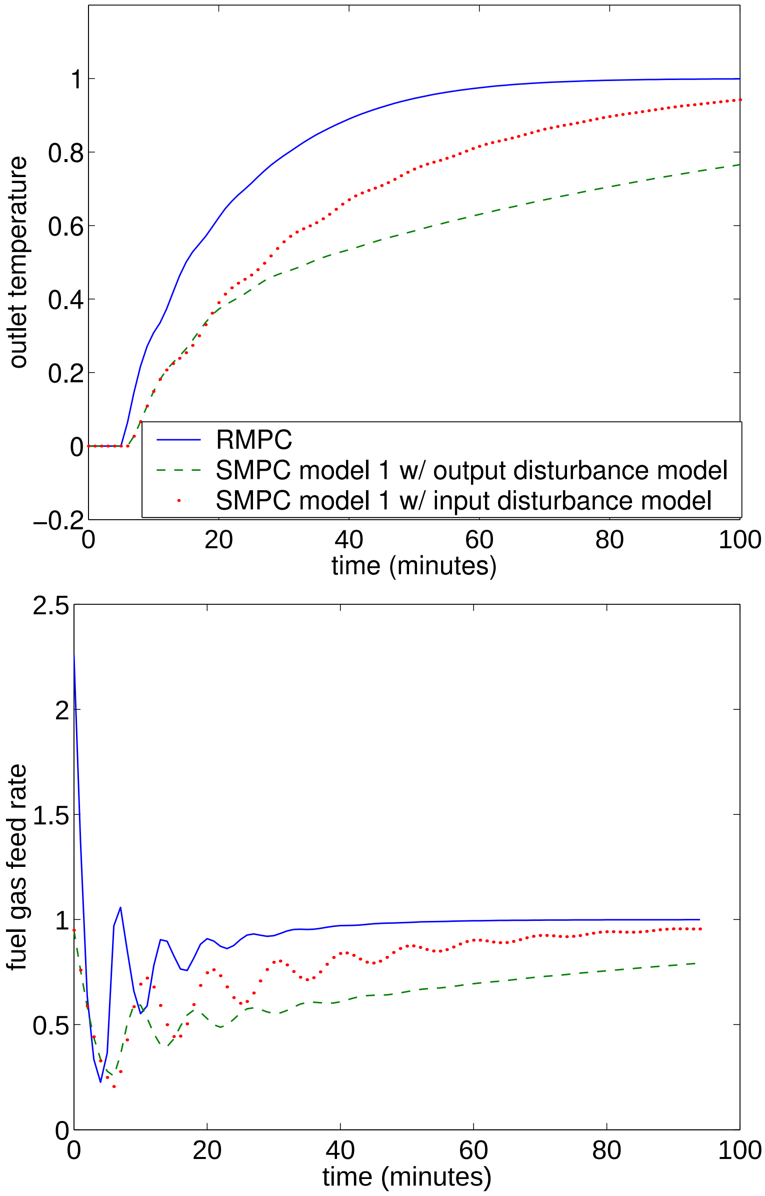

Figure 6.34: Comparison of the closed-loop control performance of the RMPC and SMPC with input or output disturbance model for the fired heater system with model 2 as the plant.

long time delays and uncertainties as to where the discrepancy between the estimated and the predicted state is coming from. Figure 6.34 compares the SMPC algorithm using model 1 with the input disturbance model to the SMPC using model 1 with the output disturbance model. Even though the input disturbance model does cause the SMPC algorithm with plant-model mismatch to reach the new set point faster than the output disturbance model, the RMPC control performance is still more efficient.

6.3 Multiple-Input-Multiple-Output Examples

6.3.1 Ill-conditioned distillation column

Figures 3.8 and 3.9 show that if no control actions are needed, the addition of a zone region to y_D is sufficient at maintaining closed-loop stability for the ill-conditioned column. But as soon as control actions are needed to maintain y_D inside its zone region and x_B at its set point, the SMPC controller is closed-loop unstable in the presence of plant-model mismatch. The models in Equations 3.23 and 3.24 have gain matrices with determinants of the opposite signs. By Lemma 5.9.3, offset free control cannot be achieved for the convex hull defined by Equations 3.23 and 3.24 because there is a model in the convex hull that is rank deficient.

- Model 1:

$$\begin{aligned}\begin{bmatrix} y_D(t+1) \\ x_B(t+1) \end{bmatrix} &= \begin{bmatrix} 0.8187 & 0 \\ 0 & 0.8187 \end{bmatrix} x(t) \\ &+ \begin{bmatrix} 0.2393 & -0.1563 \\ 0.1958 & -0.1985 \end{bmatrix} \begin{bmatrix} L(t) \\ V(t) \end{bmatrix} \\ y(t+1) &= x(t+1)\end{aligned} \tag{6.7}$$

- Model 2:

$$\begin{aligned}\begin{bmatrix} y_D(t+1) \\ x_B(t+1) \end{bmatrix} &= \begin{bmatrix} 0.8187 & 0 \\ 0 & 0.8187 \end{bmatrix} x(t) \\ &+ \begin{bmatrix} 0.1591 & -0.1563 \\ 0.1958 & -0.1985 \end{bmatrix} \begin{bmatrix} L(t) \\ V(t) \end{bmatrix} \\ y(t+1) &= x(t+1)\end{aligned} \tag{6.8}$$

Equation 6.8 is the discrete-time state-space model for Equation 3.23. By changing the $(1,1)$ element of the B matrix from 0.1591 to 0.2393, the condition number decreases from 132 (for Equation 6.8) to 9.33 (for Equation 6.7). There is no model in the convex hull that is rank deficient. Offset free control can be achieved if there are stabilizing K, F, L_1, and L_2. The RMPC Algorithm 5.9.2 found the stabilizing K, F, L_1, and L_2 for the tuning parameters $R = I$, $Q_s = I$, and $Q = C^T Q_s C$. The SMPC controller found the nominal K, L_x, and L_d parameters using the same regulator tuning parameters as RMPC and the estimator tunning parameters $R = Q_w = I$. Figures 6.35-6.39 show the closed-loop response of the

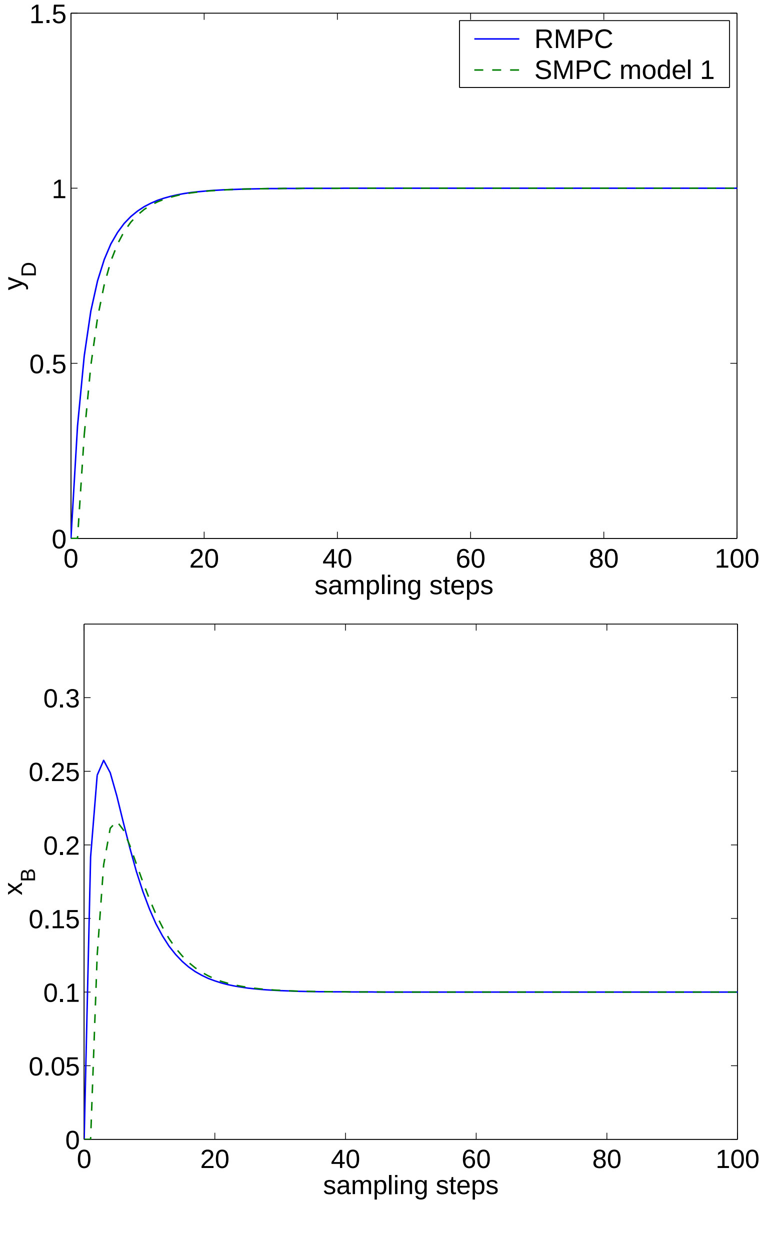

Figure 6.35: Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant.

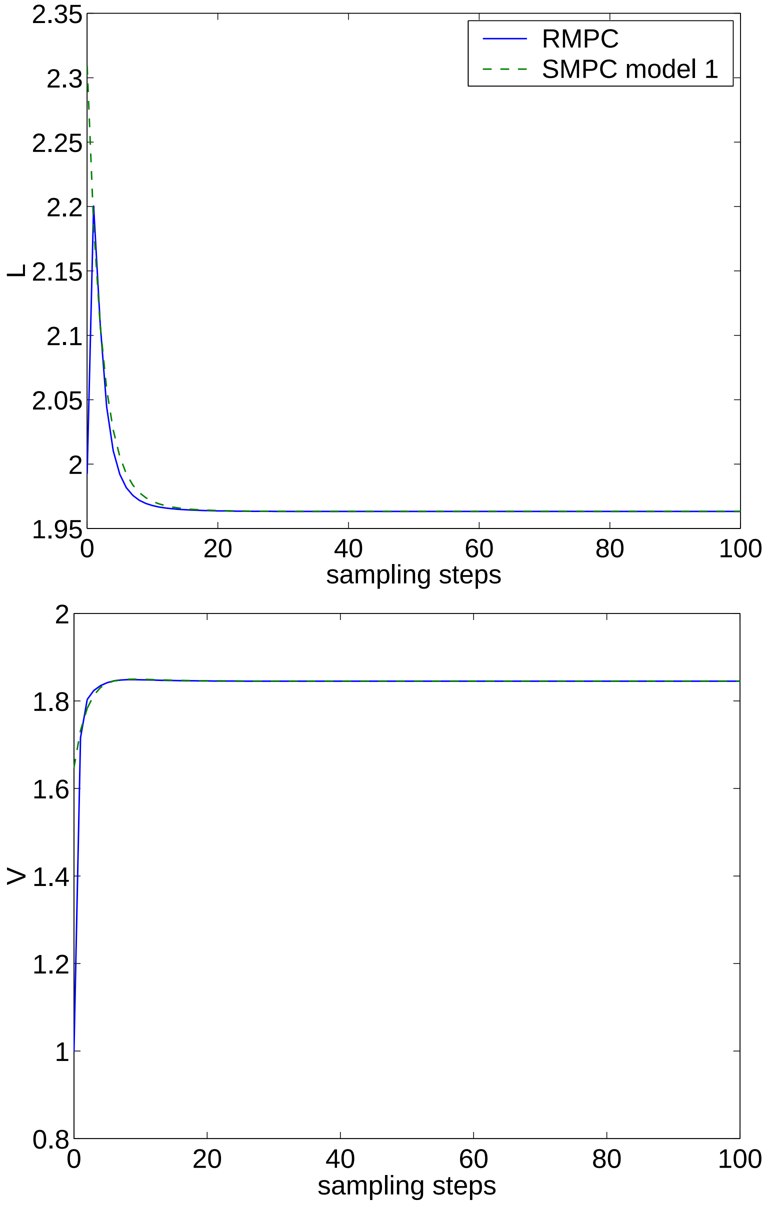

Figure 6.36: Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant.

