

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

UNIVERSITY OF CALIFORNIA

Los Angeles

Hybrid Predictive and Fault-tolerant Control of Chemical Process Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Chemical Engineering

by

Prashant Mhaskar

2005

UMI Number: 3208360

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 3208360

Copyright 2006 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

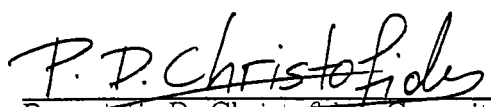
ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

The dissertation of Prashant Mhaskar is approved.


James F. Davis


Robert T. M'Closkey


Vasilios Manousiouthakis


Panagiotis D. Christofides, Committee Chair

University of California, Los Angeles

2005

Contents

1	Introduction	1
2	State Feedback Hybrid Predictive Control of Linear Systems	12
2.1	Introduction	12
2.2	Preliminaries	15
2.2.1	Bounded Lyapunov-based control	16
2.2.2	Model predictive control	18
2.3	Uniting bounded control and MPC: a hybrid control strategy	19
2.3.1	Problem formulation and overview of solution	20
2.3.2	Stability-based controller switching	21
2.3.3	Switching using advanced MPC formulations	29
2.4	Simulation example	33
3	Output Feedback Hybrid Predictive Control of Linear Systems	38
3.1	Introduction	38
3.2	Preliminaries	43
3.3	Model Predictive Control	45
3.4	Lyapunov-based Bounded Control	48
3.4.1	State feedback controller design	48

3.4.2	Stability properties under output feedback bounded control . . .	51
3.5	Hybrid Predictive Control	59
3.6	Simulation Studies	69
3.6.1	Illustrative example	69
3.6.2	Application to a chemical reactor example	78
3.7	Conclusions	83
4	Hybrid Predictive Control of Nonlinear Systems: Method And Ap- plications to Chemical Processes	85
4.1	Preliminaries	89
4.1.1	Bounded Lyapunov-based control	89
4.1.2	Model predictive control	91
4.2	The hybrid control strategy: uniting bounded control and MPC	93
4.2.1	Formulation of the switching problem	94
4.2.2	Stability-based controller switching	95
4.2.3	Enhancing closed-loop performance	103
4.3	Application to chemical process examples	109
4.3.1	Application to a chemical reactor example	109
4.3.2	Application to a continuous crystallizer example	114
4.4	Concluding remarks	116
5	Robust Hybrid Predictive Control of Nonlinear Systems	124
5.1	Introduction	124
5.2	Preliminaries	126
5.2.1	Bounded robust Lyapunov-based control	127
5.2.2	Model Predictive Control	129

5.3	Robust Hybrid Predictive Control Structure	130
5.3.1	Stability-based controller switching	130
5.3.2	Enhancing closed-loop performance	133
5.4	Application to a chemical reactor	136
6	Stabilization of Nonlinear Systems with State and Control Constraints Using Lyapunov-Based Predictive Control	142
6.1	Introduction	142
6.2	Preliminaries	145
6.2.1	Bounded Lyapunov-based control	146
6.3	Lyapunov-based Model predictive control	148
6.4	Handling soft state constraints via controller switching	155
6.5	Application to a chemical process example	160
6.6	Conclusions	162
7	Predictive Control of Switched Nonlinear Systems With Scheduled Mode Transitions	165
7.1	Introduction	165
7.2	Preliminaries	168
7.2.1	Bounded Lyapunov-based control	169
7.2.2	Lyapunov-based predictive control	174
7.3	Predictive control of switched nonlinear systems	178
7.4	Application to a chemical process example	187
7.5	Conclusions	189
8	Integrated Fault-Detection and Fault-Tolerant Control of Process Sys- tems	192

8.1	Introduction	192
8.2	Preliminaries	196
8.2.1	Process description	196
8.2.2	Motivating example	197
8.2.3	Bounded Lyapunov-based control	199
8.3	Integrated fault-detection and fault-tolerant control: state feedback case	201
8.3.1	State feedback fault-tolerant control	201
8.3.2	Simulation results	207
8.4	Integrated fault-detection and fault-tolerant control: output feedback case	211
8.4.1	Output feedback control	212
8.4.2	Integrating fault-detection and fault-tolerant output feedback control	215
8.4.3	Simulation results	223
8.5	Conclusions	227
9	Predictive Control of Linear Parabolic PDEs with State and Control Constraints	230
9.1	Introduction	230
9.2	Preliminaries	233
9.2.1	Parabolic PDEs	233
9.2.2	MPC formulation	236
9.3	Predictive control of infinite dimensional systems	238
9.3.1	Modal decomposition	238
9.3.2	MPC formulations: accounting for input and state constraints .	239
9.3.3	Higher-order MPC formulations	246

9.4	Simulation example	250
9.5	Conclusions	257
10	Predictive Control of Particle Size Distribution in Particulate Processes	259
10.1	Introduction	260
10.2	Predictive control of continuous particulate processes	263
10.2.1	A continuous crystallizer: modeling and dynamics	263
10.2.2	Hybrid predictive controller design	267
10.2.3	Application to control of PSD in a continuous crystallizer	272
10.3	Predictive control of batch particulate processes	277
10.3.1	A seeded batch crystallizer: modeling and dynamics	277
10.3.2	Predictive controller design: accounting for state and input constraints	280
10.3.3	Closed-loop simulation results	284
10.4	Conclusions	288
11	Predictive Control of Crystal Size Distribution in Protein Crystallization	289
11.1	Introduction	290
11.2	Modeling of nucleation and growth rate of tetragonal lysozyme crystal	294
11.2.1	Solubility of the tetragonal HEW lysozyme crystals	294
11.2.2	Modeling of the nucleation rate of the tetragonal HEW lysozyme crystals	295
11.2.3	Modeling of the growth rate of tetragonal HEW lysozyme crystals	297
11.3	Modeling of a batch crystallizer for protein crystallization	299

11.4 Predictive control of the batch protein crystallizer	302
11.4.1 Model reduction	303
11.4.2 State estimator design	304
11.4.3 Predictive controller formulation	306
11.5 Closed-loop simulation results	309
11.6 Conclusions	317
12 A Method for PID Controller Tuning Using Nonlinear Control Techniques	318
12.1 Introduction	318
12.2 Two-level PID Tuning Method	321
12.3 Application to a Chemical Reactor Example	333
12.4 Conclusions	340
13 Conclusions	343
Bibliography	345

List of Figures

2.1	A schematic representation of the hierarchical hybrid control structure merging the bounded and MPC controllers.	21
2.2	A schematic representation of the implementation of the stability-based controller switching proposed in Theorem 2.1.	24
2.3	A schematic representation of the implementation of switching when a stability-handling MPC formulation is used.	31
2.4	A comparison of the estimate of the null controllable region (X_{max}) with the stability region for the bounded controller estimated using Eqs.2.5-2.6.	34
2.5	Closed-loop state trajectory under MPC with $T = 1$ (solid line), under the switching scheme of Theorem 2.1 with $T = 1$ (dotted line), and under MPC with $T = 1.5$ (dashed line).	34
2.6	Input and closed-loop state profiles under MPC with $T = 1$ (solid line), under the switching scheme of Theorem 2.1 with $T = 1$ (dotted line), and under MPC with $T = 1.5$ (dashed line).	35
2.7	Closed-loop state trajectory starting close to the origin using MPC with $T = 1.2$ (solid line) and from a point farther away in state space using the switching scheme with a horizon length $T = 1.2$ (dotted line) and using only MPC with $T = 1.5$ (dashed line).	36

2.8	Input and closed-loop state profiles starting close to the origin using MPC with $T = 1.2$ (solid line) and from a point farther away in state space using the switching scheme with a horizon length $T = 1.2$ (dotted line) and using only MPC with $T = 1.5$ (dashed line).	37
3.1	Schematic representation of a hybrid control structure.	41
3.2	Schematic representation of the evolution of the trajectories of the closed-loop state and state estimate. Starting from within Ω_b , the observer gain ensures that, before the state of the closed-loop system reaches the boundary of Ω , the state estimate has converged close enough to the true value of the state, so that the norm of the estimation error is below the allowable error tolerated by the controller within Ω	57
3.3	Schematic representation of the hierarchical hybrid predictive control structure merging the bounded and model predictive controllers and a state observer.	64
3.4	Schematic representation of the evolution of the closed-loop system under the controller switching strategy proposed in Theorem 3.1. Note that MPC is not activated (even if feasible) before $\hat{x}$ enters Ω_s	65
3.5	States of the closed-loop system under the proposed hybrid predictive control strategy using two different horizon lengths of $T = 1.5$ (solid) and $T = 3.5$ (dashed) in MPC implementation, and an observer gain parameter of $\beta = 500$	71
3.6	Manipulated input profiles under the proposed hybrid predictive control strategy using two different horizon lengths of $T = 1.5$ (solid) and $T = 3.5$ (dashed) in MPC implementation, and an observer gain parameter $\beta = 500$	72

3.7	States of the closed-loop system under the proposed hybrid predictive control strategy using two different gain parameters of $\beta = 500$ (solid) and $\beta = 0.5$ (dashed) in observer implementation, and horizon length of $T = 1.5$ in the predictive controller.	73
3.8	Manipulated input profiles under the proposed hybrid predictive control strategy using two different gain parameters of $\beta = 500$ (solid) and $\beta = 0.5$ (dashed) in observer implementation, and horizon length of $T = 1.5$ in the predictive controller.	74
3.9	States of the closed-loop system under the proposed hybrid predictive control strategy in the presence of measurement noise and modeling errors. The solid profiles depict the case when the state observer is initially implemented using $\beta = 500$ and then switched to $\beta = 0.5$ at $t = 1.0$. The dotted profiles depict the case when the state observer is implemented using $\beta = 500$ for all times. The horizon length for the predictive controller is $T = 1.5$	76
3.10	Manipulated input profiles under the proposed hybrid predictive control strategy in the presence of measurement noise and modeling errors. The solid profiles depict the case when the state observer is initially implemented using $\beta = 500$ and then switched to $\beta = 0.5$ at $t = 1.0$. The dotted profiles depict the case when the state observer is implemented using $\beta = 500$ for all times. The horizon length for the model predictive controller is $T = 1.5$	77

3.11	Stability region estimate, Ω' , based on the linearization of the nonlinear model of Eq.3.28. The solid line depicts the closed-loop state trajectory starting from an initial condition for which MPC (designed based on the linear model with $T = 2$) is feasible and stabilizes the nonlinear closed-loop system, while the dotted line depicts the state trajectory starting from an initial condition for which MPC is infeasible, and the hybrid predictive control strategy is implemented instead to enforce closed-loop stability.	80
3.12	Closed-loop reactant concentration (top) and reactor temperature (bottom) starting from an initial condition for which MPC (designed based on the linear model with $T = 2$) is feasible and stabilizes the nonlinear closed-loop system (solid), and starting from an initial condition for which MPC is infeasible, and the hybrid predictive control strategy is implemented instead to enforce closed-loop stability (dotted).	81
3.13	Rate of heat input (top) and inlet reactant concentration in deviation variable form (bottom) when MPC is implemented (solid) and when the hybrid predictive control strategy is implemented (dotted).	82
4.1	Schematic representation of the hybrid control structure merging MPC and a family of fall-back bounded controllers with their stability regions.	97
4.2	Schematic representation illustrating the main idea of the controller switching scheme proposed in Theorems 1 and 2.	105
4.3	Implementation of the proposed hybrid control structure: Closed-loop state trajectory under MPC with $T = 0.25$ (solid trajectory), under the switched MPC/bounded controller(3) with $T = 0.25$ (dashed trajectory), under MPC with $T = 0.5$ (dotted trajectory), and under the switched bounded controller(2)/MPC with $T = 0.5$ (dash-dotted line). .	118

4.4	Closed-loop reactant concentration profile (a), reactor temperature profile (b) and coolant temperature profile (c) under MPC with $T = 0.25$ (solid trajectory), under the switched MPC/bounded controller(3) with $T = 0.25$ (dashed trajectory), under MPC with $T = 0.5$ (dotted trajectory), and under the switched bounded controller(2)/MPC with $T = 0.5$ (dash-dotted line).	119
4.5	Closed-loop state trajectory under the switching rules of Theorem 2 with $T = 0.1$ for MPC.	120
4.6	Closed-loop reactant concentration profile (a), reactor temperature profile (b) and coolant temperature profile (c) under the switching scheme of Theorem 2 with $T = 0.1$ for MPC.	121
4.7	Implementation of the proposed hybrid control structure to a continuous crystallizer: two-dimensional projections of the stability region for the 10 distinct combinations of the process states.	122
4.8	Closed-loop profiles of the dimensionless crystallizer moments (a)-(d), the solute concentration in the crystallizer (e) and the manipulated input (f) under MPC with stability constraints (solid line), under MPC without terminal constraints (dashed line), and using the switching scheme of Theorem 1 (dotted line).	123
5.1	Examples of the evolution of closed-loop state trajectory under the switching scheme of Theorem 5.2.	137
5.2	Closed-loop state trajectory: implementation of the robust hybrid predictive controller of Theorem 5.1 using a nominal MPC formulation without stability constraints (solid line) and an MPC formulation with stability constraints, from two different initial conditions (dashed and dotted lines).	140