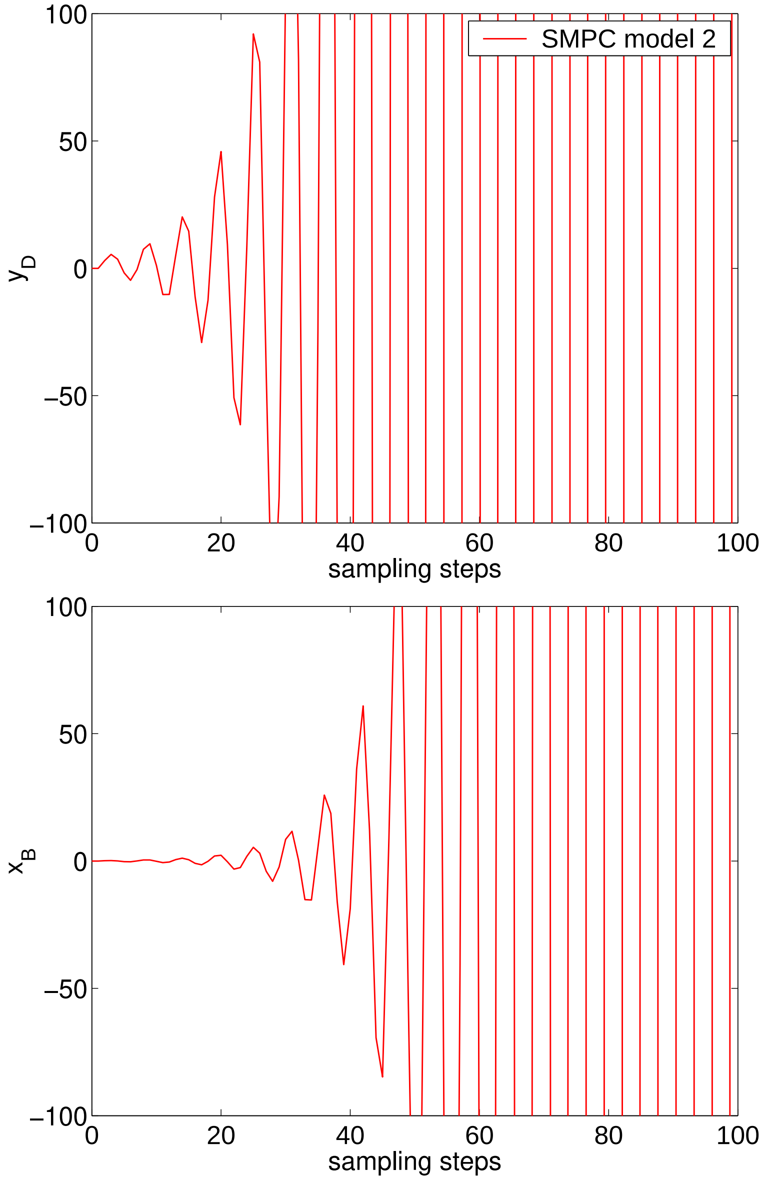

Figure 6.37: Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant.

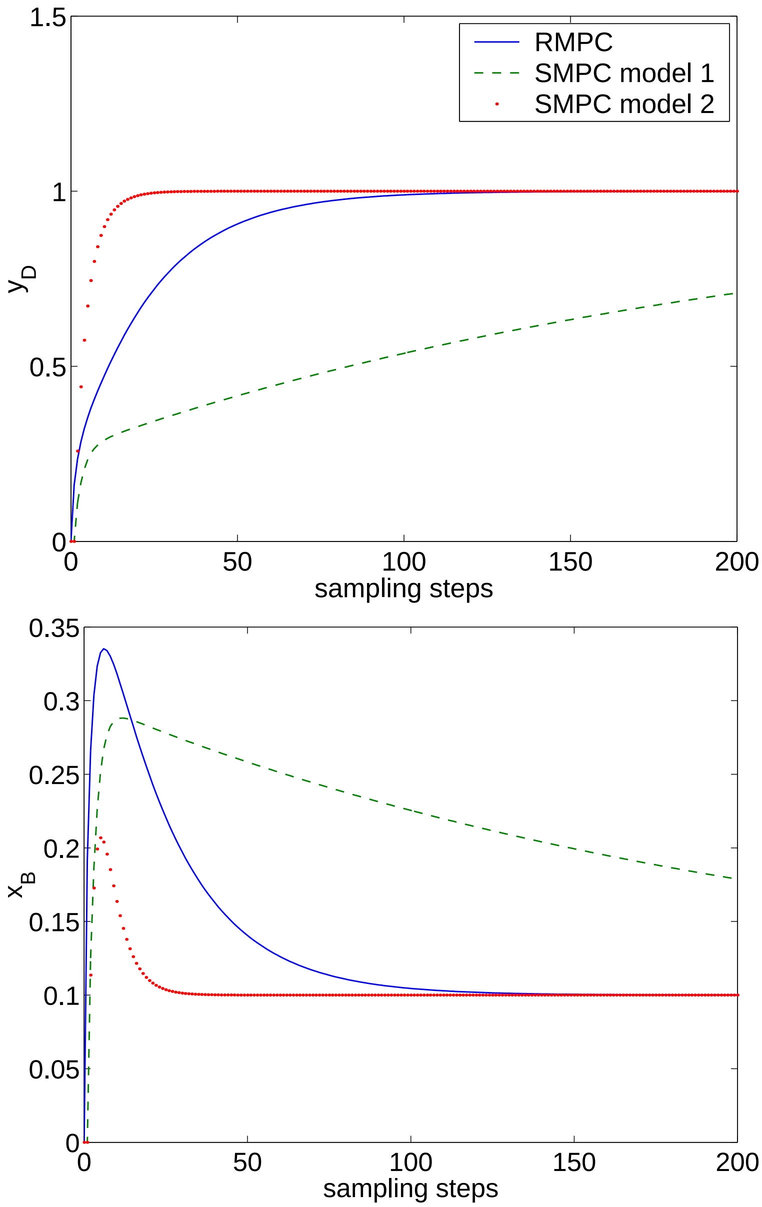

Figure 6.38: Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant.

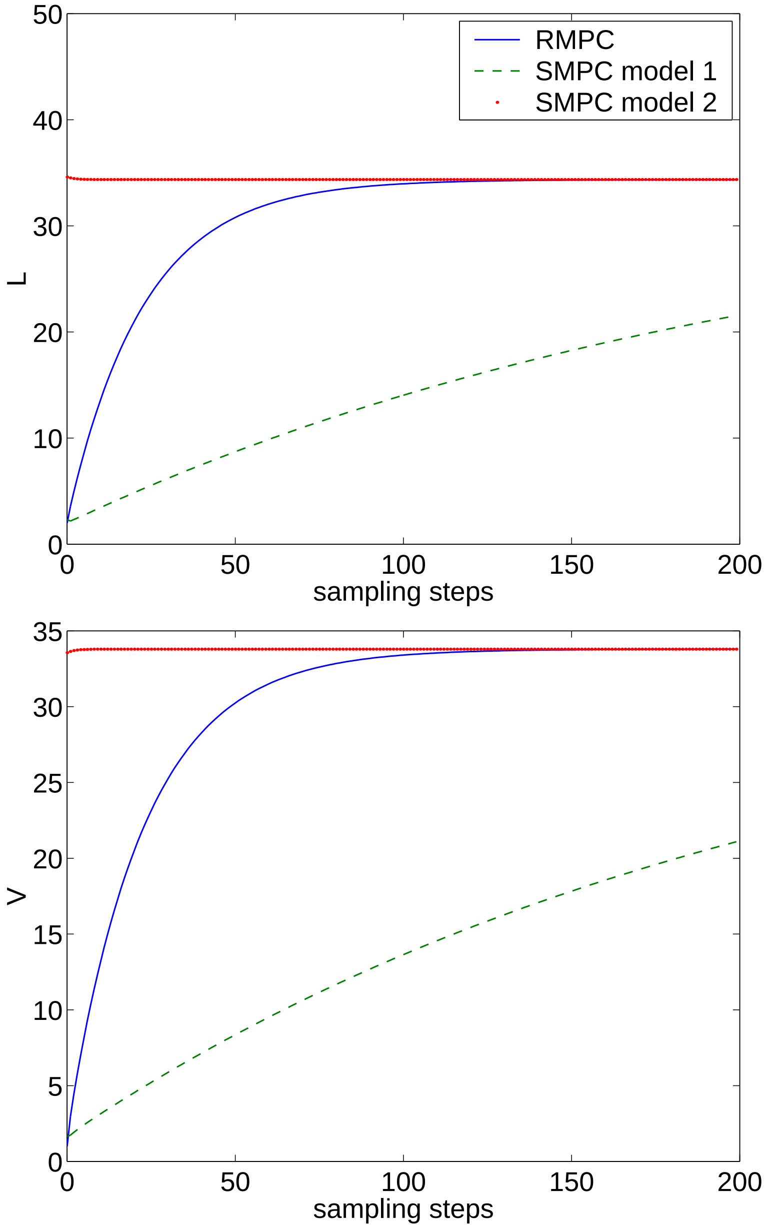

Figure 6.39: Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant.

distillation column for a set point change from 0 to 1 for y_D and from 0 to 0.1 for x_B. SMPC using model 2 is closed-loop unstable for the plant equal to model 1 due to the ill-conditioning of model 2; SMPC using model 1 is much slower than RMPC at controlling the column for the set point change when the plant is model 2.

When plant-model mismatch is present, SMPC using model 1 is too conservative while SMPC using model 2 is too aggressive, which caused the closed-loop instability. SMPC using model 2 is closed-loop stable when the plant is equal to model 1 if the estimator tuning parameters are changed to $R_v = 1000Q_w$ which implies the operator believes the sensor measurements are more accurate than the state forecasts. Figures 6.40 and 6.41 compare the closed-loop control performance of the RMPC to SMPC using models 1 and 2 with the new estimator tuning parameters given above for SMPC using model 2. Even though SMPC using model 2 is closed-loop stable, its control performance is more oscillatory than either the RMPC or the nominal SMPC control performance. Figures 6.42 and 6.43 show that for the plant equal to model 2, the ill-conditioned model, the nominal SMPC control performances for the two different set of tuning parameters are similar. SMPC using model 2 gained robustness to plant-model mismatch without sacrificing too much on performance for the nominal plant.

Figures 6.44 and 6.45 compare the closed-loop SMPC control performance to the RMPC control performance when the plant is randomly time-varying within

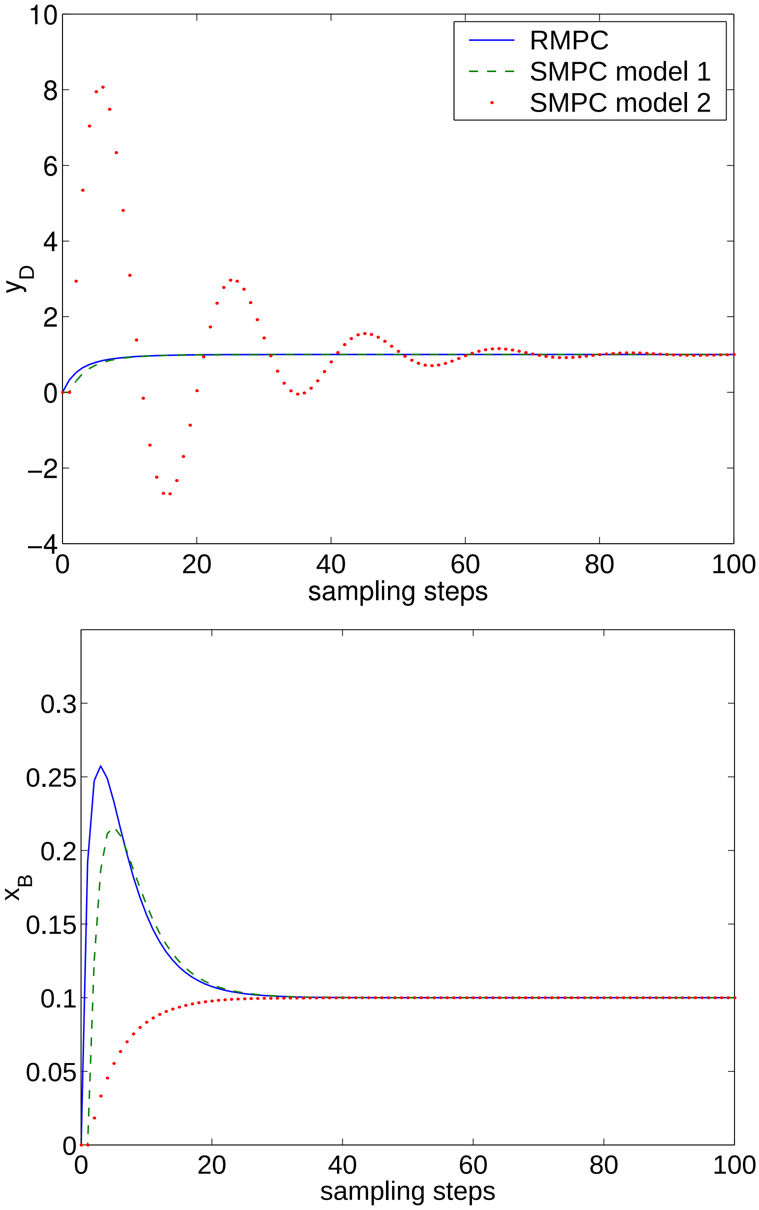

Figure 6.40: Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant and new Q_w and R_v for SMPC.

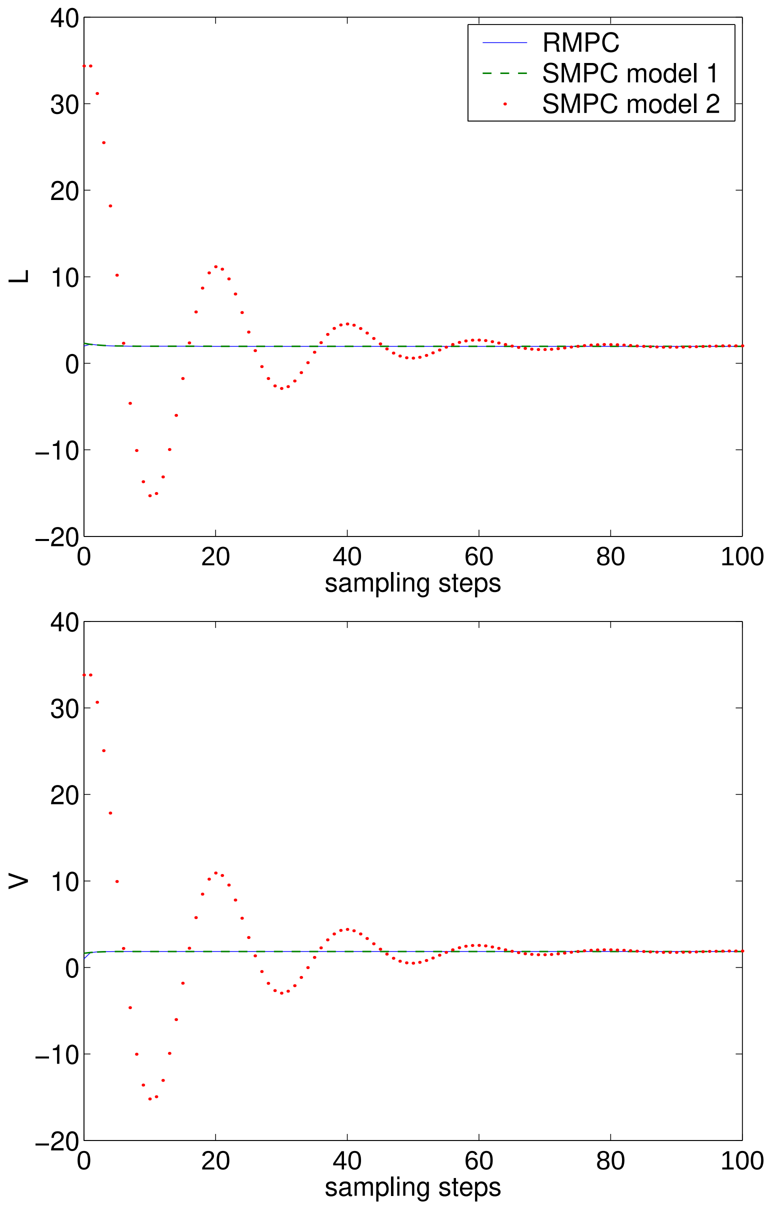

Figure 6.41: Closed-loop control performance for the ill-conditioned distillation column with model 1 as the plant and new Q_w and R_v for SMPC.

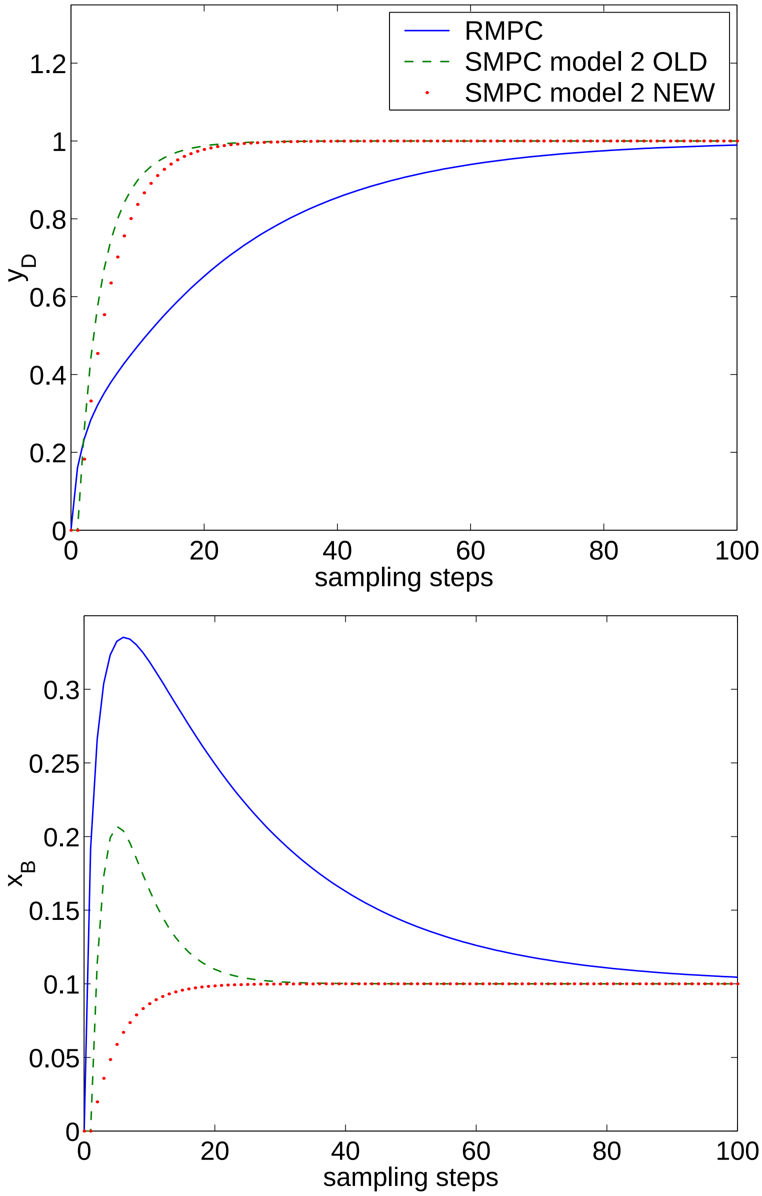

Figure 6.42: Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant and new Q_w and R_v for SMPC.

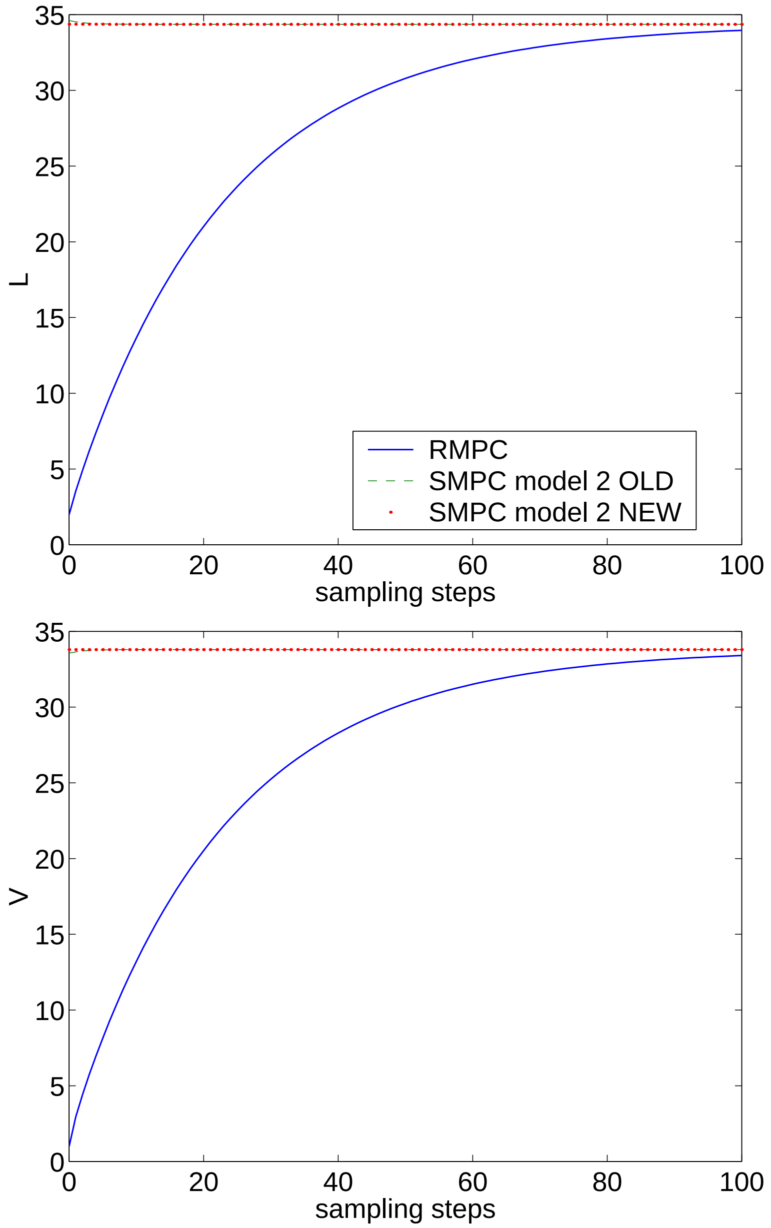

Figure 6.43: Closed-loop control performance for the ill-conditioned distillation column with model 2 as the plant and new Q_w and R_v for SMPC.