5.3	Closed-loop state trajectory: implementation of the robust hybrid predictive controller of Theorem 5.2, with $l = 2$, $N = 1$, using a nominal MPC formulation without stability constraints (solid line).	140
5.4	Closed-loop state (top) and input (bottom) profiles: implementation of the robust hybrid predictive controller of Theorem 5.1 using a nominal MPC formulation without stability constraints (solid lines), an MPC formulation with stability constraints, from two different initial conditions (dashed and dotted lines) and the robust hybrid predictive controller of Theorem 5.2, with $l = 2$, $N = 1$, using a nominal MPC formulation without stability constraints (dash-dotted lines).	141
6.1	A schematic representing the input (Φ_u, Ω_u) and state and input $(\Phi_{x,u}, \Omega_{x,u})$ constrained stability regions, together with the set (X) describing the state constraints.	152
6.2	Closed-loop state trajectory under the predictive controller of Proposition 6.2 (solid line), under the bounded controller (dashed line) and under the predictive controller of Theorem 6.1 (dotted line).	162
6.3	Closed-loop state (top) and input (bottom) profiles under the predictive controller of Proposition 6.2 (solid lines), under the bounded controller (dashed lines) and under the predictive controller of Theorem 6.1 (dotted lines).	163
7.1	A schematic representation of the stability region and the evolution of the closed-loop state trajectory under discrete implementation of the bounded controller.	172
7.2	A schematic representation of the predictive control structure comprised of the predictive and bounded controllers for the constituent modes, together with transition constraints.	183

7.3	Schematic representation of the closed-loop trajectory of a switched system under the implementation of the predictive control algorithm of Theorem 7.1 (solid line) and using the relaxed constraints of Remark 7.9 (dashed line).	184
7.4	Closed-loop state trajectory when the reactor is operated in mode 1 for all times under the stabilizing MPC formulation of Eqs.7.11-7.15 (solid line), when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller design of Eqs.7.11-7.15 (dashed line), and when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller of Theorem 7.1 (dashed-dotted line).	190
7.5	Closed-loop state (top plots) and manipulated input (bottom plots) profiles: when the reactor is operated in mode 1 for all times under the stabilizing MPC formulation of Eqs.7.11-7.15 (solid line), when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller design of Eqs.7.11-7.15 (dashed line), and when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller of Theorem 7.1 (dashed-dotted line).	191
8.1	A schematic of the CSTR showing the three candidate control configurations.	199
8.2	Integrated fault-detection and fault-tolerant control design: state feedback case.	204
8.3	Evolution of the closed-loop state profiles under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid line) and under arbitrary switching (dashed line).	209

8.4	Evolution of the closed-loop (a) temperature and (b) concentration under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).	210
8.5	Evolution of the closed-loop residual under the fault-detection filter for (a) control configuration 1 and (b) control configurations 2 and 3 under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).	210
8.6	Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).	211
8.7	Integrated fault-detection and fault-tolerant control design under output feedback.	221
8.8	Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under control configuration 1 when the fault detection filter is initialized at $t = 0.005$ minutes.	224
8.9	Evolution of (a) the residual and (b) the manipulated input profile for the first control configuration when the fault detection filter is initialized at $t = 0.005$ minutes.	225

8.10	Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2.	226
8.11	Evolution of the closed-loop state trajectory under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2, using an appropriate fault-detection filter (solid line) and in the absence of a fault-detection filter (dashed line).	226
8.12	Evolution of the residual for (a) the first control configuration and (b) the second control configuration.	227
8.13	Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2 in the absence of a fault-detection filter.	228
8.14	Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2 in the presence (solid lines) and absence (dashed lines) of a fault-detection filter. . . .	228
9.1	Closed-loop state profile under the MPC formulation of Eqs.9.55-9.56 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 3.15 \sin(3z) + 3.15 \sin(4z)$. . .	252

9.2	Closed-loop state profile under the MPC formulation of Eqs.9.57-9.58 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 0.95 \sin(3z)$	253
9.3	$R(z_c, t) = \int_0^\pi r(z) \bar{x}(z, t) dz$ under the MPC formulation of Eqs.9.55-9.56 (solid line) and under the MPC formulation of Eqs.9.57-9.58 (dashed line). The dotted line represents the lower state constraint.	253
9.4	Manipulated input profiles for the first and second control actuators applied at $z_{a_1} = \pi/3$ and $z_{a_2} = 2\pi/3$ under the MPC formulation of Eqs.9.55-9.56 (solid line) and under the MPC formulation of Eqs.9.57-9.58 (dashed line); note that the dashed and solid line coincide because of the same initial conditions of the a_s -states.	254
9.5	Closed-loop state profile under the MPC formulation of Eqs.9.59-9.60 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 2.8 \sin(3z) + 2.85 \sin(4z)$	255
9.6	Closed-loop state profile under the MPC formulation of Eqs.9.61-9.62 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 1.45 \sin(3z) + 1.5 \sin(4z)$	256
9.7	$R(z_c, t) = \int_0^\pi r(z) \bar{x}(z, t) dz$ under the MPC formulation of Eqs.9.59-9.60, (dashed line) and under the MPC formulation of Eqs.9.61-9.62 (solid line).	257
9.8	Manipulated input profiles for the first and second control actuators applied at $z_{a_1} = \pi/3$ and $z_{a_1} = 2\pi/3$ under the MPC formulation of Eqs.9.59-9.60 (dashed line) and under the MPC formulation of Eqs.9.61-9.62 (solid line).	257
9.9	$R(z_c, t) = \int_0^\pi r(z) \bar{x}(z, t) dz$ under the MPC formulation of Eqs.9.55-9.56 with $\bar{x}(z, 0) = 0.04 \sin(z) + 0.0005 \sin(2z) + 0.07 \sin(3z)$ and state constraints $-0.035 \leq R(z_c, t) \leq 2$ with $[u_{max}, u_{min}] = [-10, 10]$	258
10.1	Open-loop profiles of (a) crystal concentration and (b) solute concentration obtained from the distributed parameter model.	265
10.2	Closed-loop system under hybrid predictive control.	271

10.3	Two dimensional projections of the stability region for the 10 distinct combinations of the states of the reduced order system of Eq.10.6. . . .	274
10.4	Continuous crystallizer example: closed-loop profiles of the dimensionless crystallizer moments (a)-(d), the solute concentration in the crystallizer (e) and the manipulated input (f) under MPC with stability constraints (solid lines), under the bounded controller (dotted lines), and using the hybrid predictive controller (dash-dotted lines).	275
10.5	Closed-loop profiles of the PSD: (a) under MPC without terminal constraints; (b) under the hybrid predictive controller.	276
10.6	Simulation results for the linear cooling strategy: (a) reactor temperature and concentration profiles (dashed and dotted lines represent the upper and lower constraints on the concentration respectively); (b) the evolution of crystal size distribution.	279
10.7	Simulation results for the linear cooling strategy: (a) the zeroth and (b) third moments of the PSD of the crystals formed by nucleation (dashed lines) and the those growing from the seeds (solid lines).	281
10.8	Closed-loop simulation results for matched models: (a) Jacket temperature and concentration profiles (dashed and dotted lines represent the upper and lower constraints on the concentration, respectively); (b) the evolution of crystal size distribution.	284
10.9	Closed-loop simulation results for mismatched models: (a) Jacket temperature and concentration profiles (dashed and dotted lines represent the upper and lower constraints on the concentration, respectively); (b) the evolution of crystal size distribution.	285
10.10	Comparison of the final crystal size distributions under 4 different control strategies.	286

11.1	Solubility data of tetragonal lysozyme crystal: dotted and dashed lines show the data in [27], Δ shows data in [194], $\circ$ shows data in [199], and the solid line shows the empirical model of Eq.11.1).	295
11.2	Comparison of the experimental results on the nucleation rate of tetragonal HEW lysozyme crystal: $\circ$ show the data in [82], Δ show the data in [23], and the empirical model of Eq.11.2 is shown by the solid line.	297
11.3	Structure of a tetragonal HEW lysozyme crystal.	298
11.4	Comparison of the experimental results on the growth rate of tetragonal HEW lysozyme crystal: $\diamond$ shows the data in [172], Δ shows data in [61], and $\square$ and $\circ$ show the data in [119].	299
11.5	Open-loop simulation results of the solute concentration (C), the supersaturation (σ) and the CSD at a constant crystallizer temperature ($T = 15^\circ\text{C}$).	301
11.6	Comparison between the true value and the estimates of C (the solute concentration) and $\mu_i, i = 0, 1, \dots, 4$.	306
11.7	(a) Jacket temperature and (b) supersaturation profiles under 1) state feedback control: dotted (implementation of the initially computed manipulated input trajectory) and dash-dotted (closed-loop state feedback control) lines, and 2) output feedback control; dashed (implementation of the initially computed manipulated input trajectory) and solid (closed-loop output feedback control) lines.	310
11.8	Comparison of the simulation results for (a) jacket temperature and (b) solute concentration under four different control strategies.	311
11.9	Comparison of the simulation results for (a) supersaturation and (b) volume density under four different control strategies.	311

11.10	Evolution of crystal size distribution under four different control strategies: (a) Predictive control with matched model, (b) Predictive control with mismatched model, (c) Predictive control with constraint on fines, (d) constant temperature control, (e) constant supersaturation control.	312
12.1	Schematic representation of the implementation of the proposed two-level, optimization-based tuning method.	323
12.2	Closed-loop output (a) and manipulated input (b) profile under a linearizing controller (solid line) and a PI controller tuned using the proposed method (dashed line).	337
12.3	Closed-loop output (a) and manipulated input (b) profile using IMC tuning rules for PI controller (solid line), using Ziegler-Nichols tuning rules (dotted line) and the proposed method (dashed line).	338
12.4	Closed-loop output (a) and manipulated input (b) profile under a linearizing controller (solid line) and a PID controller tuned using the proposed method (dashed line).	341
12.5	Closed-loop output (a) and manipulated input (b) profile using IMC tuning rules I (dotted line), IMC tuning rules II (solid line), Ziegler-Nichols tuning rules (dash-dotted line) and the proposed method (dashed line).	342

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Panagiotis Christofides, for his unwavering support, encouragement and guidance throughout the course of the thesis.

I would like to thank Professor Jim Davis, Professor Robert M'Closkey and Professor Vasilios Manousiouthakis for agreeing to participate in my examination committee and for their constructive comments and encouragement.

I would like to thank my family for their continuous support, affection and patience.

Finally, I would like to thank my wife Neerja Pophli, who really made this happen.

VITA

- 1977 Born on January 12, Varanasi, India
- 1999 B. Tech., Department of Chemical Engineering
IIT Mumbai
Mumbai, India
- 2001 M. S., Department of Chemical Engineering
Louisiana State University
Baton Rouge, Louisiana
- 2001-05 Graduate Student
Department of Chemical Engineering
University of California, Los Angeles, CA

PUBLICATIONS

Dubljevic, S., N. H. El-Farra, P. Mhaskar and P. D. Christofides, "Predictive Control of Parabolic PDEs with State and Control Constraints," *IEEE Trans. Automat. Contr.*, provisionally accepted, 2005.

Dubljevic, S., P. Mhaskar, N. H. El-Farra and P. D. Christofides, "Predictive Control of Transport-Reaction Processes," *Comp. & Chem. Eng.*, in press, 2005.

El-Farra, N. H., P. Mhaskar and P. D. Christofides, "Hybrid Predictive Control of Nonlinear Systems: Method and Applications to Chemical Processes," *Int. J. Rob. & Non. Contr.*, **14**, 199-225, 2004.

El-Farra, N. H., P. Mhaskar and P. D. Christofides, "Uniting Bounded Control and MPC for Stabilization of Constrained Linear Systems," *Automatica*, **40**, 101-110, 2004.

El-Farra, N. H., P. Mhaskar and P. D. Christofides, "Output Feedback Control of Switched Nonlinear Systems Using Multiple Lyapunov Functions," *Sys. & Contr. Lett.*, in press, 2005.

Mhaskar, P., A. Gani, N. H. El-Farra, P. D. Christofides and J. F. Davis, "Integrated

Fault-Detection and Fault-Tolerant Control for Process Systems,” *AIChE J.*, provisionally accepted, 2005.

Mhaskar, P., A. Gani and P. D. Christofides, “Fault-tolerant Control of Chemical Processes: Performance-based Reconfiguration and Robustness,” *Int. J. Rob. & Non. Contr.*, submitted, 2005.

Mhaskar, P., M. A. Hjortso and M. A. Henson, “Cell Population Modeling and Parameter Estimation for Continuous Cultures of *Saccharomyces Cerevisiae*,” *Biotechnology Progress*, **18**, 1010-1026, 2002.

Mhaskar, P., N. H. El-Farra and P. D. Christofides, “Hybrid Predictive Control of Process Systems,” *AIChE J.*, **50**, 1242-1259, 2004.

Mhaskar, P., N. H. El-Farra and P. D. Christofides, “Robust Hybrid Predictive Control of Nonlinear Systems,” *Automatica*, **41**, 209-217, 2005.

Mhaskar, P., N. H. El-Farra and P. D. Christofides, “A Method for PID Controller Tuning Using Nonlinear Control Techniques,” *AIChE J.*, in press, 2005.