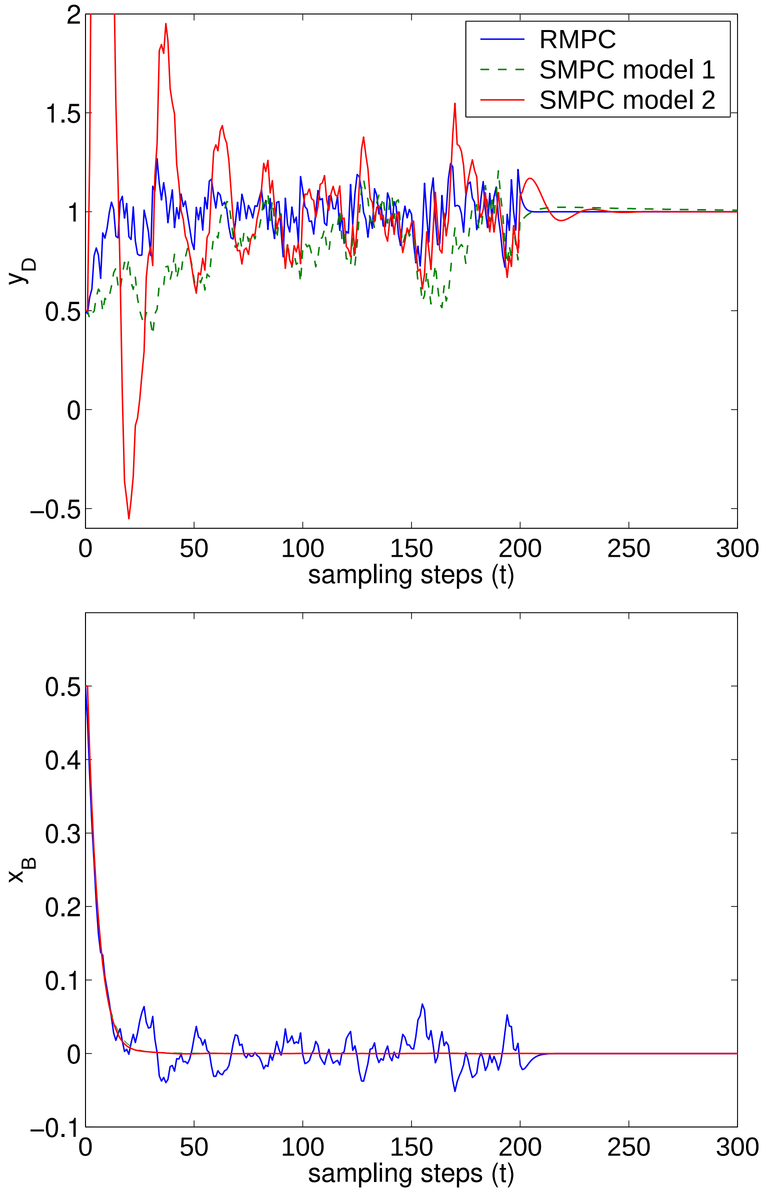

Figure 6.44: Closed-loop control performance for the ill-conditioned distillation column with time-varying plant and the new Q_w and R_v for SMPC using model 2.

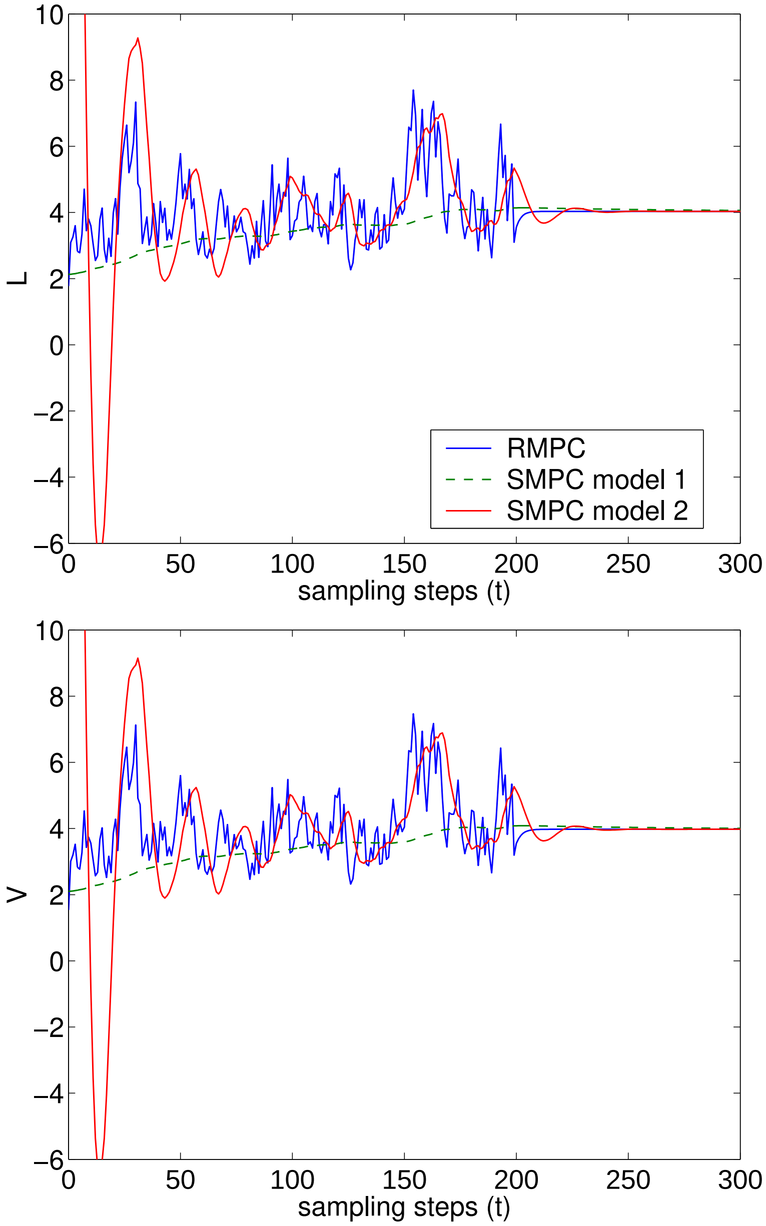

Figure 6.45: Closed-loop control performance for the ill-conditioned distillation column with time-varying plant and the new Q_w and R_v for SMPC using model 2.

the convex hull between the time steps 0 and 200. After $t = 200$, the plant is equal to the average of models 1 and 2. The objective is control the system from an initial condition of $y_D = 0.5$ and $x_B = 0.5$ to the new set point of $y_D = 1$ and $x_B = 0$. The closed-loop responses in x_B for SMPC using models 1 and 2 are identical because there is no uncertainty in A and no uncertainty in the elements of the B matrix that describe the effect of changes in L and V on x_B. On the other hand, their responses in y_D are very different due to the uncertainty in the $(1,1)$ element of the B matrices. Figure 6.44 shows that when the plant is time-varying, the SMPC using model 1 control actions are the least oscillatory of the three controllers. It may appear that SMPC using model 1 is the better controller for the ill-conditioned column, but recall Figure 6.38 shows that SMPC using model 1 is overly conservative when the plant is model 2. The SMPC using model 2, the model with the higher condition number is stable but more oscillatory even with the new tuning parameters. We conclude, therefore, the RMPC controller is the best of the three controllers studied in Figures 6.38-6.44 for the ill-conditioned distillation column with model uncertainty described by the convex hull of Equations 6.7 and 6.8.

6.3.2 Three-by-three system

Ogunnaike and Ray [112] presented the Hulbert and Woodburn wet-grinding circuit with the following transfer function model obtained by experimental means.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \frac{119}{217s+1} & \frac{153}{337s+1} & \frac{-21}{10s+1} \\ \frac{0.00037}{500s+1} & \frac{0.000767}{33s+1} & \frac{-0.00005}{10s+1} \\ \frac{930}{500s+1} & \frac{-667}{166s+1} & \frac{-1033}{47s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{6.9}$$

in which

- y_1 = Torque required to turn the mill (Nm),
- y_2 = Flow rate from the mill (m^3/s),
- y_3 = Density of the cyclone feed (kg/m^3),
- u_1 = Feed rate of the solids to the mill (kg/s),
- u_2 = Feed rate of the water to the mill (kg/s),
- u_3 = Feed rate of the ater to the sump (kg/s).

Figures 6.46 and 6.47 compare the closed-loop RMPC and SMPC control performances for uncertainties in the B matrices only. After identifying the ARX model that corresponds to the transfer function model in Equation 6.9, the uncertainties are defined as

$$B_1 = 0.5B_0 \qquad \text{and} \qquad B_2 = 1.5B_0$$

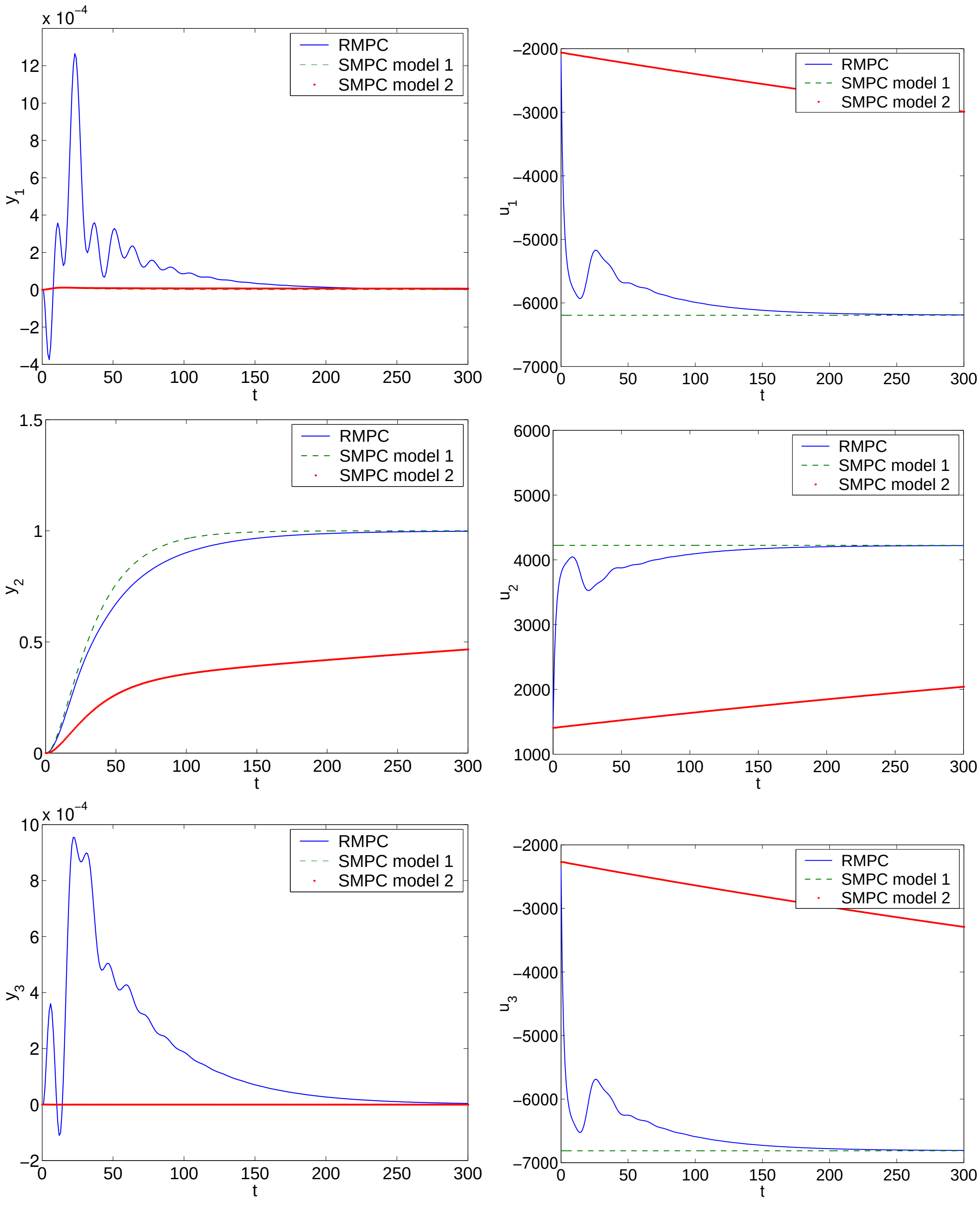

Figure 6.46: Closed-loop control performance for the three by three system with model 1 as the plant.

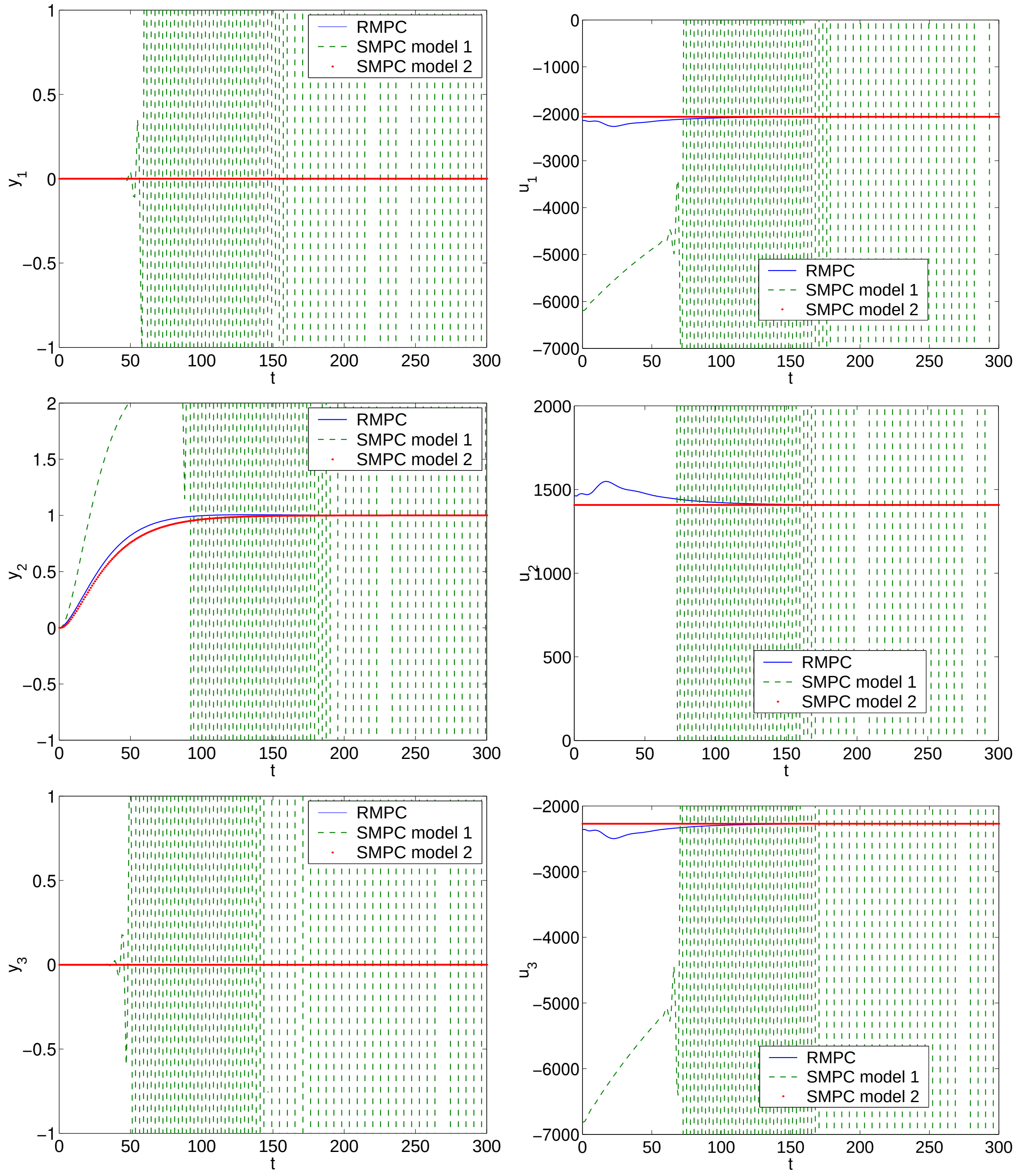

Figure 6.47: Closed-loop control performance for the three by three system with model 2 as the plant.

in which B_0 is the B matrix for the model in Equation 6.9. There is no uncertainty in the A matrix. The objective is to control the system for a set point change from 0 to 1 for y_2. Figure 6.46 shows the SMPC with plant-model mismatch is closed-loop stable when the plant is model 1, but the control performance is very sluggish. When the plant is model 2, SMPC with plant-model mismatch is closed-loop unstable while RMPC is successful at controlling the system for the required set point change in y_2.

Chapter 7

Conclusion

The success of the single-model MPC controller design procedure is dependent on the accuracy of the process model. Modeling errors cause sub-optimal control performances and may cause the control system to become closed-loop unstable. The goal of this research project is to design a new controller that can maintain closed-loop stability in the presence of model uncertainty and process constraints. Besides stability, the controller needs to be able to achieve non-zero set point tracking and disturbance rejection for time-varying uncertainties.

7.1 Single-Model MPC with Zone Regions

The first proposal to increase the robustness of the SMPC controller is to decrease the stringentness of the control objectives by changing some of the set points to zone regions. When a controlled output has a zone region as its control objective, the controller takes control actions only when the output deviates from the zone limits. Modeling errors may cause the output to wander inside the zone, but as long as it remains inside, no control actions are taken.

The addition of the zone regions is effective at maintaining closed-loop stability only when the modeling errors are not large enough to cause the controlled outputs to deviate from the zone regions. As soon as control actions become necessary to return the controlled output back inside the zone, closed-loop stability cannot be guaranteed. The amount of model uncertainty that SMPC with zones can tolerate and remain closed-loop stable is dependent on the size of the zone region. The larger the zone region, the more model uncertainty the SMPC can tolerate and still remain closed-loop stable. In the extreme case, the SMPC controller with zone regions is closed-loop stable for all open-loop stable processes if the zone limits are equal to $y_{\min}$ and $y_{\max}$, the output constraints. The system is uncontrolled.

7.2 Robust MPC

The robust MPC I propose incorporate model uncertainty directly into the controller design procedure. For a process that can be described by a time-varying combination of process models in the convex hull of Π, a set of (A_i, B_i), the objectives of the RMPC controller are:

- maintain the system with time-varying model uncertainty at a given set point by rejecting non-zero exogenous disturbances,

- move the system with time-varying model uncertainty from one set point to another without offset,
- satisfy all process constraints,
- achieve offset free control with output feedback.

I propose a new RMPC method that includes a closed-loop stability condition that determines if offset-free control is possible given the model uncertainty description. The stability condition examines the effects of the robust regulator, target calculation, and state estimation on the closed-loop stability. Offset-free control in the presence of model uncertainty is accomplished by adding steady-state biases to the steady-state equality constraints so that for a single u_s value, all models in the model uncertainty description have the same steady-state x_s value. The target calculation determines the x_s and u_s values necessary to achieve offset-free control. The tree trajectory in the robust regulator is designed to forecast the state for a system with time-varying uncertainty. Offset-free control is achieved when all branches in the tree trajectory converge to the x_s and u_s values.

Table 7.1 summarizes the examples examined in this paper. The examples include systems with:

- process dynamic uncertainty,

- process gain uncertainty,
- both stable and unstable models,
- time-delay uncertainty,
- constraint saturation,
- ill-conditioned models,
- multivariable models,
- integrating models,
- uncertainty set containing as many as three models.

The RMPC method I propose is successful at controlling all of the above examples for non-zero set point changes with time-varying uncertainties and input and output constraints. The new RMPC controller increases the tolerance of the controller to model uncertainty, but the decrease in nominal performance is relatively small for most of the examples studied. The RMPC control performance is almost as good as the nominal SMPC performance. Although finding the K, F, and L_i's that satisfy the closed-loop stability condition proposed in Chapter 5 may be difficult, the calculation is performed off-line and does not affect the on-line implementation of the RMPC method. The on-line implementation cost of the RMPC method is the computationally intensive *min-max* optimization needed to compute the control actions in the tree trajectory. From my experience, however, the prediction horizon needed before the branches reach the terminal invariant region is usually small.

Even though the examples show the new RMPC method is successful at controlling systems with time-varying model uncertainty for non-zero set point tracking subject to input and output constraints, many challenges remain. Model identification is an open issue. The model uncertainty set affects both the closed-loop stability algorithm and the on-line performance of the *min-max* optimization. The best way to characterize model uncertainty remains an open question.

Even though we only need feasible K, F and L_i's that satisfy the closed-loop stability condition, the proposed algorithm does not guarantee the existence of a solution. For systems with large numbers of states, inputs, measured outputs, and models in the uncertainty set, the closed-loop stability algorithm is computationally intensive. Because the algorithm is performed off-line, however, it does not affect the on-line implementability of the RMPC method. For large systems, the on-line *min-max* optimization is computationally intensive as well, especially for long prediction horizon length N. I have shown for systems with up to three models in the uncertainty set and models with up to seven states, that as long as the prediction horizon length is relatively small, the on-line optimization can be accomplished in a reasonable amount of time. A more extensive study of industrially relevent examples is needed to fully understand the possible impact of this RMPC method.

Models	$\mathcal{I}$	m	n	p	δ[a]	Const.	OL stability[b]	Figure	Obj.[c]	Plant	Stable controller[d]
$A_1 = A_2 = 0.4, B_1 = 0.1, B_2 = 0.7$	2	1	1	1	no	no	stable	6.1, 6.2	s	model 2	R, $\mathrm{S}_{2,d}$
$A_1 = A_2 = 0.4, B_1 = 0.1, B_2 = 0.4$	2	1	1	1	no	no	stable	6.3, 6.4	s	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
$A_1 = A_2 = 0.4, B_1 = 0.1, B_2 = 0.4$	2	1	1	1	no	no	stable	6.5	s	model 1	R,$\mathrm{S}_{1,d}$
$A_1 = 0.5, A_2 = 1.6, B_1 = B_2 = 1$	2	1	1	1	no	no	unstable	6.6	s	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
$A_1 = 0.5, A_2 = 1.6, B_1 = B_2 = 1$	2	1	1	1	no	no	unstable	6.7	s	model 2	R,$\mathrm{S}_{2,d}$
Inverse response	2	1	3	1	no	no	stable	6.9	s	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Inverse response	2	1	3	1	no	no	stable	6.10	s	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Oscillatory system	3	1	2	1	no	no	stable	6.13	s	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$,$\mathrm{S}_{3,d}$
Oscillatory system	3	1	2	1	no	no	stable	6.14	s	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$,$\mathrm{S}_{3,d}$
Oscillatory system	3	1	2	1	no	no	stable	6.15	s	model 3	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$,$\mathrm{S}_{3,d}$
Underdamped system	3	1	2	1	no	no	stable	6.17	s	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Underdamped system	3	1	2	1	no	no	stable	6.18	s	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Underdamped system	3	1	2	1	no	no	stable	6.19, 6.20	s	model 3	R,$\mathrm{S}_{2,d}$,$\mathrm{S}_{3,d}$
Integrating system[e]	2	1	2	1	no	no	unstable	6.23	s	model 1	R,$\mathrm{S}_{1,d}$
Integrating system	2	1	2	1	no	no	unstable	6.24	s	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Integrating system	2	1	2	1	no	no	unstable	6.25	s	$\mu_1 = 0.101$	R,$\mathrm{S}_{\mu_1=0.9091,d}$
Constraint saturation	2	1	1	1	no	yes	stable	6.26,6.27	s,d	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Heater with time-delay	2	1	7	1	yes	no	stable	6.29,6.32	s,d	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Heater with time-delay	2	1	7	1	yes	no	stable	6.30,6.34, 6.33	s,d	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{1,z}$,$\mathrm{S}_{2,d}$
Heater with time-delay	2	1	7	1	yes	no	stable	6.31	s	$\mu_1 = 0.5$	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Ill-conditioned column	2	2	2	2	no	no	stable	6.40,6.41	s	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Ill-conditioned column	2	2	2	2	no	no	stable	6.38,6.39	s	model 2	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Three-by-three system	2	3	9	3	no	no	stable	6.46	s	model 1	R,$\mathrm{S}_{1,d}$,$\mathrm{S}_{2,d}$
Three-by-three system	2	3	9	3	no	no	stable	6.47	s	model 2	R,$\mathrm{S}_{2,d}$

Table 7.1: Summary

[a]Time delay
[b]Open-loop stability
[c]s = set point change, d = disturbance rejection
[d]R = RMPC, $\mathrm{S}_{i,d}$ = SMPC using model i and $d(t)$, $\mathrm{S}_{i,z}$ = SMPC using model i and $z(t)$.
[e]Angular positioning system

Appendix A

Notation

Scalars

i	The model index.
j	The number of sampling time steps forecasted in the prediction horizon or the level of the tree trajectory.
l	The branch index for the tree trajectory.
m	The dimension of the manipulated input vector $u(t)$.
n	The dimension of the state vector $x(t)$.
p	The dimension of the controlled output vector $y(t)$.
t	The current sampling time.
N	The finite prediction horizon length.
δ	The time delay.
η	The total number of nodes in the tree trajectory.