Mhaskar, P., N. H. El-Farra and P. D. Christofides, “Stabilization of Nonlinear Systems with State and Control Constraints Using Lyapunov-Based Predictive Control,” *Sys. & Contr. Lett.*, in press, 2005.

Mhaskar, P., N. H. El-Farra and P. D. Christofides, “Predictive Control of Switched Nonlinear Systems with Scheduled Mode Transitions,” *IEEE Trans. Automat. Contr.*, in press as a regular paper, 2005.

Shi, D., N. H. El-Farra, M. Li, P. Mhaskar and P. D. Christofides, “Predictive Control of Particle Size Distribution in Particulate Processes,” *Chem. Eng. Sci.*, in press, 2005.

Shi, D., P. Mhaskar, N. H. El-Farra and P. D. Christofides, “Predictive Control of Crystal Size Distribution in Protein Crystallization,” *Nanotechnology*, **16**, S562-S574, 2005.

Venkataraman, C., A. Mehra and P. Mhaskar, “Mechanisms of sulphate aerosol pro-

duction in clouds: effect of cloud characteristics and season in the Indian region,”
Tellus, **53b**, 260-272, 2001.

ABSTRACT OF THE DISSERTATION

Hybrid Predictive and Fault-tolerant Control of Chemical Process Systems

by

Prashant Mhaskar

Doctor of Philosophy in Chemical Engineering

University of California, Los Angeles, 2005

Professor Panagiotis D. Christofides, Chair

Modern-day chemical plants involve a complex arrangement of processing units connected, in series and/or in parallel, and highly integrated with respect to material and energy flows through recycle streams. Moreover, chemical processes need to satisfy high quality product specifications and increasingly stringent safety and environmental regulations. The more stringent operating conditions, induced by performance, safety or environmental considerations, have placed new constraints on the operating flexibility of the process and dictates the need for continuous monitoring of the operation of a chemical plant and effective external intervention (automatic control) to guarantee the satisfaction of the process objectives. Central to the design of effective process control and monitoring systems are the issues of strong nonlinear dynamics, plant-model mismatch, external disturbances and actuator constraints. In addition, increased automation tends to increase the vulnerability of the plant to faults (e.g., defects/malfunctions in process equipment, sensors and actuators, failures in the controllers or in the control loops) potentially causing a host of economic, environmental, and safety problems that can seriously degrade the operating efficiency of the plant.

Motivated by the above considerations, this doctoral thesis develops methods and

strategies for the design of advanced process control systems that account for the complex process dynamics, including issues such as constraints and uncertainty, and ensure an efficient and timely response to enhance fault recovery, prevent faults from propagating or developing into total failures, and reduce the risk of safety hazards. To this end, we first develop a hybrid predictive control framework that seamlessly unites two different control approaches in a way that reconciles their respective stability and optimality properties and address several practical issues including constraints, unavailability of measurements and uncertainty, for nonlinear systems via explicitly switching between the predictive and Lyapunov-based control approaches. Then, we integrate the two approaches at a design level to develop Lyapunov-based predictive control approaches that allow for an explicit computation of the stability region in the presence of state and control constraints and use the developed Lyapunov-based predictive controllers for the task of ensuring the satisfaction of schedules for chemical processes that switch between various modes of operations. Furthermore, the developed control approaches, and the stability region information that they provide are used in the development of fault-detection and fault-tolerant control strategies that account for the complex dynamics of chemical processes and allow for early detection and reconfiguration to preserve process stability in the event of occurrence of a fault. Finally, the insight gained in the design of predictive control strategies to appropriately express and account for the presence of state and control constraints is used toward developing predictive control strategies for the control of distributed and particulate processes that are operated in continuous and batch mode. Throughout the thesis, numerous examples are used to illustrate the application and demonstrate the advantages of the new control methods.

Chapter 1

Introduction

Modern-day chemical plants involve a complex arrangement of processing units connected, in series and/or in parallel, and highly integrated with respect to material and energy flows through recycle streams. Moreover, chemical processes need to satisfy high quality product specifications and increasingly stringent safety and environmental regulations. These more stringent operating conditions have placed new constraints on the operating flexibility of the process and made the performance requirements for process plants increasingly difficult to satisfy. The increased emphasis placed on safe and efficient plant operation dictates the need for continuous monitoring of the operation of a chemical plant and effective external intervention (automatic control) to guarantee the satisfaction of the process objectives. The design of effective process control and monitoring systems, however, requires that a host of fundamental and practical problems, which transcend the boundaries of specific applications, be adequately taken into account at the stage of controller design. Central to these issues are the problems of strong nonlinear dynamics, plant-model mismatch, external disturbances and actuator constraints. In addition, increased automation tends to increase the vulnerability of the plant to faults (e.g., defects/malfunctions in process equipment, sensors and actuators, failures in the controllers or in the control loops) potentially causing a host of economic, environmental, and safety problems that can seriously degrade the operating

efficiency of the plant.

Model Predictive Control (MPC), also known as receding horizon control, is one control method for handling constraints (both on manipulated inputs and state variables) within an optimal control setting [186]. In MPC, the control action is obtained by solving repeatedly, on-line, a finite-horizon constrained open-loop optimal control problem. The popularity of this approach stems largely from its ability to handle, among other issues, multi-variable interactions, constraints on controls and states, and optimization requirements. Its success in many commercial applications is well-documented in the literature (see, for example, [84, 174]). Numerous research studies have investigated the stability properties of model predictive controllers and led to a plethora of MPC formulations that focus on closed-loop stability (see, for example, [111, 87, 182, 48] and the review paper [141]).

Many systems of practical interest exhibit highly nonlinear behavior that must be accounted for when designing the controller. In the literature, several nonlinear model predictive control (NMPC) schemes have been developed (e.g., see [140, 149, 31, 115, 217, 132, 219]) that focus on the issues of stability, constraint satisfaction and performance optimization for nonlinear systems. A common idea of these approaches is to enforce stability by means of suitable penalties and constraints on the state at the end of the finite optimization horizon. Because of the system nonlinearities, however, the resulting optimization problem is non-convex and, therefore, much harder to solve, even if the cost functional and constraints are convex. The computational burden is more pronounced for NMPC algorithms that employ terminal stability constraints (e.g., equality constraints) whose enforcement requires intensive computations that typically cannot be performed within a limited time-window. The problem of analysis and design of predictive controllers for uncertain linear systems has been extensively investigated (see [19, 141] for surveys of results in this area). For uncertain nonlinear systems, the problem of robust MPC design continues to be an area of ongoing research

(see, for example, [149, 131]). While min-max formulations provide a natural setting within which to address this problem, computational problems with these approaches are well known, and stem in part from the nonlinearity of the model which typically makes the optimization problem non-convex and in part from performing the min-max optimization over the non-convex problem. In general, the stability guarantee in MPC formulations (with or without stability conditions, and with or without robustness considerations) for nonlinear uncertain systems is contingent upon the assumption of initial feasibility, and the set of initial conditions starting from where feasibility and stability is guaranteed a priori (i.e., without online computation) is not explicitly characterized. To this end, development of control strategies that allow for the implementation of predictive control strategies in a way that provides the predictive controllers with a safety net (via an explicit computation of the stability region) is of significant value to the operation of chemical processes.

Other than exhibiting complex behavior such as nonlinearity, constraints and uncertainty, the operation of chemical processes also often exhibits hybrid behavior, that may arise due to a variety of reasons including the need to follow a desired schedule, or in the context of fault-tolerant control. When a process described by linear dynamics is required to follow a prescribed schedule, the switched system can be transformed into a mixed logical dynamical system, and a mixed-integer linear program can be solved to come up with an optimal switching sequence and switching times [1, 55]. For nonlinear systems, one can, in principle, set up the mixed integer nonlinear programming problem, where the decision variables include the control action together with the switching schedule. The presence of nonlinear dynamics, that typically make the optimization problem non-convex, together with the discrete decision variables results in high computational complexity, and the computation time requirements render it unsuitable for the purpose of real-time control. In many systems of practical interest, the switched system is required to follow a prescribed switching schedule, where the switching times are no longer decision variables, but are prescribed via an operating

schedule, and it is important to develop control approaches that guides the process trajectory in a way that preserves closed-loop stability.

The hybrid nature of processes also arises in the context of fault-tolerant control, which has been an active area of research for the past ten years (see, e.g., [163, 245] for research studies in the context of aerospace engineering applications) and is based on the underlying assumption of the availability of more control configurations than is required. Under this assumption, the reliable control approach dictates use of all the control loops at the same time so that failure of one control loop does not lead to the failure of the entire control structure (e.g., [237]). The use of only as many control loops as is required at a time, is often motivated by economic considerations (to save on unnecessary control action), and in this case, fault-tolerant control can be achieved through control-loop reconfiguration. Recently, fault-tolerant control has gained increasing attention in the context of chemical process control; however, the available results are mostly based on the assumption of a linear process description (e.g., [17, 233]), and do not account for complexities such as control constraints or the unavailability of state measurements.

The occurrence of faults in chemical processes and subsequent switching to fall-back control configurations naturally leads to the superposition of discrete events on the underlying continuous process dynamics thereby making a hybrid systems framework a natural setting for the analysis and design of fault-tolerant control structures. Proper coordination of the switching between multiple (or redundant) actuator/sensor configurations provides a means for fault-tolerant control. However, at this stage, despite the large and growing body of research work on a diverse array of hybrid system problems (e.g., [90, 85, 94, 50, 19, 70]), the use of a hybrid system framework for the study of fault-tolerant control problems for nonlinear systems subject to constraints has received limited attention. In [71], a hybrid systems approach to fault-tolerant control was employed where, under the assumption of full state measurements and knowledge of the

fault, stability region-based reconfiguration is implemented to achieve fault-tolerant control.

Existing results on the design of fault-detection filters include those that use past plant-data and those that use fundamental process models for the purpose of fault-detection filter design. Statistical and pattern recognition techniques for data analysis and interpretation (e.g., [116, 191, 160, 59, 157, 58, 47, 220, 5, 243]) use past plant-data to construct indicators that identify deviations from normal operation to detect faults. The problem of using fundamental process models for the purpose of detecting faults has been studied extensively in the context of linear systems [139, 78, 79, 144]; and more recently, some existential results in the context of nonlinear systems have been derived [198, 53].

In summary, a close examination of the existing literature indicates the lack of general and practical methods for the design controllers that account for complex process dynamics including nonlinearity, constraints and uncertainty, together with incorporating performance considerations, and provide an explicit characterization of their stability region, and their use within an integrated approach in the context of scheduling of process operations and in fault-detection and fault-tolerant control structures for chemical plants.

Motivated by the above, the objectives of the present thesis are:

- To develop a systematic framework for control of nonlinear chemical processes subject to uncertainty and constraints,
- To develop predictive control designs that account for the hybrid nature of a nonlinear processes and guide the system trajectory in a way that ensures stabilization in the presence of constraints,
- To use the developed control designs within an integrated fault-detection and fault-tolerant control designs for nonlinear processes, and

- To apply the developed control methods to chemical processes.

The rest of the thesis is organized as follows: Chapters 2 and 3 consider linear systems subject to constraints and propose a hybrid predictive control structure that seamlessly unites MPC and bounded control for the state and output feedback stabilization of linear systems with input constraints. The proposed strategy is based upon the idea of embedding the implementation of MPC within the output feedback stability region of the bounded controller, and using this controller as a “fall-back” in the event that MPC is unable to achieve closed-loop stability (due, for example, to improper tuning of MPC parameters). Due to the absence of complete state measurements, a state observer is constructed to provide the controllers, as well as the supervisor, with appropriate state estimates. The observer is tuned in a way so as to guarantee closed-loop stability for all initial conditions within the bounded controller’s output feedback stability region (which can be chosen arbitrarily close in size to its state feedback counterpart, provided that the observer gain is sufficiently large). Switching laws, that monitor the evolution of the state estimates, are derived to orchestrate the transition between the two controllers in a way that reconciles their respective stability and optimality properties and safeguards closed-loop stability in the event of MPC infeasibility or instability. In addition to the set of switching rules being different from that under state feedback, an important characteristic of the hybrid predictive control strategy under output feedback is the inherent coupling, brought about by the lack of full state measurements, between the tasks of controller design, characterization of the stability region and supervisory switching logic design, on one hand, and the task of observer design, on the other. Having developed the hybrid predictive controllers for linear systems, and linear systems subject to unavailability of measurements, in chapters 4–5 we consider nonlinear systems, and nonlinear systems subject to uncertainty. We present hybrid predictive control structures, that provide predictive control formulations with a safety net by using a combination of several fall-back controllers and switching rules

that allow for the enhanced implementation of predictive controllers thereby allowing the possibility of exploiting their optimality properties.

In chapter 6, we use the insight developed in the design of the hybrid predictive controllers in developing a Lyapunov-based model predictive control design for the stabilization of nonlinear systems with both state and input constraints. The design of the Lyapunov-based MPC uses a bounded controller, with its associated region of stability, as an auxiliary controller (instead of switching between the two controllers, as is done in chapters 2-5) that is used to characterize the stability properties of the Lyapunov-based MPC. The proposed Lyapunov-based MPC is shown to possess an explicitly characterized set of initial conditions, starting from where it is guaranteed to be feasible, and hence stabilizing, while enforcing the state and input constraints at all times. For the case when the state constraints are soft, we propose a switched predictive control strategy that reduces the time for which state constraints are violated, driving the states into the state and input constraints feasibility region of the Lyapunov-based predictive controller. We demonstrate the application of the Lyapunov-based predictive controller designs through a chemical process example.