$\mu_i(t)$	The parameter specifying the linear combination for (A_i, B_i) necessary to form $(A, B) \in \Omega$. $0 \leq \mu_i(t) \leq 1$ for all $i = 1, \ldots, \mathcal{I}$ and $\sum_{i=1}^{\mathcal{I}} \mu_i(t) = 1$.
$\mathcal{I}$	The total number of model in Π.
$\mathcal{L}$	The total number of branches in the tree trajectory.
Φ	The performance objective.
Φ^*	The optimal performance objective.
$\Phi(t)$	The performance objective at time t.
$\Phi^*(t)$	The maximum performance objective at time t.
$\Phi_l(t)$	The performance objective at time t for branch l in the tree trajectory.
$\Phi_l^*(t)$	The optimal performance objective at time t for branch l.
Φ_s	The steady-state performance objective.
Φ_N	The infinite horizon regulator terminal cost, $x^T(t+N)Fx(t+N)$.
$\mathfrak{C}$	is the compensator for a SISO system.
$\mathfrak{L}$	is the transfer function $\mathfrak{L} = \mathfrak{P}\mathfrak{C}$ for a SISO system.
$\mathfrak{P}$	is the plant transfer function for a SISO system.
$\mathfrak{S}$	is the sensitivity function for a SISO system.
$\mathfrak{T}$	is the closed-loop complementary sensitivity function.

$\mathfrak{W}$	is the bound for the relative perturbation of $\mathfrak{L}_0$.

Vectors

$d(t)$	Output disturbance vector used as an integrator.
$\hat{d}(t\|t-1)$	Output disturbance estimate at time t with measurements up to time $t-1$.
$p_i(t)$	The state disturbance for model i at time t.
$p_{i,s}$	The steady-state bias for model i.
$u(t)$	Input vector at time t.
$u(t+j\|t)$	Input vector at time j in the prediction horizon with measurement up to time t.
$\overline{u}(t)$	Input vector at time t in deviation variable form. $\overline{u}(t) = u(t) - u_s$.
$u_\tau(t)$	The input vector at node τ and time t.
$\overline{u}_\tau(t)$	Input vector at node τ and time t in deviation variable form.
$u^*(t)$	The optimal control move at time t.
$u_{\min}, u_{\max}$	Minimum and maximum input constraints.
u_s	The steady-state input vector.
u_t	The input target.
$v(t), w(t)$	Zero-mean, uncorrelated, normally distributed, stochastic variables modeling output and state noise.

$x(t)$	State vector at time t.
$x(t+j\|t)$	State vector at time j in the prediction horizon with measurement up to time t.
$\overline{x}(t)$	State vector at time t in deviation variable form. $\overline{x}(t) = \overline{x} - x_s$.
$x_\tau(t)$	State vector at node τ and time t.
$\overline{x}_\tau(t)$	State vector at node τ and time t in deviation variable form.
$x_{\text{min}}, x_{\text{max}}$	The minimum and maximum state constraints.
x_s	The state vector at steady state.
x_t	The state set point.
$\hat{x}(t+1\|t)$	The state estimate at $t+1$ with measurement up to t.
$x^{(r)}$	The r^{th} element of the vector $x \in \mathbb{R}^n$.
$y(t)$	The output vector at time t.
$y(t+j\|t)$	The output vector at time j in the prediction horizon with measurement up to time t.
$y_\tau(t)$	The output vector at node τ and time t.
$\overline{y}(t)$	The output vector at time t in deviation variable form. $\overline{y}(t) = y(t) - y_s$.
$\overline{y}_\tau(t)$	The output vector at node τ and time t in deviation variable form.
$y_{\text{lo}}, y_{\text{hi}}$	The lower and upper zone limits.

$y_{\min}, y_{\max}$	The minimum and maximum output constraints.
y_s	The output vector at steady state.
y_t	The output set point.
$z(t)$	The input disturbance vector used as an integrator.
$\hat{z}(t\|t-1)$	The input disturbance estimate at time t with measurements up to time $t-1$.
ϵ	The slack variable that measures the difference between $y(t)$ and either y_{lo} or y_{hi}.
$\pi(t)^*$	The optimal input trajectory computed at time k.
τ	The node index for the tree trajectory.
τ_l	The node index for the terminal node of branch l.
θ	is the parameter vector used to specify the uncertain process model (A, B, C).
θ_0	is the parameter vector for the nominal model in the model uncertainty description.
$\xi(t)$	The output tracking error.
T_l	The set of all nodes on branch l.
$U_l(t)$	The control sequence at time time t for branch l.

Matrices

$A_{11}, A_{12}, A_{21}, A_{22}$	The partitioned A matrices.
B_1	The partitioned B matrix.
C_1	The partitioned C matrix.
A, B, C	The state-space model matrices.
$A(t), B(t), C$	The time-varying state-space model matrices.
A_i, B_i	The state-space model matrices for model i in Π.
A_0, B_0	The nominal model in the model uncertainty description.
A_m, B_m, C_m	The state-space nominal MPC algorithm model.
A_p, B_p, C_p	The state-space plant model.
$\tilde{A}_c$	The augmented closed-loop stability matrix for the nominal MPC algorithm with or without plant model mismatch.
$\tilde{A}'_c$	The augmented closed-loop stability matrix for the nominal MPC algorithm without plant model mismatch.
$\tilde{A}_d, \tilde{B}_d, \tilde{C}_d$	The augmented state-space model matrices with output disturbance model $d(t)$.
$\tilde{A}_z, \tilde{B}_z, \tilde{C}_z$	The augmented state-space model matrices with input disturbance model $z(t)$.
$\Delta A, \Delta B$	The parametric uncertainty for (A, B).

F	The terminal state penalty matrix.
G_i	The gain matrix for model i. $G_i = C(I - A_i)^{-1}B_i$.
G_w	The state disturbance matrix.
$\tilde{G}_w$	The augmented state disturbance matrix.
K	The linear state feedback gain.
L	The Kalman filter.
L_d, L_x, L_z	The partitioned Kalman filter.
L_i	The state disturbance filter for model i.
P, Y	The LMI variables needed to find K and F.
P_i	The symmetric, positive definite penalty matrix for the difference between the steady-state bias $p_{i,s}$ and the state disturbance $p_i(t)$ for model i.
Q	The state penalty matrix.
$\bar{Q}$	The output penalty matrix.
Q_s	The steady-state output penalty matrix.
Q_w	The covariance matrix for $w(t)$.
R	The input penalty matrix.
R_s	The steady-state input penalty matrix.
R_v	The covariance matrix for $v(t)$.

S	The penalty matrix for the output deviations from the zone limits.
W	The state's terminal region.
W_d	$W_d = W - x_s$
Π	The set of models that define the vertices of the model uncertainty description. $\Pi = \{(A_1, B_1), \ldots, (A_{\mathcal{I}}, B_{\mathcal{I}})\}$
Ω	The convex hull of Π.
$\mathbf{C}$	The controllability matrix.
$\mathbf{O}$	The observability matrix.
$\mathcal{Q}$	The terminal state penalty matrix.

Mathematical Symbols

$\mathbb{C}^n$	The set of all complex numbers in n-space.
$\mathbb{R}^n$	The set of all real numbers in n-space.
$\mathbb{R}^{n \times m}$	The set of all real numbers in the matrix with n rows and m columns.
$\mathbb{U}$	The constrained subspace for the input vector.
$\mathbb{U}_d$	$\mathbb{U}_d = \mathbb{U} - u_s$
$\mathbb{X}$	The constrained subspace for the state vector.
$\mathbb{X}_d$	$\mathbb{X}_d = \mathbb{X} - x_s$

$\mathbb{Y}$	The constrained subspace for the output vector.
$\mathbb{Y}_d$	$\mathbb{Y}_d = \mathbb{Y} - y_s$

Acronym

LMI	Linear Matrix Inequalities
MPC	Model Predictive Control
RMPC	Robust Model Predictive Control
SMPC	Single-Model Model Predictive Control

Appendix B

Definition

Below are some basic linear algebra definitions used in this report

Definition B.0.1 *The* **Euclidean norm** *on* $\mathbb{C}^n$ *is*

$$\| \chi \|_2 \equiv (|\chi_1|^2 + \cdots + |\chi_n|^2)^{\frac{1}{2}}.$$

Definition B.0.2 *The* **infinity norm** *on* $\mathbb{C}^n$ *is*

$$\| \chi \|_\infty \equiv \max\{|\chi_1|, \ldots, |\chi_n|\}.$$

Definition B.0.3 $\chi \in \mathbb{R}^n$*, if* $\| \cdot \|_\alpha$ *and* $\| \cdot \|_\beta$ *are norms on* $\mathbb{R}^n$*, then there exist positive constants* c_1 *and* c_2 *such that*

$$c_1 \| \chi \|_\alpha \le \| \chi \|_\beta \le c_2 \| \chi \|_\alpha .$$

For $\alpha = \infty$ *and* $\beta = 2$*,* $c_1 = 1$ *and* $c_2 = \sqrt{n}$

$$\| \chi \|_\infty \le \| \chi \|_2 \le \sqrt{n} \| \chi \|_\infty . \tag{B.1}$$

See Golub and Van Loan [55] for more information.

Definition B.0.4 *The* **triangle inequality** *states for any two vectors* $\chi \in \mathbb{R}^n$ *and* $\varphi \in \mathbb{R}^n$ *is*

$$\| \chi + \varphi \|_2 \leq \| \chi \|_2 + \| \varphi \|_2 . \tag{B.2}$$

If $\chi \geq 0$ *and* $\varphi \geq 0$ *are scalars,*

$$\| \chi + \varphi \|_2 = \sqrt{\chi^2 + \varphi^2}$$

$$\| \chi \|_2 = \chi \quad \text{and} \quad \| \chi \|_2 = \varphi.$$

The triangle inequality becomes

$$\begin{aligned} \| \chi + \varphi \|_2 &\leq \chi + \| \varphi \|_2 \\ &\leq \chi + \varphi. \end{aligned}$$

Definition B.0.5 $S \in \mathbb{R}^{n\times n}$ *is* **positive definite** *if*

$$\chi^* S \chi > 0 \quad \forall \chi \neq 0 \quad \text{and} \quad \chi \in \mathbb{C}^n.$$

$S > 0$ *indicates* S *is positive definite.*

Definition B.0.6 $S \in \mathbb{R}^{n\times n}$ *is* **positive semidefinite** *if*

$$\chi^* S \chi \geq 0 \quad \forall \chi \neq 0 \quad \text{and} \quad \chi \in \mathbb{C}^n.$$

$S \geq 0$ *indicates* S *is positive semidefinite. If* S *is positive definite, then* S *is also positive semidefinite.*

Appendix C

Linear Matrix Inequalities

This appendix contains a brief introduction to linear matrix inequalities. Please refer to Boyd et al. [21] for more details. A linear matrix inequalities or LMI is a matrix inequality of the form

$$S(\chi) = S_0 + \sum_{j=0}^{\iota} \chi_j S_j > 0$$

in which $\chi_1, \chi_2, \ldots, \chi_\iota \in \mathbb{R}$ are the variables and $S_j = S_j^T \in \mathbb{R}^{n\times n}$ are given. Multiple LMIs $S_1(\chi) > 0, \ldots, S_n(\chi) > 0$ can be expressed as a single LMI

$$\mathrm{diag}(S_1(\chi), \ldots, S_n(\chi)) > 0.$$

Convex quadratic inequalities can be converted to the LMI form using the Schur complements. The conversion is an if and only if conversion. Let $S_{11}(\chi) = S_{11}^T(\chi)$, $S_{22}(\chi) = S_{22}^T(\chi)$, and $S_{12}(\chi)$ depend affinely on χ. Then the LMI

$$\begin{bmatrix} S_{11}(\chi) & S_{12}(\chi) \\ S_{12}^T(\chi) & S_{22}(\chi) \end{bmatrix} > 0 \tag{C.1}$$

is equivalent to the matrix inequalities

$$S_{22}(\chi) > 0, \quad S_{11}(\chi) - S_{12}(\chi)S_{22}^{-1}(\chi)S_{12}^{T}(\chi) > 0$$

or, equivalently,

$$S_{11}(\chi) > 0, \quad S_{22}(\chi) - S_{12}^{T}(\chi)S_{11}^{-1}(\chi)S_{12}(\chi) > 0.$$

There are many problems in which the LMI variables are matrices. In such cases, the LMI is not written out explicitly in the form $S(\chi) > 0$, but the matrices are clarified as matrix variables.