In chapter 7, we use the Lyapunov-based predictive controllers developed in chapter 6 to propose a predictive control framework for the constrained stabilization of switched nonlinear systems that transit between their constituent modes at prescribed switching times. The main idea is to design a Lyapunov-based predictive controller for each constituent mode in which the switched system operates, and incorporate constraints in the predictive controller design which upon satisfaction ensure that the prescribed transitions between the modes occur in a way that guarantees stability of the switched closed-loop system. This is achieved as follows: for each constituent mode, a Lyapunov-based model predictive controller (MPC) is designed, and an analytic bounded controller, using the same Lyapunov function, is used to explicitly characterize a set of initial conditions for which the MPC, irrespective of the controller

parameters, is guaranteed to be feasible, and hence stabilizing. Then, constraints are incorporated in the MPC design which, upon satisfaction, ensure that: (1) the state of the closed-loop system, at the time of the transition, resides in the stability region of the mode that the system is switched into, and (2) the Lyapunov function for each mode is non-increasing wherever the mode is re-activated, thereby guaranteeing stability. The proposed control method is demonstrated through application to a chemical process example.

In chapter 8 we consider the problem of implementing fault-tolerant control to nonlinear processes with input constraints subject to control actuator failures, and present and demonstrate an approach predicated upon the idea of integrating fault-detection, feedback and supervisory control. To illustrate the main idea behind the proposed approach, we first assume availability of measurements of all the process state variables. For the processes under consideration, a family of candidate control configurations, characterized by different manipulated inputs, is first identified. For each control configuration, a Lyapunov-based controller that enforces asymptotic closed-loop stability in the presence of constraints, is designed, and the constrained stability region, associated with it, is explicitly characterized. A nonlinear model-based fault-detection filter is used to compute the expected closed-loop behavior in the absence of faults. Deviations of the process states from the expected closed-loop behavior are used to detect faults. A switching policy is then derived, on the basis of the stability regions, to orchestrate the activation/deactivation of the constituent control configurations in a way that guarantees closed-loop stability in the event that a failure is detected. Often, in chemical process applications, all state variables are not available for measurement. To deal with the problem of lack of process state measurements, a nonlinear observer is designed to generate estimates of the states, which are then used to implement the state feedback controller and the fault-detection filter. A switching policy is then derived to orchestrate the activation/deactivation of the constituent control configurations in a way that accounts for the estimation error. Finally, simulation studies

are presented to demonstrate the implementation and evaluate the effectiveness of the proposed fault-tolerant control scheme.

In chapter 9 we focus on predictive control of linear parabolic partial differential equations (PDEs) with state and control constraints. Initially, the PDE is written as an infinite-dimensional system in an appropriate Hilbert space. Next, modal decomposition techniques are used to derive a finite-dimensional system that captures the dominant dynamics of the infinite-dimensional system, and express the infinite-dimensional state constraints in terms of the finite-dimensional system state constraints. A number of model predictive control (MPC) formulations, designed on the basis of different finite-dimensional approximations, are then presented and compared. The closed-loop stability properties of the infinite-dimensional system under the low order MPC controller designs are analyzed, and sufficient conditions that guarantee stabilization and state constraint satisfaction for the infinite-dimensional system under the reduced order MPC formulations are derived. Other formulations are also presented which differ in the way the evolution of the fast eigenmodes is accounted for in the performance objective and state constraints. The impact of these differences on the ability of the predictive controller to enforce closed-loop stability and state constraints satisfaction in the infinite-dimensional system is analyzed. Finally, the MPC formulations are applied through simulations to the problem of stabilizing the spatially-uniform unstable steady-state of a linear parabolic PDE subject to state and control constraints.

In chapters 10–11, we focus on the development and application of predictive-based strategies for control of particle size distribution (PSD) in continuous and batch particulate processes described by population balance models (PBMs). The control algorithms are designed on the basis of reduced-order models, utilize measurements of principle moments of the PSD, and are tailored to address different control objectives for the continuous and batch processes. For continuous particulate processes, we develop a hybrid predictive control strategy to stabilize a continuous crystallizer

at an open-loop unstable steady-state. The hybrid predictive control strategy employs logic-based switching between model predictive control (MPC) and a fall-back bounded controller with a well-defined stability region. The strategy is shown to provide a safety net for the implementation of MPC algorithms with guaranteed stability region. For batch particulate processes, the control objective is to achieve a final PSD with desired characteristics subject to both manipulated input and product quality constraints. An optimization-based predictive control strategy that incorporates these constraints explicitly in the controller design is formulated and applied to seeded batch cooling crystallizer and protein crystallization example.

In chapter 12, we propose a two-level, optimization-based method for the derivation of tuning guidelines for PID controllers that take nonlinear process behavior explicitly into account. The central idea behind the proposed method is the selection of the tuning parameters in a way that has the PID controller emulate, as closely as possible, the control action and closed-loop response obtained under a given nonlinear controller, for a broad set of initial conditions and set-point changes. To this end, classical tuning guidelines (typically derived on the basis of linear approximations, running open or closed-loop tests) are initially used in the first level to obtain reasonable bounds on the range of stabilizing tuning parameters over which the search for the parameters best matching the PID and nonlinear controllers is to be conducted. In addition to stability, performance and robustness considerations for the linearized closed-loop system can be introduced in the first level to further narrow down the parameter search range. The bounds obtained from the first level are then incorporated as constraints on the optimization problem solved at the second level to yield a set of tuning parameter values that enforce closed-loop behavior under the PID controller that closely matches the closed-loop behavior under the nonlinear controller. Implications of the proposed method, as a transparent and meaningful link between the classical and nonlinear control domains, as well as possible extensions of the tuning guidelines and other implementation issues are discussed. Finally, the implementation of the proposed tuning

method is demonstrated through a nonlinear chemical reactor example.

Finally, Chapter 13 concludes by summarizing the overall research contributions of this doctoral thesis.

Chapter 2

State Feedback Hybrid Predictive Control of Linear Systems

2.1 Introduction

Virtually all practical control systems are subject to hard constraints on their manipulated inputs. One of the key limitations imposed by input constraints is the restriction on the set of initial states of the closed-loop system that can be steered to the origin with the available control action. For a control policy to be effective in dealing with the problem of input constraints, it needs to provide not only the stabilizing feedback control law, but also an explicit characterization of the set of initial conditions starting from where constrained closed-loop stability is guaranteed. The absence of an a priori explicit characterization of this set can impact on the practical implementation of the given control policy by requiring extensive closed-loop simulations over the whole set of possible initial conditions, to check for stability, or by limiting operation within an unnecessarily small and conservative neighborhood of the operating point. These considerations have motivated significant work on the design of stabilizing bounded control laws that guarantee explicitly-defined, large regions of attraction for the closed-loop system (e.g., see [127, 221]). Recently, we developed in [66] (see also [69]) a class of

bounded robust Lyapunov-based nonlinear controllers, inspired in part by the results on bounded control originally presented in [127], that enforce robust stability in the closed-loop system and provide, at the same time, an explicit characterization of the region of guaranteed closed-loop stability. Despite their well-characterized stability and constraint-handling properties, the above controllers are not necessarily optimal with respect to an arbitrary performance criterion (in [66] it is shown that these controllers are inverse optimal with respect to meaningful costs).

Currently, MPC, also known as receding horizon control (RHC), is one of the few control methods for handling constraints within an optimal control setting. Here the control is obtained by solving repeatedly, on-line, a finite-horizon constrained open-loop optimal control problem. The popularity of this approach stems largely from its ability to handle, *inter alia*, multi-variable interactions, constraints on controls and states, and optimization requirements. Its use in several commercial applications is also well-documented in the literature (e.g., see [186, 174]). This motivated numerous research investigations into the stability properties of MPC leading to a plethora of MPC formulations that focus on closed-loop stability (e.g., see [3, 153, 141] for extensive surveys of these developments). For example, for open-loop stable and integrating linear systems with bounded inputs, infinite horizon formulations have been developed that are globally stabilizing (e.g., see [182, 244]). This progress notwithstanding, the issue of obtaining, *a priori* (i.e. before controller implementation), an analytic characterization of the region of constrained closed-loop stability for MPC controllers remains to be adequately addressed, particularly for open-loop (exponentially) unstable plants where, unlike stable plants, stabilization is the overriding requirement. Part of the difficulty in this regard owes to the fact that, unlike in analytical bounded control, the stability of MPC feedback loops depends on a complex interplay between several factors such as the stabilizability of the initial condition, the penalties in the performance index, and the choice of the control horizon. *A priori* knowledge of the stability region of MPC requires an explicit characterization of these interplays. This difficulty

can impact on the implementation of MPC on unstable plants by requiring extensive closed-loop simulations over the whole set of possible initial conditions, to check for stability.

Motivated by the above, we propose in this chapter a controller switching strategy that merges the bounded control approach with MPC in a way that allows both approaches to complement the stability and optimality properties of each other. The guiding principle in realizing this strategy is the idea of decoupling optimality from the characterization of the region of constrained closed-loop stability. Specifically, by relaxing the optimality requirement, an explicit bounded feedback control law is designed and an explicit large estimate of the region of constrained closed-loop stability, which is not unnecessarily conservative, is computed. An optimal MPC controller that minimizes a given cost functional subject to the same constraints is then designed and implemented within the stability region of the bounded controller. Switching laws, that place appropriate restrictions on the evolution of the closed-loop trajectory under MPC within the stability region are then constructed to orchestrate the transition between the two controllers in a way that guarantees closed-loop stability for all initial conditions within the stability region. The switching scheme is shown to provide a safe mechanism for the practical implementation of MPC, especially for open-loop unstable systems, by providing, through off-line computations, a priori knowledge of a large set of initial conditions for which closed-loop stability is guaranteed.

The idea of switching, between different controllers (or models), for the purpose of achieving some objective that either cannot be achieved or is more difficult to achieve using a single controller has been widely used in the literature, and in a variety of contexts. Examples include controller switching in gain scheduled control (e.g., [197]), logic-based switching in adaptive control (e.g., [92]), hybrid control of mixed logical dynamical systems [18], the use of multiple linear models for transition control (e.g., [16]) and scheduled predictive control of nonlinear processes (e.g., [13]). In this chap-

ter, switching is employed between two structurally-different, though complementary, control approaches as a tool for reconciling the objectives of optimal stabilization of the constrained closed-loop system (through MPC) and the a priori (off-line) determination of set of initial conditions for which closed-loop stability is guaranteed (through bounded control). The rest of the chapter is organized as follows. In section 2, we present the class of systems considered and review briefly how the constrained control problem is addressed in both bounded and MPC. We then proceed in section 3 to formulate the controller switching problem and propose a number of switching schemes that, with varying degrees of flexibility, address the problem. Finally, the implementation of the proposed switching schemes is demonstrated through numerical simulations.

2.2 Preliminaries

In this chapter, we consider continuous-time linear time-invariant (LTI) systems with input constraints, with the following form:

$$\dot{x}(t) = Ax(t) + Bu(t), u(t) \in \mathcal{U} \subset \mathbb{R}^m \quad (2.1)$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables, $u = [u_1 \cdots u_m]'$ is the vector of manipulated inputs, taking values in a compact and convex subset, $\mathcal{U}$, of $\mathbb{R}^m$ that contains the origin in its interior. The matrices A and B are constant $n \times n$ and $n \times m$ matrices, respectively. The pair (A, B) is assumed to be controllable. Throughout the chapter, the notation $\|\cdot\|$ is used to denote the standard Euclidean norm of a vector, while the notation $\|\cdot\|_Q$ refers to the weighted norm, defined by $\|x\|_Q^2 = x'Qx$ for all $x \in \mathbb{R}^n$, where Q is a positive definite symmetric matrix and x' denotes the transpose of x . Furthermore, the notation $x(T^-)$ is used to denote the limit of the trajectory $x(t)$ as T is approached from the left, i.e. $x(T^-) = \lim_{t \rightarrow T^-} x(t)$. We assume that measurements of the entire state, $x(t)$, are available for all t .

2.2.1 Bounded Lyapunov-based control

Consider the Lyapunov function candidate $V = x'Px$, where P is a positive-definite symmetric matrix that satisfies the Riccati equation

$$A'P + PA - PBB'P = -\bar{Q} \quad (2.2)$$

for some positive-definite matrix $\bar{Q}$. Using this Lyapunov function, we can construct, using Sontag's formula for bounded controls proposed in [127] (see also [66]), the following bounded nonlinear controller

$$u(x) = -2k(x)B'Px := b(x) \quad (2.3)$$

where

$$k(x) = \left(\frac{L_f V + \sqrt{(L_f V)^2 + (u_{max}\|(L_g V)'\|)^4}}{\|(L_g V)'\|^2 \left[1 + \sqrt{1 + (u_{max}\|(L_g V)'\|)^2} \right]} \right) \quad (2.4)$$

$L_f V = x'(A'P + PA)x$ and $(L_g V)' = 2B'Px$. This control law is continuous everywhere in the state space and smooth away from the origin. For the above controller, one can show, using a Lyapunov argument, that whenever the closed-loop state trajectory evolves within the state space region described by the set

$$\Phi(u_{max}) = \{x \in \mathbb{R}^n : L_f V < u_{max}\|(L_g V)'\|\} \quad (2.5)$$

the resulting control action respects the constraints (i.e. $\|u\| \leq u_{max}$) and enforces, simultaneously, the negative-definiteness of the time-derivative of the Lyapunov function, $\dot{V} < 0$, along the trajectories of the closed-loop system. Note that the size of the set Φ depends on the magnitude of the constraints in a way such that the tighter the constraints, the smaller the region described by this set. Starting from any initial state within $\Phi(u_{max})$, asymptotic stability of the constrained closed-loop system can be guaranteed, provided that the closed-loop trajectory remains within the region described by $\Phi(u_{max})$. To ensure this, we consider initial conditions that belong to an invariant subset (preferably the largest) which we denote by $\Omega(u_{max})$. This idea was

first proposed in [66] in the context of bounded robust control of constrained nonlinear systems. A common way of constructing such a subset is using the level sets of V , i.e.

$$\Omega(u_{max}) = \{x \in \mathbb{R}^n : x'Px \leq c_{max}\} \quad (2.6)$$

where $c_{max} > 0$ is the largest number for which all nonzero elements of $\Omega(u_{max})$ are contained within $\Phi(u_{max})$. The invariant region, described by the set $\Omega(u_{max})$, provides an estimate of the stability region, starting from where the origin of the constrained closed-loop system under the control law of Eqs.2.3-2.4 is guaranteed to be asymptotically stable.

Remark 2.1: The bounded control law of Eqs.2.3-2.4 will be used in section 3 to illustrate the basic idea of the proposed hybrid control scheme. Our choice of using this particular design is motivated by its explicit structure and well-defined region of stability. However, our results are not restricted to this particular design. Any other analytical bounded control law, with an explicit structure and well-defined region of stability, can be used including, for example, the bounded controls developed in [221] for certain classes of constrained linear systems. It is also worth noting that the bounded nonlinear control law of Eqs.2.3-2.4 can be viewed as a “linear” controller with a variable (state-dependent) nonlinear gain. The fact that the nonlinear gain is shaped to account explicitly for the presence of input constraints allows one to obtain a larger region, in the state space, where the controller respects the constraints, than would be obtained, for example, by bounding (clipping) a constant-gain linear controller that has already been designed without taking the constraints into account (see [66] for examples).

Remark 2.2: The exact computation of the null controllable region, for general constrained linear systems, remains an open problem in control (e.g., see [98] for some recent results in this area). Most research has therefore focused on obtaining an estimate of the domain of attraction under a given stabilizing feedback law. A common, and computationally convenient, way of providing this estimate is through Lyapunov-

based invariant regions (level sets). While these regions do not necessarily capture the entire domain of attraction, we will use them throughout the chapter only for a concrete illustration of the basic ideas of the results. It is possible to obtain better (i.e. less conservative) estimates by using, for example, combination of several Lyapunov functions.

2.2.2 Model predictive control

We consider model predictive control of the system described by Eq.2.1. For this case, MPC at state x and time t is conventionally obtained by solving, on-line, a finite horizon optimal control problem of the form

$$P(x, t) \quad : \quad \min\{J(x, t, u) | u \in S\} \quad (2.7)$$

where $S = S(t, T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ , mapping $[t, t + T]$ into $\mathcal{U}$, T is the specified horizon. A control $u(\cdot)$ in S is characterized by the sequence $\{u[k]\}$ where $u[k] := u(k\Delta)$. A control $u(\cdot)$ in S satisfies $u(t) = u[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. The performance index is given by

$$\begin{aligned} J(x, t, u(\cdot)) &= \int_t^{t+T} [\|x^u(s; x, t)\|_Q^2 + \|u(s)\|_R^2] ds \\ &+ F(x(t+T)) \end{aligned} \quad (2.8)$$

where R and Q are strictly positive definite, symmetric matrices, $x^u(s; x, t)$ denotes the solution of Eq.2.1, due to control u , with initial state x at time t , and $F(\cdot)$ denotes the terminal penalty. In addition to penalties on the state and control action, the objective function may also include penalties on the rate of input change, reflecting limitations on actuator speed (e.g., a large valve requiring few seconds to change position). The minimizing control $u^0(\cdot) \in S$ is then applied to the plant over the interval $[k\Delta, (k+1)\Delta)$ and the procedure is repeated indefinitely. This defines an implicit model predictive control law

$$M(x) \quad := \quad u^0(t; x, t) \quad (2.9)$$

It is well known that the control law defined by Eqs.2.7-2.9 is not necessarily stabilizing. To achieve closed-loop stability, early versions of MPC focused on tuning the horizon length, T , and/or increasing the terminal penalty (see [24] for a critique of these approaches), while more recent formulations focused on imposing stability constraints on the optimization (see [3, 141] for surveys of different constraints proposed in the literature and the concomitant theoretical issues). Obviously, even when stability constraints are imposed on the optimization, any guarantee of closed-loop stability remains contingent upon making the appropriate choice of the initial condition, which must belong to the stability region of the MPC controller which, in turn, is a complex function of the constraints, the performance objective, and the horizon length. However, the implicit nature of the MPC control law, obtained through repeated on-line optimization, limits our ability to obtain, a priori, an explicit characterization of the admissible initial conditions starting from where the given MPC controller (with fixed performance index and horizon length) is guaranteed to enforce asymptotic closed-loop stability. Therefore, the initial conditions are tested usually through closed-loop simulations which can add to the computational burden prior to the implementation of MPC.

2.3 Uniting bounded control and MPC: a hybrid control strategy

By comparing the bounded and MPC controller designs reviewed in the previous section, some tradeoffs with respect to their stability and optimality properties are observed. For example, while the bounded controller possesses a well-defined region of admissible initial conditions that guarantee closed-loop stability in the presence of constraints, the performance of this controller is not guaranteed to be optimal with respect to an arbitrary performance criterion (in [66] we show that a more general class of bounded robust controllers are inverse optimal with respect to a meaningful, though

not specified a priori, cost functional that imposes penalty on the state and control action). On the other hand, the MPC controller is well-suited for handling constraints within an optimal control setting; however, the analytical characterization of its set of admissible initial conditions is a more difficult task than it is through bounded control. In this section, we show how to reconcile the two approaches by means of a switching scheme that combines the desirable properties of both approaches.

2.3.1 Problem formulation and overview of solution

Consider the linear time-invariant system of Eq.2.1, subject to input constraints, $\|u\| \leq u_{max}$, for which the bounded controller of Eqs.2.3-2.4 and MPC controller of Eqs.2.7-2.9 have been designed. We formulate the controller switching problem as the one of designing a set of switching laws that orchestrate the transition between the MPC controller and the bounded controller in a way that: 1) respects input constraints, 2) guarantees asymptotic stability of the origin of the closed-loop system starting from any initial condition in the set $\Omega(u_{max})$ defined in Eq.2.6, and 3) guarantees recovery of the MPC optimal performance whenever the pertinent stability criteria are met. For a precise statement of the problem, we first cast the system of Eq.2.1 as a switched system of the form:

$$\dot{x} = Ax + Bu_{i(t)}, \|u_i\| \leq u_{max}, i(t) \in \{1, 2\} \quad (2.10)$$

where $i : [0, \infty) \rightarrow \{1, 2\}$ is the switching signal which is assumed to be a piecewise continuous (from the right) function of time, implying that only a finite number of switches between the two controllers is allowed on any finite interval of time. The index $i(t)$, which takes values in the finite set $\{1, 2\}$, represents a discrete state that indexes the control input, u , with the understanding that $i(t) = 1$ if and only if $u_i(x(t)) = M(x(t))$ (i.e. MPC is used) and $i(t) = 2$ if and only if $u_i(x(t)) = b(x(t))$ (i.e. bounded control is used). The value of $i(t)$ is determined by a higher-level supervisor responsible for executing the transition between the two controllers. Our goal is to construct

a switching law of the form $i(t) = \psi(x(t), t)$ that provides the supervisor with the switching times that ensure stabilizing transitions between the two controllers. This, in turn, determines the time-course of the discrete state $i(t)$. A schematic representation of the proposed hybrid control structure is depicted in Figure 2.1.

In the remainder of this section, we present two switching schemes that address this problem. The first scheme is formalized in Theorem 2.1 (section 2.3.2) and focuses primarily on closed-loop stability using classical MPC formulations, while the second scheme deals with feasibility-based switching and shows how more advanced MPC formulations can also be accommodated by adjusting the switching logic appropriately. The proof of Theorem 2.1 is given after the theorem.

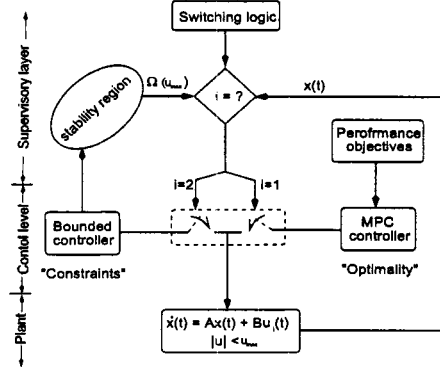


Figure 2.1: A schematic representation of the hierarchical hybrid control structure merging the bounded and MPC controllers.

2.3.2 Stability-based controller switching

Theorem 2.1: *Consider the constrained LTI system of Eq.2.10, with any initial condition $x(0) = x_0 \in \Omega(u_{max})$, where $\Omega(u_{max})$ was defined in Eq.2.6, under the model predictive controller of Eqs.2.7-2.9. Also let $T_s > 0$ be the earliest time for which the closed-loop state under MPC satisfies*

$$- \|x(T_s^-)\|_Q^2 + \|B'Px(T_s^-)\|^2 + 2x'(T_s^-)PBu(T_s^-) \geq 0 \quad (2.11)$$

then the switching rule

$$i(t) = \begin{cases} 1, & 0 \leq t < T_s \\ 2, & t \geq T_s \end{cases} \quad (2.12)$$

where $i(t) = 1 \Leftrightarrow u_i(x(t)) = M(x(t))$ and $i(t) = 2 \Leftrightarrow u_i(x(t)) = b(x(t))$. guarantees that the origin of the constrained closed-loop system is asymptotically stable.

Proof of Theorem 2.1: We proceed in two steps. First, we show that, starting from any initial condition within the set $\Omega(u_{max})$, the bounded control law of Eqs.2.3-2.4 asymptotically stabilizes the closed-loop. Second, we establish that switching between the bounded and the MPC, according to the rule of Eqs.2.11-2.12, guarantees asymptotic closed-loop stability, for any $x_0 \in \Omega(u_{max})$.

Step 1: Consider the system of Eq.2.1, under the control law of Eq.2.3-2.4. Evaluating the time-derivative of the Lyapunov function along the closed-loop trajectories

$$\begin{aligned} \dot{V} &= L_f V + L_g V u \\ &= \frac{L_f V \sqrt{1 + (u_{max} \|(L_g V)'\|)^2} - \sqrt{(L_f V)^2 + (u_{max} \|(L_g V)'\|)^4}}{\left[1 + \sqrt{1 + (u_{max} \|(L_g V)'\|)^2}\right]} \end{aligned} \quad (2.13)$$

It is clear from the last equality above that when $L_f V < 0$, we have $\dot{V} < 0$. Furthermore, when $0 \leq L_f V < u_{max} \|(L_g V)'\|$, we have $(L_f V)^2 < (u_{max} \|(L_g V)'\|)^2$ and, therefore,

$$-\sqrt{(L_f V)^2 + (u_{max} \|(L_g V)'\|)^4} < L_f V \sqrt{1 + (u_{max} \|(L_g V)'\|)^2} \quad (2.14)$$

Substituting the above estimate into the expression for $\dot{V}$ in Eq.2.13, we have that $\dot{V} < 0$. To summarize: whenever $L_f V < u_{max} \|(L_g V)'\|$, we have $\dot{V} < 0$. Since $\Omega(u_{max})$ is taken to be the largest invariant set, where this inequality holds for all $x \neq 0$, then starting from any initial state $x(0) \in \Omega(u_{max})$, the inequality of Eq.2.5 holds for all times, and consequently, $\dot{V} < 0 \quad \forall 0 \neq x \in \Omega(u_{max})$, which implies that the closed-loop system, under the control law of Eqs.2.3-2.4 is asymptotically stable.

Step 2: Consider the switched system of Eq.2.10, subject to the switching rule of Eqs.2.11-2.12, with any initial state $x(0) \in \Omega(u_{max})$. From the definition of T_s given in Theorem 2.1, it is clear that if T_s is a finite number, then $\dot{V}(x^M(t)) < 0 \quad \forall 0 \leq t < T_s$,

where the notation $x^M(t)$ denotes the closed-loop state under MPC at time t , which implies that $x(t) \in \Omega(u_{max}) \forall 0 \leq t < T_s$ (or that $x(T^-) \in \Omega(u_{max})$). This fact, together with the continuity of the solution of the switched system, $x(t)$, (which follows from the fact that the right hand side of Eq.2.10 is continuous in x and piecewise continuous in t since only finite number of switches is allowed over any finite time interval) implies that, upon switching (instantaneously) to the bounded controller at $t = T_s$, we have $x(T_s) \in \Omega(u_{max})$ and $u(t) = b(x(t))$ for all $t \geq T_s$. Therefore, from our analysis in step 1 we conclude that $\dot{V}(x^b(t)) < 0 \forall t \geq T_s$. In summary, the switching rule of Eqs.2.11-2.12 guarantees that, starting from any $x(0) \in \Omega(u_{max})$, $\dot{V}(x(t)) < 0 \forall 0 \neq x \in \Omega(u_{max}), \forall t \geq 0$. which implies that the switched closed-loop system is asymptotically stable. Note that if no such T_s exists, then we simply have $\dot{V}(x^M(t)) < 0 \forall t \geq 0$ and the closed-loop system is, again, asymptotically stable.