Lemma C.0.7 *Suppose $S_{11}(x) = S_{11}(x)^T$, $S_{22}(x) = S_{22}(x)^T$, and $S_{12}(x)$ depends affinely on x. The convex nonlinear inequalities*

$$S_{22}(x) > 0, \quad S_{11}(x) - S_{12}(x)S_{22}(x)^{-1}S_{12}(x)^T > 0$$

are equivalent to the linear matrix inequality

$$\begin{bmatrix} S_{11}(x) & S_{12}(x) \\ S_{12}(x)^T & S_{22}(x) \end{bmatrix} > 0$$

Proof. See [151]. □

Bibliography

[1] K. J. Åström and L. Rundqwist. Integrator windup and how to avoid it. In *Proceedings of the 1989 American Control Conference*, pages 1693–1698, 1989.

[2] F. Alizadeh, J. Haeberly, and M. Overton. A new primal-dual interior- point method for semidefinite programming. In *Proceedings of the 5th SIAM Conf. On Applied Linear Algebra, Snowbird, UT*, 1994.

[3] J. C. Allwright and G. C. Papavasiliou. On linear programming and robust model predictive control using impulse-responses. *Sys. Cont. Let.*, 18:159–164, 1992.

[4] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduling H_∞ control of linear parameter-varying systems. *Automatica*, 31(9):1251–1261, 1995.

[5] T. Başar and P. Bernhard. *H^∞-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkauser, Boston, 1995.

[6] T. A. Badgwell. A robust model predicitive algorithm for nonlinear plants. Submitted for presentation at the *International Symposium on Advanced Control of Chemical Processes* (ADCHEM), Banff, Canada, June 1997.

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

[8] T. A. Badgwell. Robust stability conditions for SISO model predictive control algorithms. *Automatica*, 33(7):1357–1361, 1997.

[9] A. Bahnasawi, A. Alfuhaid, and M. Mahmound. Linear feedback approach to the stabilization of uncertain discrete-systems. *IEE Proceedings-D Control Theory and Applications*, 136(1):47–52, 1989.

[10] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, New York, 1974.

[11] A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 1384–1389, Tampa, Florida USA, 1998.

[12] A. Bemporad and E. Mosca. Fulfilling hard constrains in uncertain linear systems by reference managing. *Automatica*, 34(4):451–461, 1998.

[13] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[14] R. Berber. *Methods of Model Based Process Control (NATO Asi Series. Series E, Applied Sciences, Vol. 293)*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[15] L. T. Biegler. Efficient solution of dynamic optimization and NMPC problems. In *International Symposium on Nonlinear Model Predictive Control, Ascona, Switzerland*, 1998.

[16] F. Blanchini. Control synthesis for discrete time systems with control and state bounds in the presence of disturbances. *J. Optim. Theory Appl.*, 65:29–40, 1990.

[17] F. Blanchini. Constrained control for uncertain linear systems. *J. Optim. Theory Appl.*, 71:465–484, 1991.

[18] F. Blanchini and R. Pesenti. Min-Max control of uncertain multi-inventory systems with multiplicative uncertainties. *IEEE Trans. Auto. Cont.*, 46(6):955–960, June 2001.

[19] S. Boyd, C. Crusius, and A. Hansson. Control applications of nonlinear convex programming. *Journal of Process Control*, 8(5-6):313–324, 1998.

[20] S. Boyd and L. E. Ghaoui. Methods of centers for minimizing generalized eigenvalues. *Linear Algebra Applications*, 188:63–111, 1993.

[21] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in Control Theory*. SIAM, Philadelphia, 1994.

[22] P. J. Campo and M. Morari. Robust model predictive control. In *Proceedings of the 1987 American Control Conference*, pages 1021–1026, June 1987.

[23] Y. Cao and J. Lam. On simultaneous H_∞ control and strong H_∞ stabilization. *Automatica*, 36(6):859–865, 2000.

[24] A. Casavola, M. Giannelli, and E. Mosca. Min-max predictive control strategies for input-saturated polytopic uncertain systems. *Automatica*, 36(1):125–133, 2000.

[25] C. C. Chen and L. Shaw. On receding horizon control. *Automatica*, 16(3):349–352, 1982.

[26] H. Chen, C. W. Scherer, and F. Allgöwer. A game theoretic approach to nonlinear robust receding horizon control of constrained systems. In *Proceedings of American Control Conference, Albuquerque, NM*, pages 3073–3077, 1997.

[27] M. Chilali, P. Gahinet, and P. Apkarian. Robust pole placement in LMI regions. *IEEE Trans. Auto. Cont.*, 44(12):2257–2269, 1999.

[28] L. Chisci, J. Rossiter, and G. Zappa. Robust predictive control with restricted constraints to cope with estimation errors. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

[29] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restriced constraints. *Automatica*, 37(7):1019–1028, 2001.

[30] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Sys. Cont. Let.*, 29:121–129, 1996.

[31] P. Christofides. Robust control of nonlinear processes. In *Proceedings of the 1998 IFAC DYCOPS Symposium, Corfu, Greece*, pages 477–483, 1998.

[32] D. W. Clarke and R. Scattolini. Constrained receding–horizon predictive control. *IEE Proceedings–D*, 138(4):347–354, 1991.

[33] M. Corless and J. Manela. Control of uncertain discrete-time systems. In *Proceedings of the American Control Conference*, Seattle, Washington, June 1986.

[34] C. Cutler, A. Morshedi, and J. Haydel. An industrial perspective on advanced control. In *AIChE Meeting*, Washington DC, 1983.

[35] C. R. Cutler. Dynamic matrix control of imbalanced systems. *ISA Transactions*, 21:1–6, 1982.

[36] C. R. Cutler and B. L. Ramaker. Dynamic matrix control—a computer control algorithm. In *Proceedings of the Joint Automatic Control Conference*, 1980.

[37] E. Davison. The robust control of a servomechanism problem for linear time-invariant multivariable systems. *IEEE Trans. Auto. Cont.*, 21(1):25–34, 1976.

[38] E. Davison and I. Ferguson. The design of controllers for the multivariable robust servomechanism problem using parameter optimization methods. *IEEE Trans. Auto. Cont.*, 26(1):93–110, 1981.

[39] E. Davison and A. Goldenberg. Robust control of a general servomechanism problem: the servo compensator. *Automatica*, 11:461–471, 1975.

[40] E. J. Davison and H. W. Smith. Pole assignment in linear time-invariant multivariable systems with constant disturbances. *Automatica*, 7:489–498, 1971.

[41] G. De Nicolao, L. Magni, and R. Scattolini. Stability and robustness of nonlinear receding horizon control. In *International Symposium on Nonlinear Model Predictive Control, Ascona, Switzerland*, 1998.

[42] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Trans. Auto. Cont.*, 43(7):1030–1036, 1998.

[43] C. de Souza and U. Shaked. An LMI method for output-feedback H_∞ control design for systems with real parameter uncertainty. In *Proceedings of the 37th Conference on Decision and Control*, pages 1777–1779, 1998.

[44] S. E. Dreyfus. Some types of optimal control of stochastic systems. *SIAM J. Cont.*, 2(1):120–134, 1962.

[45] C. Edwards and I. Postlethwaite. Anti-windup and bumpless transfer schemes. *Automatica*, 34(2):199–210, 1998.

[46] J. Farison and S. Kolla. Asymptotic stability for a class of linear descrete-systems with bounded uncertainties - comment. *IEEE Trans. Auto. Cont.*, 35:382–384, 1990.

[47] J. Farison and S. Kolla. Stability robustness bounds for discrete-time linear regulator. *Int. J. Systems Science*, 22(11):2285–2298, 1991.

[48] B. A. Francis and W. M. Wonham. The internal model principle for linear multivariable regulators. *J. Appl. Math. Optim.*, 2:170–195, 1975.

[49] R. Freeman and P. Kokotovic. Robust integral control for a class of uncertain nonlinear systems. In *Proceedings of the 34th Conference on Decision and Control*, pages 2245–2250, 1995.

[50] B. Froisy and T. Matsko. Idcom-m application to the shell fundamental control problem. AIChE Annual Meeting, November 1990.

[51] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[52] H. Genceli and M. Nikolaou. Design of robust constrained model-predictive controllers with Volterra series. *AIChE J.*, 41(9):2098–2107, 1995.

[53] E. Gilbert, I. Kolmanovsky, and K. Tan. Nonlinear control of discrete-time linear systems with state and control constraints: a reference governor with global convergence properties. In *Proceedings of the 33rd Conference on Decision and Control*, pages 144–149, Lake Buena Vista, FL, December 1994.

[54] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Trans. Auto. Cont.*, 36(9):1008–1020, September 1991.

[55] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 1983.

[56] O. Grasselli, S. Longhi, and A.Tornambe. Robust tracking and performance for multivariable systems under physical parameter uncertainties. *Automatica*, 29(1):169–179, 1993.

[57] P. Grosdidier, B. Froisy, and M. Hammann. The IDCOM-M controller. In T. J. McAvoy, Y. Arkun, and E. Zafiriou, editors, *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, pages 31–36, Oxford, 1988. Pergamon Press.

[58] D. Henrion, D. Arzelier, D. Peaucelle, and M. Sebek. An LMI condition for robust stability of polynomial matrix polytopes. *Automatica*, 37(3):461–468, 2001.

[59] M. A. Henson and D. E. Seborg. *Nonlinear Process Control*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.

[60] C. Hollot and M. Arabacioglu. lth-step Lyapunov min-max controllers: stabilizing discrete-time systems under real parameter variations. In *Proceedings of the American Control Conference*, Minneapolis, Minnesota, June 1987.

[61] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Engineering*, pages 35–45, March 1960.

[62] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, J. Basic Engineering*, pages 95–108, March 1961.

[63] P. Kapasouris and M. Athans. Multivariable control systems with saturating actuators, antireset windup strategies. In *Proceedings of the 1985 American Control Conference*, pages 1579–1584, June 1985.

[64] D. Kassmann, T. Badgwell, and R. Hawkins. Robust steady-state target calculation for model predictive control. *AIChE J.*, 46(5):1007–1024, 2000.

[65] S. S. Keerthi and E. G. Gilbert. Computation of minimum-time feedback control laws for systems with state-control constraints. *IEEE Trans. Auto. Cont.*, 32:432–435, 1987.

[66] M. Khammash and L. Zou. Analysis of steady-state tracking errors in sampled-data systems with uncertainty. *Automatica*, 37(6):889–897, 2001.

[67] D. L. Kleinman. An easy way to stabilize a linear constant system. *IEEE Trans. Auto. Cont.*, 15(12):692, December 1970.

[68] I. Konstantopoulos and P. Antsaklis. Optimal design of robust controllers for uncertain discrete-time systems. *International Journal of Control*, 65(1):71–91, 1996.

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

[70] M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett. A unified framework for the study of anti-windup designs. *Automatica*, 30(12):1869–1883, 1994.

[71] B. Kouvaritakis, J. Rossiter, and A. Chang. Stable generalized predictive control: an algorithm with guaranteed stability. *IEE Proceedings–D*, 139(4):349–362, 1992.

[72] B. Kouvaritakis, J. Rossiter, and J. Schuurmans. Efficient robust predictive control. In *Proceedings of the American Control Conference*, San Diego, CA, June 1999.

[73] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Trans. Auto. Cont.*, 45(8):1545–1549, August 2000.

[74] H. Kwakernaak. Robust control and H_∞ optimization- tutorial paper. *Automatica*, 29(2):255–273, 1993.

[75] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. John Wiley and Sons, New York, 1972.

[76] W. H. Kwon and A. E. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans. Auto. Cont.*, 22(5):838–842, October 1977.

[77] W. H. Kwon and A. E. Pearson. On feedback stabilization of time-varying discrete linear systems. *IEEE Trans. Auto. Cont.*, 23(3):479–481, June 1978.

[78] J. Lee, K. Lee, and W. Kim. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36(5):641–657, 2000.

[79] J. Lee and M. Morari. Robust inferential control of multi-rate sampled-data systems. *Chemical Engineering Science*, 47(4):865–885, 1992.

[80] J. H. Lee and B. Cooley. Recent advances in model predictive control and other related areas. In J. C. Kantor, C. E. García, and B. Carnahan, editors, *Proceedings of Chemical Process Control – V*, pages 201–216. CACHE, AIChE, 1997.