Remark 2.3: Theorem 2.1 describes a stability-based switching strategy for control of linear systems with input constraints. The three main components of this strategy include the bounded controller, the MPC controller, and a higher-level supervisor that orchestrates the switching between the two controllers. The implementation of this hybrid strategy is best understood through the following stepwise procedure (see Figure 2.2):

- Given the system model of Eq.2.1 and the constraints on the input, design the bounded controller using Eqs.2.3-2.4. Given the performance objective and a choice of the horizon length, design the MPC controller.
- Compute the stability region estimate for the bounded controller, $\Omega(u_{max})$, using Eqs.2.5-2.6.
- Initialize the closed-loop system, using the MPC controller, at any initial condition x_0 within $\Omega(u_{max})$.

- Monitor the evolution of the closed-loop trajectory (by checking Eq.2.11 at each time) until the earliest time that Eq.2.11 is met, T_s .
- At the earliest time that Eq.2.11 is met, discontinue MPC implementation, switch to the bounded controller and implement it for all future times.

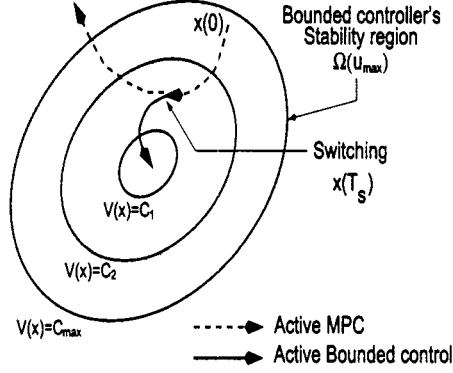


Figure 2.2: A schematic representation of the implementation of the stability-based controller switching proposed in Theorem 2.1.

Remark 2.4: The relations of Eqs.2.11-2.12 represent the switching rule that the supervisor observes when deciding if a switch between the two controllers is needed. The implementation of this rule requires only the evaluation of the algebraic expression on the left hand-side in Eq.2.11, which is the rate at which the Lyapunov function (used in designing the bounded controller and characterizing its stability region) grows or decays along the trajectories of the closed-loop system. By observing this rule, the supervisor is essentially tracking the temporal evolution of V under MPC, so that whenever an increase in V is detected after its initial implementation, MPC is disengaged from the closed-loop and the bounded controller is switched in, thus steering the closed-loop trajectory to the origin asymptotically. This logic guarantees that, under MPC, the closed-loop trajectory never escapes $\Omega(u_{max})$ before the bounded controller can be switched in. The idea of designing the switching logic based on monitoring the state's temporal evolution with respect to the stability region was introduced in [67] for control of switched (multi-modal) nonlinear systems with input constraints.

Remark 2.5: At this point, it is important to clarify the following issue. For most predictive controller formulations with finite horizon and constraints it is not possible to determine a priori (i.e., without any calculation being performed on-line) the stability region for the closed-loop system (i.e., a closed and bounded invariant set of initial conditions for which a finite-horizon predictive controller guarantees closed-loop stability). On this issue of computation of stability regions, we want to explicitly elaborate on the paper [36]. While not shown explicitly as a result, based on the work in [36] that focuses on optimal control of linear systems with state and control constraints, it can be argued that for linear systems without uncertainty, for a given convex polyhedral set that lies within the so-called null controllable region (denoted in the work by X_{max} —the set of initial conditions starting from where a given system can be stabilized in the presence of constraints, irrespective of the choice of the controller) one can come up with a sufficiently large finite value of the prediction horizon (that may be conservative, and even more so for points inside the polyhedron) that the problem is equivalent to an infinite horizon problem, and furthermore, for this sufficiently large finite value of the prediction horizon, the corresponding polyhedral set can be taken to be an estimate of the closed-loop stability region under the finite-time optimal controller with the sufficiently large horizon. This argument for computing an estimate of the stability region is based on convexity and is restricted to linear systems without uncertainty.

Note also that it can be shown that for infinite time formulations (which cannot be implemented in practice, and is essentially a theoretical construct), the set of initial conditions starting from where closed-loop stability is guaranteed is exactly the null controllable region. Formulations that allow for the prediction horizon to depend on the initial condition, and in essence allow for the value of the horizon to be increased as much as required (with a subsequent increase in the resulting cost of the optimization problem needed to be solved for the computation of the control action), their stability region approximates the null-controllable region to desired accuracy (via allowing the

use of horizons that tend to infinity, for example, see [36]; see also [135] for similar results in the nonlinear setting). For most predictive controllers with a finite horizon, however, a stability region cannot be a priori and explicitly computed; this is definitely the case for general classes of nonlinear systems as well as for linear/nonlinear systems with uncertainty (in particular, results of the type [36] for computing an estimate of the closed-loop stability region for a sufficiently large horizon for linear systems as discussed above cannot be extended to nonlinear systems and linear/nonlinear systems with uncertainty). This motivates the development of the hybrid predictive control structure where a hybrid predictive controller together with its associated stability region can be computed off-line and closed-loop stability under such a controller is guaranteed for any initial condition with this stability region. Note also, that the main idea behind switching remains that of switching to a controller with an off-line explicit estimate of its stability region; and the choice of controllers is not restricted to Lyapunov-based controllers alone. Any other controller that provides an explicit estimate of its stability region can be used in conjunction with the other predictive control approaches within a switched framework as proposed in this chapter, for providing an explicit characterization of the closed-loop stability region.

Remark 2.6: In the case when Eq.2.11 is never fulfilled, i.e. the Lyapunov function continues to decay monotonically along the closed-loop trajectories under MPC, the switching rule of Eq.2.12 ensures that only MPC is implemented for all times (no switching occurs) since it is asymptotically stabilizing. In this case, the optimal performance of MPC is fully recovered.

Remark 2.7: Note that the proposed approach does not turn $\Omega(u_{max})$ into a stability region for MPC. What the approach does, however, is turn $\Omega(u_{max})$ into a stability region for the switched closed-loop system. The value of this can be understood in light of the difficulty in obtaining, a priori, an analytical characterization of the set of admissible initial conditions that the MPC controller can steer to the origin in

the presence of input constraints. By using the bounded controller as a fall-back controller, the switching scheme allows us to safely initialize the closed-loop system anywhere within $\Omega(u_{max})$ using MPC, with the guarantee that the bounded controller can always intervene (through switching) to “rescue” closed-loop stability in case the closed-loop starts to become unstable under MPC (due, for example, to a poor choice of the initial condition or improper tuning of MPC). This safety feature distinguishes the bounded controller from other fall-back controllers that could be used, such as PID controllers, which do not provide a priori knowledge of the constrained stability region and, therefore, do not guarantee a safe transition in the case of unstable plants. For these controllers, a safe transition is critically dependent on issues such as whether or not the operator is able to properly tune the controller online, which, if not achieved, can possibly result in instability and/or performance degradation. The transition from the MPC to the bounded controller, on the other hand, is not fraught with such uncertainty and is always safe because of the a priori knowledge of the stability region. This aspect helps make plant operation safer and smoother.

Remark 2.8: In contrast to Lyapunov-based MPC approaches (e.g. contractive MPC [167], CLF-based RHC [170]), the Lyapunov-based stability condition is enforced at the supervisory level, via continuous monitoring of $V(t)$, rather than being incorporated as an inequality constraint in the optimization problem. This fact may help reduce the complexity of the optimization problem, without loss of stability, by reducing the number of constraints involved. The underlying idea here is that of decoupling the stability requirement from optimality. In the proposed hybrid scheme, it is switching to the bounded controller, with its well-defined stability region, that safeguards against potential closed-loop instability arising as a consequence of implementing MPC, for which the stability region is not known a priori. On the other hand, MPC provides, by design, the desired optimal performance under constraints. The switching logic then ensures that any such optimal solution is implemented only when it is asymptotically stabilizing.

Remark 2.9: The fact that the switching rule of Eqs.2.11-2.12 does not permit any transient increase in V under MPC (by immediately terminating MPC at the first instance that an increase is detected) can limit the achievable closed-loop performance in cases when V , despite the observed initial increase, eventually decays monotonically. From an optimal performance standpoint, it would be desirable in this case to keep MPC in the closed-loop instead of switching to the bounded controller. This can be realized by modifying the switching rule of Eq.2.12 in a way that allows the supervisor to keep MPC in the closed-loop, even during finite transient increases in V , provided that the closed-loop trajectory does not leave $\Omega(u_{max})$ during this time. This requirement is necessary because $\Omega(u_{max})$ is not invariant under MPC and, therefore, once the trajectory leaves, there are no guarantees that it can ever be steered back to the origin, even with the bounded controller. Note, however, that this requirement by itself guarantees only that the closed-loop state is bounded. To ensure asymptotic convergence to the origin, a choice needs to be made as to how long the increases in V under MPC can be tolerated before switching to the bounded controller becomes necessary to enforce the monotonic decay of V for all future times.

Remark 2.10: Another direction in which the switching scheme of Theorem 2.1 can be generalized is to allow for multiple switchings between MPC and the bounded controller. According to Theorem 2.1, once MPC is terminated, the bounded controller is implemented for all future times. In a more general scheme, however, the bounded controller need not stay in the closed-loop for all future times. Instead, it can be employed only until it brings the closed-loop state trajectory sufficiently closer to the origin (for example, to a pre-determined inner level set of V) at which point MPC is re-activated and its behavior monitored once again. Chattering problems, due to back and forth switching, are avoided by allowing only a finite number of switches over any finite time-interval. This scheme offers the possibility of further enhancement in the closed-loop performance (over that obtained from the bounded controller) by implementing MPC for longer periods of time (as permitted by stability considerations

of course). It also allows for the possibility of stabilizing the closed-loop system starting from initial conditions for which a given (fixed-horizon) MPC is not stabilizing. This can occur, for example, when the stability region for such a controller is contained within $\Omega(u_{max})$. In this case, only a finite number of switches between the two controllers is needed to steer the closed-loop trajectory to that point, beyond which MPC can stay in the closed-loop for all times (with no further switching).

Remark 2.11: When the proposed hybrid control scheme is applied to a process online, state measurements are typically available only at discrete sampling instants (and not continuously). In this case, the restricted access that the supervisor has to the state evolution between sampling times can lead to the possibility that the closed-loop state trajectory under MPC leaves $\Omega(u_{max})$ without being detected, particularly if the initial condition is close to the boundary of $\Omega(u_{max})$ and/or the sampling period is too large. In such an event, switching to the bounded controller at the next sampling time may be too late to recover from the instability of MPC. To guard against this possibility, the switching rule can be modified by restricting the implementation of MPC within a subset of $\Omega(u_{max})$, computed such that, starting from this subset, the system trajectory is guaranteed to remain within $\Omega(u_{max})$ after a single sampling period. Explicit estimates of this subset, parameterized by the sampling period, can be readily obtained by integrating the Lyapunov dissipation inequality over one sampling period.

2.3.3 Switching using advanced MPC formulations

In Theorem 2.1, a classical MPC formulation (with no stability constraints) was used as an example to illustrate the basic idea of controller switching and how it can be used to aid MPC implementation. The value of the hybrid control structure (see Figure 2.1), however, is not restricted to classical MPC formulations, and extends to any MPC formulation, for which a priori knowledge of the set of admissible initial conditions is lacking (or computationally expensive to obtain). The structure can be adapted

to more advanced versions of MPC by appropriate design of the switching logic. To illustrate this point, consider the case when advanced MPC algorithms, that employ stability constraints in the optimization formulation, are used to compute the control action. An example of such formulations is the following:

$$\begin{aligned} \min_{u(\cdot)} \quad & \int_t^{t+T} (x'(s)Qx(s) + u'(s)Ru(s))ds \\ \text{s.t.} \quad & \dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \\ & u(\cdot) \in S, \quad x(t+T) = 0 \end{aligned} \tag{2.15}$$

which uses a terminal equality constraint (other types of stability constraints, such as control Lyapunov function (CLF)-based inequality constraints of the form $V(x(t+T)) < V(x(t))$ can be treated similarly). By incorporating stability conditions as part of the constrained optimization problem, asymptotic stability under MPC can be guaranteed provided that the initial optimization yields a feasible solution. However, inclusion of the stability constraints does not, by itself, provide any a priori explicit knowledge of the feasible initial conditions, which must then be identified through simulations. This observation suggests that the hybrid control structure can be used here to safeguard closed-loop stability in the event of MPC infeasibility. Specifically, in lieu of monitoring the growth of V , a different switching logic, based on feasibility of the optimization in Eq.2.15, needs to be implemented. The switching algorithm in this case can be summarized as follows (see also Figure 2.3 for a schematic representation):

- Given any initial condition x_0 within $\Omega(u_{max})$, check, off-line, whether the constrained optimization in Eq.2.15 yields a feasible solution. If a solution exists, no switching is needed and MPC can be implemented for all time.
- If an initial feasible solution is not found, implement the bounded controller instead, i.e. set $u(0) = b(x_0)$.
- While the bounded controller is in the closed-loop, continue to check, off-line, and as frequently as desired, the feasibility of the optimization in Eq.2.15.

- At the earliest time that a feasible solution is found, switch to the MPC controller, else the bounded controller remains active.

The above discussion underscores an important attribute of the proposed hybrid control structure, and that is the fact that it is not proposed as a substitute for stability constraints (even though it guarantees stability when the MPC formulation is not stabilizing), but as a reliable fall-back mechanism from which any MPC formulation can benefit.

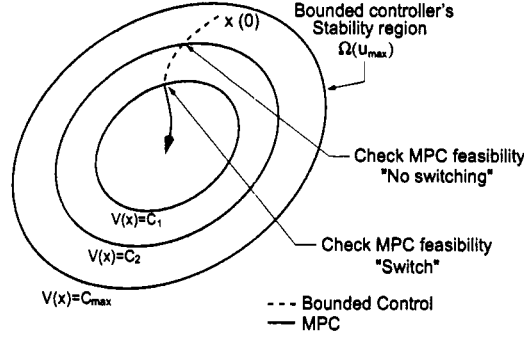


Figure 2.3: A schematic representation of the implementation of switching when a stability-handling MPC formulation is used.

Remark 2.12: An important issue in the practical implementation of MPC is the selection of the horizon length. It is well known that this selection can have a profound effect on nominal closed-loop stability, and results are available in the literature that establish, under certain assumptions, the existence of “sufficiently large” finite horizons that ensure stability (see, e.g., [182, 36]). However, a priori knowledge (without closed-loop simulations) of the minimum horizon length that guarantees feasibility or closed-loop stability, from an arbitrary initial condition, is currently not available. Therefore, in practice the horizon length is typically chosen based on heuristic criteria, tested through closed-loop simulations, and varied, if necessary, to achieve stability. Note that in all the switching schemes proposed in section 3, closed-loop stability is maintained independent of the horizon length, and therefore selection of the horizon

length can be made solely on the basis of what is computationally practical for the size of the optimization problem, without increasing, unnecessarily, the horizon length (and consequently the computational load) out of concern for stability.

Remark 2.13: The supervisory checks required in the implementation of the switching scheme do not incur any additional computational costs beyond that involved in MPC implementation alone. In the event that MPC is infeasible from a given initial condition (e.g., due to insufficient number of control moves), the supervisor, checks as frequently as desired, the initial feasibility of the MPC by actually solving the optimization problem at each time step and implementing the resulting solution (if found). If a solution is not found at a given time step, the supervisor implements the bounded controller (an inexpensive algebraic computation) and continues, in the mean time, to solve the optimization until it finds a feasible solution to implement.

Remark 2.14: It is important to point out that the switching schemes proposed in this chapter differ, both in their objective and implementation, from other MPC formulations involving switching which have appeared earlier in the literature. For example, in dual mode MPC [149] the strategy includes switching from MPC to a locally stabilizing controller once the state is brought near the origin by MPC. The purpose of switching in this approach is to relax the terminal equality constraint whose implementation is computationally burdensome for nonlinear systems. However, the set of initial conditions for which MPC is guaranteed to steer the state close to the origin is not explicitly known a priori. In contrast, switching from MPC to the bounded controller is used in our work only to prevent any potential closed-loop instability arising from implementing MPC without the a priori knowledge of the admissible initial conditions. So depending on the stability properties of the chosen MPC, switching may or may not occur, and if it occurs, it can take place near or far from the origin. Finally, we note that the notion of switching from a predictive to an unconstrained LQR controller, for times beyond a finite control horizon, has been used in [218, 36, 203]

in the context of determining a finite horizon that solves the infinite horizon linear quadratic regulation problem with constraints.

2.4 Simulation example

Consider the following linear system

$$\dot{x} = \begin{bmatrix} 0.5 & 0.25 \\ 0.5 & 1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u \quad (2.16)$$

where both inputs u_1, u_2 are constrained in the interval $[-5, 5]$ and the open-loop system has an unstable equilibrium point at the origin (A has two positive eigenvalues). We initially use Eqs.2.3-2.4 to design the bounded controller and construct its stability region via Eqs.2.5-2.6. The stability region estimated using Eqs.2.5-2.6 compares favorably with the estimate of the null controllable region (X_{max}) as shown in Figure 2.4; note also that improved estimates of the stability region could be obtained using a combination of several Lyapunov-functions. The matrix P is chosen to be $P = \begin{bmatrix} 1.1362 & 0.8102 \\ 0.8102 & 1.8658 \end{bmatrix}$. For the MPC controller, the parameters in the objective function of Eq.2.8 are chosen as penalty on the states, $Q = qI$, with $q = 1$, penalty on the control inputs, $R = rI$, with $r = 3$ and a horizon length of $T = 1$. The resulting quadratic program is solved using the MATLAB subroutine QuadProg, and the set of ODEs are integrated using the MATLAB solver ODE45.

As shown by the solid curve in Figure 2.5, applying the MPC controller from the initial condition $x_0 = [6 \ -2]'$ (which belongs to the stability region of the bounded controller, Ω , represented by the ellipse in Figure 2.5) leads to closed-loop instability. The corresponding input and output profiles are shown by the solid lines in Figure 2.6. Using the switching scheme of Theorem 2.1, however, we find that the supervisor detects an increase in V at $t = 0.425$ and therefore immediately switches at this time from the MPC controller to the bounded controller in order to preserve closed-loop stability. As expected, the bounded controller asymptotically stabilizes the plant (see

dotted lines in Figures 2.5 and 2.6). It is important to note that one could try to tune the horizon length further in order to achieve stability using the MPC controller. For example, we see from the dashed lines in Figures 2.5 and 2.6 that stability can be achieved by increasing the horizon length to $T = 1.5$. However, this conclusion could not be reached a priori, i.e. before running the closed-loop simulation in its entirety to check whether the choice $T = 1.5$ is appropriate. In contrast, closed-loop stability starting from the given initial condition, is guaranteed, a priori, under the switching scheme of Theorem 2.1.

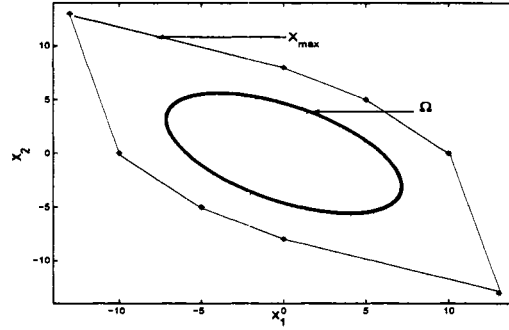


Figure 2.4: A comparison of the estimate of the null controllable region (X_{max}) with the stability region for the bounded controller estimated using Eqs.2.5-2.6.

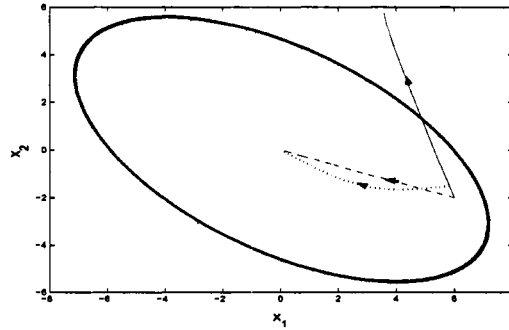


Figure 2.5: Closed-loop state trajectory under MPC with $T = 1$ (solid line), under the switching scheme of Theorem 2.1 with $T = 1$ (dotted line), and under MPC with $T = 1.5$ (dashed line).

In the following set of simulation runs, we demonstrate an application of the

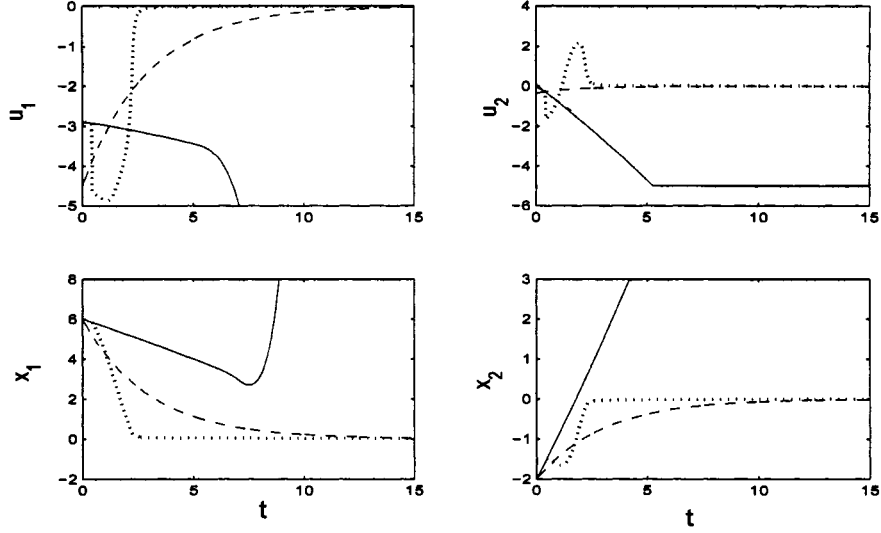


Figure 2.6: Input and closed-loop state profiles under MPC with $T = 1$ (solid line), under the switching scheme of Theorem 2.1 with $T = 1$ (dotted line), and under MPC with $T = 1.5$ (dashed line).

feasibility-based switching scheme, proposed in section 3.3, which uses a stable version of MPC, specifically, one employing a terminal equality constraint. For the MPC controller, the parameters in the objective function (Eq.2.15) are chosen as $q = 1$ and $r = 1$. The horizon length is chosen to be $T = 1.2$. As shown by the solid lines in Figure 2.7, starting from an initial condition $x_0 = [2 \ -2]'$, MPC is found to be feasible and is therefore implemented and kept in the closed-loop for all times driving the trajectory asymptotically to the origin. The corresponding input and output profiles are shown by the solid lines in Figure 2.8. The scenario, therefore, represents a reasonably well-tuned MPC for initial conditions around the origin. Now suppose that, after settling at the origin, a temporary disturbance drives the closed-loop trajectory to the point $x_i = [5 \ -5]'$ (which is still within the stability region of the bounded controller). After the disturbance disappears, MPC starting from this initial condition is found to be infeasible and, therefore, the bounded controller is employed instead. Implementation of the bounded controller proceeds until the closed-loop trajectory crosses the level set

denoted by Ω_2 in Figure 2.7, which occurs at $t = 0.6$. At this point, feasibility of MPC is checked again and the MPC controller is found to be feasible. Consequently, the bounded controller is terminated immediately and the MPC controller is employed in the closed-loop for the remaining time (inside Ω_2). This scenario is shown by the dotted lines in Figures 2.7 and 2.8. Once again, note that using a higher value of the horizon length, $T = 1.5$ can lead to MPC being feasible from the initial condition $x_i = [5 \ -5]'$ (dashed lines in Figures 2.7 and 2.8). This implies that, once the closed-loop state is driven by the disturbance to x_i , a re-tuning of the horizon length can make MPC feasible. However, a priori knowledge of the value of the horizon length that is guaranteed to work could not be obtained, short of testing feasibility at that condition. The prospect of disturbances throwing the system around would therefore necessitate running extensive closed loop simulations to determine such a value. The feasibility-based switching scheme, on the other hand, clearly guarantees closed-loop stability for any value of the horizon length. Following the disturbance, it drives the closed-loop trajectory to a point at which the original MPC controller (with $T = 1.2$) becomes feasible again and can therefore be implemented. In this particular example, feasibility was checked only at one level set. The frequency of feasibility checking could be made larger by introducing more level sets, or by checking feasibility at predetermined times.

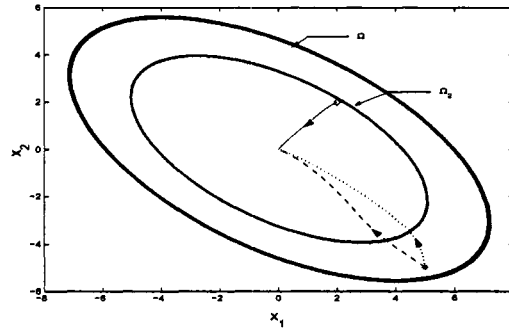


Figure 2.7: Closed-loop state trajectory starting close to the origin using MPC with $T = 1.2$ (solid line) and from a point farther away in state space using the switching scheme with a horizon length $T = 1.2$ (dotted line) and using only MPC with $T = 1.5$ (dashed line).