[81] J. H. Lee and B. L. Cooley. Stable min-max control for state-space systems with bounded input matrix. In *Proceedings of American Control Conference, Albuquerque, NM*, pages 2945–2949, 1997.

[82] J. H. Lee and B. L. Cooley. Min-max predictive control techniques for a linear state-space system with a bounded set of input matrices. *Automatica*, 36(3):463–473, 2000.

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

[84] Y. Lee and B. Kouvaritakis. A linear programming approach to constrained robust predictive control. *IEEE Trans. Auto. Cont.*, 1999. Submitted.

[85] Y. I. Lee and B. Kouvaritakis. Robust receding horizon predictive control for systems with uncertain dynamics and input saturation. *Automatica*, 36(10):1497–1504, 2000.

[86] F. Lin, R. Brandt, and J. Sun. Robust control of nonlinear systems: compensating for uncertainty. *International Journal of Control*, 56(6):1453–1459, 1992.

[87] F. Lin and A. Olbrot. An LQR approach to robust control of linear systems with uncertain parameters. In *Proceedings of the 35th Conference on Decision and Control*, pages 4158–4163, 1996.

[88] F. Lin and W. Zhang. Robust control of nonlinear systems without matching condition. In *Proceedings of the 32nd Conference on Decision and Control*, pages 2572–2577, 1993.

[89] R. Liu. Convergent systems. *IEEE Trans. Auto. Cont.*, AC-13:384–391, 1968.

[90] Y. Lu and Y. Arkun. Polytope updating in quasi-min-max MPC algorithms. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

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

[92] M. Magana and S. Zak. Robust state feedback stabilization of discrete-time uncertain dynamical-systems. *IEEE Trans. Auto. Cont.*, 33:887–891, 1988.

[93] L. Magni, H. Nijmeijer, and A. J. van der Schaft. A receding-horizon approach to the nonlinear h_∞ control problem. *Automatica*, 37(3):429–435, 2001.

[94] M. S. Mahmoud. Robust H_∞ control of linear neutral systems. *Automatica*, 36(5):757–764, 2000.

[95] P. Makila. On three puzzles in robust control. *IEEE Trans. Auto. Cont.*, 45(3):552–556, March 2000.

[96] L. Marconi and A. Isidori. Mixed internal model-based and feedforward control for robust tracking in nonlinear systems. *Automatica*, 36(7):993–1000, 2000.

[97] D. Mayne and W. Schroeder. Robust time-optimal control of constrained linear systems. *Automatica*, 33(12):2103–2118, 1997.

[98] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Trans. Auto. Cont.*, 35(7):814–824, July 1990.

[99] D. Q. Mayne and H. Michalska. Receding horizon control of non-linear systems without differentiability of the optimal value function. *Sys. Cont. Let.*, 16:123–130, 1991.

[100] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[101] D. Megias, J. Serrano, and G. de Prada. Min-max constrained quasi-infinite horizon model predictive control using linear programming. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

[102] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Auto. Cont.*, 38(11):1623–1633, 1993.

[103] M. Morari and E. Zafiriou. *Robust Process Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[104] E. Mosca and J. Zhang. Stable redesign of predictive control. *Automatica*, 28(6):1229–1233, 1992.

[105] E. Mulder, M. Kothare, and M. Morari. Mutlivariable anti-windup controller synthesis using iterative linear matrix inequalities. In *Proceedings of European Control Conference, Karlsruhe, Germany*, 1999.

[106] E. Mulder, M. Kothare, and M. Morari. Multivariable anti-windup controller synthesis using linear matrix inequalities. *Automatica*, 37(9):1407–1416, 2001.

[107] R. Murray-Smith and T. Johansen. *Multiple Model Approaches to Nonlinear Modeling and Control*. Taylor & Frances, London, 1997.

[108] K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control, 2001. Submitted.

[109] K. R. Muske and J. B. Rawlings. Linear model predictive control of unstable processes. *J. Proc. Cont.*, 3(2):85–96, 1993.

[110] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE J.*, 39(2):262–287, 1993.

[111] Y. Nesterov and A. Nemirovsky. Interior-point polynomial methods in convex programming. In *Proceedings of the 5th SIAM Conf. On Applied Linear Algebra, Snowbird, UT*, 1994.

[112] B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York, 1994.

[113] G. Oliveira, W. Amaral, G. Favier, and G. Dumont. Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions. *Automatica*, 36(4):563–571, 2000.

[114] G. Pannocchia and J. B. Rawlings. The velocity algorithm LQR: a survey. Technical Report 2001–01, TWMCC, Department of Chemical Engineering, University of Wisconsin-Madison, May 2001.

[115] G. Pannocchia and J. B. Rawlings. Disturbance models for offset-free MPC control. Submitted for publication in *AIChE Journal*, March 2001.

[116] G. Pannocchia and D. Semino. Optimal modified models for robust predictive controllers. In *Proceedings of the 1999 IFAC World Congress, Beijing, China*, 1999.

[117] D. Prett and C. Garcia. The second shell process control workshop: solutions to the shell standard control problem. In *London, Butterworths*, 1989.

[118] D. M. Prett and C. E. Garcìa. *Fundamental Process Control*. Butterworths, Boston, 1988.

[119] J. A. Primbs. The analysis of optimization based controllers. *Automatica*, 37(6):933–938, 2001.

[120] J. S. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In *International Conference on Chemical Process Control, Lake Tahoe, California*, 1996.

[121] J. S. Qin and T. A. Badgwell. An overview of nonlinear model predictive control applications. In *International Symposium on Nonlinear Model Predictive Control, Ascona, Switzerland*, 1998.

[122] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In J. C. Kantor, C. E. García, and B. Carnahan, editors, *Proceedings of Chemical Process Control–V*, pages 232–256. CACHE, AIChE, 1997.

[123] L. Qiu and E. Davison. Performance limitations of non-minimum phase systems in the servomechanism problem. *Automatica*, 29(2):337–349, 1993.

[124] S. Ralhan and T. Badgwell. Robust disturbance rejection for FIR systems with bounded parameters. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

[125] M. A. Rami and X. Y. Zhou. Linear matrix inequalities, Riccati equations, and indefinite stochastic linear quadratic controls. *IEEE Trans. Auto. Cont.*, 45(6):1131–1143, June 2000.

[126] C. V. Rao, S. J. Wright, and J. B. Rawlings. On the application of interior point methods to model predictive control. *J. Optim. Theory Appl.*, 99:723–757, 1998.

[127] J. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20:38–52, 2000.

[128] J. B. Rawlings. Advanced process dynamics and control. Course notes, University of Wisconsin, Madison, Spring 1996.

[129] J. B. Rawlings, E. S. Meadows, and K. R. Muske. Nonlinear model predictive control: a tutorial and survey. In *ADCHEM '94 Proceedings, Kyoto, Japan*, pages 185–197, 1994.

[130] J. B. Rawlings and K. R. Muske. Stability of constrained receding horizon control. *IEEE Trans. Auto. Cont.*, 38(10):1512–1516, October 1993.

[131] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Algorithmic control of industrial processes. In *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, pages 1119–1167. North-Holland Publishing Company, 1978.

[132] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

[133] M. Rodrigues and D. Odloak. Robust stable MPC via dynamic output feedback. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

[134] J. Rossiter and B. Kouvaritakis. Youla parameter and robust predictive control with constraint handling. Workshop on Nonlinear Predictive Control, Ascona, Switzerland, 1998.

[135] W. J. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.

[136] W. Schmitendorf and B. Barmish. Robust asymptotic tracking for linear systems with unknown parameters. *Automatica*, 22(3):355–360, 1986.

[137] P. O. Scokaert and D. Q. Mayne. Min–max feedback model predictive control for constrained linear systems. *IEEE Trans. Auto. Cont.*, 43(8):1136–1142, August 1998.

[138] P. O. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Trans. Auto. Cont.*, 43(8):1163–1169, August 1998.

[139] P. O. M. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. In *Proceedings of the 13th IFAC World Congress*, San Francisco, California, July 1996.

[140] D. Semino and G. Pannocchia. Use of different kinds of linear models in predictive control of distillation columns. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

[141] U. Shaked, Y. Theodor, and I. Yaesh. Robust h_∞ control of discrete-time polytopic systems with application to the ecc95 benchmark problem. Technical Report No. EES98-1, Tel Aviv University, Israel, 1999.

[142] J. Shamma and M. Athans. Gain scheduling: potential hazards and possible remedies. In *Proceedings of the American Control Conference*, pages 516–521, Boston, MA, June 1991.

[143] F. G. Shinskey. *Feedback Controllers for the Process Industries*. McGraw-Hill, New York, 1994.

[144] A. Sideris and A. Tchernychev. Multiplier-based robust H_∞ design with time-varying uncertainties. *Automatica*, 36(4):579–586, 2000.

[145] S. Skogestad, M. Morari, and J. C. Doyle. Robust control of ill-conditioned plants: High-purity distillation. *IEEE Trans. Auto. Cont.*, 33(12):1092–1105, December 1988.

[146] O. Slupphaug and B. A. Foss. Bilinear matrix inequalities and robust stability of nonlinear multi-model MPC. In *Proceedings of the American Control Conference*, Philadelphia, PA, June 1998.

[147] O. Slupphaug and B. A. Foss. Quadratic stabilization of discrete-time uncertain nonlinear multi-model systems using piecewise affine state-feedback. *International Journal of Control*, 72(7/8):686–701, 1999.

[148] O. Slupphaug and B. A. Foss. Multi-model based uncertainty and robust control design. In *ADCHEM 2000 Proceedings, Pisa, Italy*, 2000.

[149] A. Teel and N. Kappor. The l_2 anti-windup problem: Its definition and solution. In *Proceedings of European Control Conference, Brussels, Belgium*, 1997.

[150] Y. A. Thomas. Linear quadratic optimal estimation and control with receding horizon. *Electron. Lett.*, 11:19–21, January 1975.

[151] J. VanAntwerp and R. Braatz. A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, 10(4):363–385, 2000.

[152] L. Vandenberghe and S. Boyd. A primal-dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming*, 69(1):205–236, 1995.

[153] H. Wang and B. K. Ghosh. A new roust control for a class of uncertain discrete-time systems. *IEEE Trans. Auto. Cont.*, 42(9):1252–1254, September 1997.

[154] S. Wang, L. Shieh, and J. Sunkel. Robust optimal pole-placement in a vertical strip and disturbance rejection. In *Proceedings of the 32nd Conference on Decision and Control*, pages 1134–1139, 1993.

[155] F. Wu and A. Packard. LQG control design for LPV systems. In *Proceedings of the American Control Conference*, pages 4440–4444, Seattle, Washington, June 1995.

[156] G. Zames. Feedback and optimal sensitivity: Model reference transformations, weighted seminorm, and approximate inverses. In *Proceedings of 17th Allerton Conference*, pages 744–752, 1979.

[157] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Trans. Auto. Cont.*, 26(2):301–320, April 1981.

[158] A. Zheng. Robust stability analysis of constrained model predictive control. *J. Proc. Cont.*, 9:271–278, 1999.

[159] Z. Q. Zheng and M. Morari. Robust stability of constrained model predictive control. In *Proceedings of the 1993 American Control Conference*, pages 379–383, 1993.

Vita

Yiyang (Jenny) Wang was born in Beijing, China to Fusheng Wang and Lily Yang. After living in Beijing for 10 years, she emigrated to the United States to join her parents in Houston, Texas where she attended Cummings Elementary School, Holub Middle School and Bellaire High School. In May, 1996, she received her B.S. in Chemical Engineering from Rice University. In August, 1996, she moved to Madison, Wisconsin to pursue her Ph.D. in Chemical Engineering.

Permanent Address: Yiyang Jenny Wang

5207 Laurel Street

Bellaire, TX 77401

This dissertation was prepared with LaTeX 2ε[1] by the author.

[1]This particular University of Wisconsin complient style was carved from The University of Texas at Austin styles as written by Dinesh Das (LaTeX 2ε), Khe–Sing The (LaTeX), and John Eaton (LaTeX). Knives and chisels wielded by John Campbell and Rock Matthews.