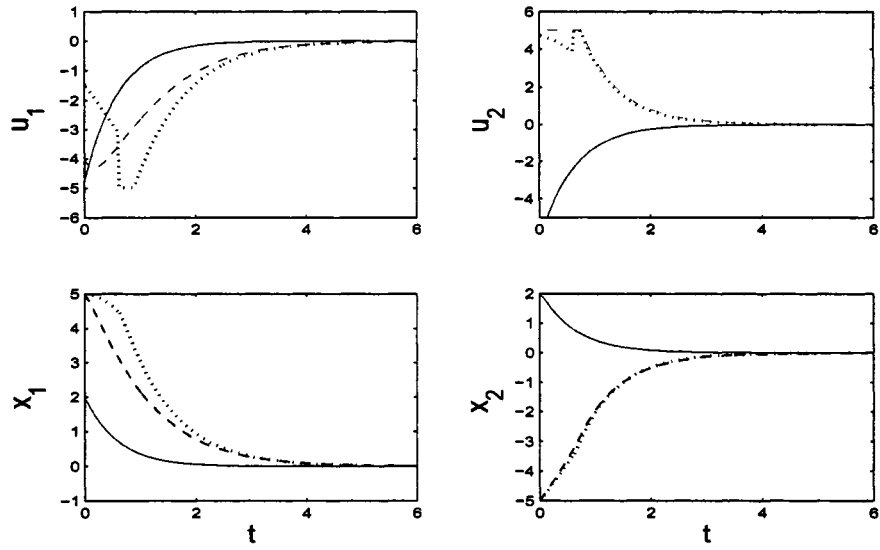


Figure 2.8: Input and closed-loop state profiles starting close to the origin using MPC with $T = 1.2$ (solid line) and from a point farther away in state space using the switching scheme with a horizon length $T = 1.2$ (dotted line) and using only MPC with $T = 1.5$ (dashed line).

Chapter 3

Output Feedback Hybrid Predictive Control of Linear Systems

3.1 Introduction

The operation of chemical processes invariably requires respecting constraints. The physical limitations on the functioning of the control actuators (for example, the maximum flow rate that a given pump can generate is always limited) impose constraints on the manipulated inputs, while performance, safety and environmental considerations (requiring, for example, the reactor temperature to stay below a certain value, or the emissions to be within an allowable range) give rise to state constraints. Constraints automatically impose limitations on our ability to steer the dynamics of the closed-loop system at will, and can cause severe deterioration in the closed-loop performance, and may even lead to closed-loop instability, if not explicitly taken into account at the stage of controller design.

For open-loop unstable processes, one of the key limitations imposed by input constraints is the restriction on the set of initial states of the closed-loop system that can be steered to the origin with the available control action (the so-called null controllable region). Furthermore, a given control law can typically guarantee closed-loop stability

from only a subset of initial states included in the null controllable region. An adequate characterization of the stability properties of a given stabilizing control law involves *a priori* (i.e., before controller implementation) knowledge of this region of stability, required to ascertain stabilizability of the system from a given initial condition. The absence of an *a priori* explicit characterization of this set (or an appropriate estimate thereof) can impact on the practical implementation of the given control policy.

Model Predictive Control (MPC), also known as receding horizon control, is one control method for handling constraints (both on manipulated inputs and state variables) within an optimal control setting [186]. In MPC, the control action is obtained by solving repeatedly, on-line, a finite-horizon constrained open-loop optimal control problem. The popularity of this approach stems largely from its ability to handle, among other issues, multi-variable interactions, constraints on controls and states, and optimization requirements. Its success in many commercial applications is well-documented in the literature (see, for example, [84, 174]). Numerous research studies have investigated the stability properties of model predictive controllers and led to a plethora of MPC formulations that focus on closed-loop stability (see, for example, [111, 87, 182, 48] and the review paper [141]).

The significant progress in understanding and improving the stability properties of MPC notwithstanding, the issue of obtaining, *a priori*, an explicit characterization of the region of constrained closed-loop stability for model predictive controllers remains to be adequately addressed. Part of the difficulty in this direction owes to the fact that the stability of model predictive controllers depends on a complex interplay between several factors such as the choice of the horizon length, the penalties in the performance index, the initial condition and the constraints on the state variables and manipulated inputs. *A priori* knowledge of the stability region requires an explicit characterization of these interplays which is a very difficult task. This difficulty can impact on the practical implementation of MPC by imposing the need for extensive closed-loop simulations

over the whole set of possible initial conditions to check for closed-loop stability, or by potentially limiting operation within an unnecessarily small neighborhood of the nominal equilibrium point.

The desire to implement control approaches that allow for an explicit characterization of their stability properties has motivated significant work on the design of stabilizing control laws, using Lyapunov techniques, that provide explicitly-defined, large regions of attraction for the closed-loop system; the reader may refer to [114] for a survey of results in this area. Recently, in [66, 68], a class of Lyapunov-based bounded robust nonlinear controllers, inspired by the results on bounded control originally presented in [127], was developed. The controllers enforce robust stability in the closed-loop system and provide, at the same time, an explicit characterization of the region of guaranteed closed-loop stability. Despite their well-characterized stability and constraint-handling properties, those Lyapunov-based controllers are not guaranteed to be optimal with respect to an arbitrary performance criterion.

However, the fact, that Lyapunov-based control methods provide an easily implementable controller with an explicitly characterized stability region for the closed-loop system motivates developing control techniques that use the *a priori* guarantees of stability provided by the bounded controller as a safety net for the implementation of the high performance model predictive controller. Reconciliation of different controllers (designed to satisfy different control objectives) calls for invoking the hybrid control paradigm, where the control structure consists of a blend of continuous (i.e., classical) controllers and discrete components (for example, a logic-based supervisor that determines switching between the various continuous controllers; see Figure 3.1). In [75], we proposed such a hybrid control structure, that employs switching between a model predictive controller and a bounded analytical controller for the state feedback stabilization of linear systems with input constraints. The hybrid control structure was subsequently extended to nonlinear systems with input constraints in [73]. This

structure differs from other hybrid control structures found in the literature, in the sense that it uses *structurally different* controllers as opposed to most of the work on hybrid control that employs switching between different models (which results in an implicit switching between different controllers), or employs multiple *structurally similar* controllers; examples along these lines include gain-scheduling as a tool for control of nonlinear systems (for example, see [197]), the use of multiple linear models for transition control (for example, see [16, 215]) and for predictive control (for example, see [201, 13]), and hybrid control of switched nonlinear systems [70].

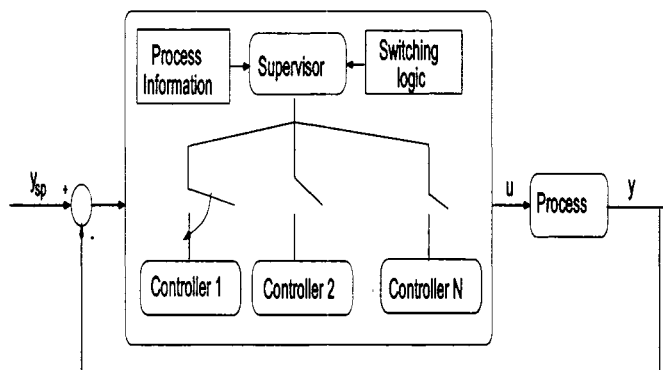


Figure 3.1: Schematic representation of a hybrid control structure.

In addition to constraints, another important issue in chemical process control is the lack of complete measurements of the process state variables. This problem is usually addressed by designing appropriate state estimators (observers) to obtain estimates of the state variables from the available measurements (outputs). The state estimators are then appropriately coupled with state feedback controllers, leading to output feedback controllers that can be implemented with the available measurements. The problems of state estimation and output feedback controller design for constrained systems have been the subject of several research studies. For example, recent results on moving horizon estimation, as an extension of Kalman filtering, for constrained and nonlinear processes can be found in [178]. Examples of results on output feedback stabilization of constrained systems include output feedback control of linear systems [208]

and the combination of a moving horizon regulator with a moving horizon observer for output feedback control of nonlinear systems [150]. In these works, however, the output feedback stability region of the constrained closed-loop system is not explicitly characterized.

Motivated by the above considerations, this chapter proposes a hybrid predictive control structure that seamlessly unites MPC and bounded control for the output feedback stabilization of linear systems with input constraints. The proposed strategy is based upon the idea of embedding the implementation of MPC within the output feedback stability region of the bounded controller, and using this controller as a “fall-back” in the event that MPC is unable to achieve closed-loop stability (due, for example, to improper tuning of MPC parameters). Due to the absence of complete state measurements, a state observer is constructed to provide the controllers, as well as the supervisor, with appropriate state estimates. The observer is tuned in a way so as to guarantee closed-loop stability for all initial conditions within the bounded controller’s output feedback stability region (which can be chosen arbitrarily close in size to its state feedback counterpart, provided that the observer gain is sufficiently large). Switching laws, that monitor the evolution of the state estimates, are derived to orchestrate the transition between the two controllers in a way that reconciles their respective stability and optimality properties and safeguards closed-loop stability in the event of MPC infeasibility or instability. In addition to the set of switching rules being different from the one proposed in [75] under state feedback, an important characteristic of the hybrid predictive control strategy under output feedback is the inherent coupling, brought about by the lack of full state measurements, between the tasks of controller design, characterization of the stability region and supervisory switching logic design, on one hand, and the task of observer design, on the other. The rest of the chapter is organized as follows: initially, we present some preliminaries that describe the class of systems considered, and review the design of a Luenberger state observer. Next, we review the design procedure, and stability properties, of both MPC and Lyapunov-based

bounded control, respectively. Then, we formulate the hybrid control problem within the framework of switched linear systems and propose the hybrid predictive control structure. The theoretical underpinnings and practical implications of the proposed hybrid predictive control structure are highlighted, and possible extensions of the supervisory switching logic, that address a variety of practical implementation issues, are discussed. Finally, two simulation studies are presented to demonstrate the implementation and evaluate the effectiveness of the proposed control strategy, as well as test its robustness with respect to modeling errors and measurement noise.

3.2 Preliminaries

In this chapter, we consider the problem of output feedback stabilization of continuous-time linear time-invariant (LTI) systems with input constraints, with the following state-space description:

$$\dot{x} = Ax + Bu \quad (3.1)$$

$$y = Cx \quad (3.2)$$

$$u \in \mathcal{U} \subset \mathbb{R}^m \quad (3.3)$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables, $y = [y_1 \cdots y_k]' \in \mathbb{R}^k$ denotes the vector of output variables and $u = [u_1 \cdots u_m]' \in \mathbb{R}^m$ is the vector of manipulated inputs, taking values in a compact and convex subset, $\mathcal{U} := \{u \in \mathbb{R}^m : \|u\| \leq u_{max}\}$, that contains the origin in its interior, where $\|\cdot\|$ denotes the standard Euclidean norm of a vector and $u_{max} > 0$ is the magnitude of actuator constraints. The matrices A , B and C are constant $n \times n$, $n \times m$ and $k \times n$ matrices, respectively. The pair (A, B) is assumed to be controllable and the pair (C, A) is assumed to be observable. Throughout the chapter, the notation $\|\cdot\|_Q$ refers to the weighted norm, defined by $\|x\|_Q^2 = x'Qx$ for all $x \in \mathbb{R}^n$, where Q is a positive-definite symmetric matrix and x' denotes the transpose of x .

We will now briefly review the design of a Luenberger state observer for the system of Eqs.3.1–3.2 which will be used later in the development of the hybrid predictive control structure (see Remark 3.1 below for a discussion on the use of other possible estimation schemes). Specifically, we consider a standard Luenberger state observer described by:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (3.4)$$

where $\hat{x} = [\hat{x}_1 \cdots \hat{x}_n]' \in \mathbb{R}^n$ denotes the vector of estimates of the state variables and L is a constant $n \times k$ matrix that multiplies the discrepancy between the actual and estimated outputs. Under the state observer of Eq.3.4, the estimation error in the closed-loop system, defined as $e = x - \hat{x}$, evolves, independently of the controller, according to the following equation:

$$\dot{e} = (A - LC)e \quad (3.5)$$

The pair (C, A) is assumed to be observable in the sense that the observer gain matrix, L , can be chosen such that the norm of the estimation error in Eq.3.5 evolves according to $\|e(t)\| \leq \kappa(\beta)\|e(0)\|\exp(-\beta t)$, where $-\beta < 0$ is the largest eigenvalue of $A - LC$ and $\kappa(\beta)$ is a polynomial function of β .

Remark 3.1: Referring to the state observer of Eq.3.4, it should be noted that the results presented in this chapter are not restricted to this particular class of observers. Any other observer that allows us to control the rate of decay of the estimation error at will, can be used. Our choice of using this particular observer design is motivated by the fact that it provides a transparent relationship between the temporal evolution of the estimation error bound and the observer parameters. For example, this design guarantees convergence of the state estimates in a way such that for larger values of β , the error decreases faster. As we discuss later (see the hybrid predictive control section), the ability to ensure a sufficiently fast decay of the estimation error is necessary in order to guarantee closed-loop stability under output feedback control. This requirement or constraint on the error dynamics is present even when other estimation schemes, such as

moving horizon observers, are used (for example, see [150, 178]) to ensure closed-loop stability. For such observers, however, it is difficult in general to obtain a transparent relationship between the tunable observer parameters and the error decay rate.

Remark 3.2: For large values of β , the estimation error could possibly increase to large values before eventually decaying to zero (a phenomenon known as “peaking”; see [216]). However, this does not pose a problem in our design because the physical constraints on the manipulated inputs prevent transmission of the incorrect estimates to the process (see also Remark 3.9 for a detailed discussion of this issue). In the special case when C is a square matrix of rank n , the observer gain matrix, L , can be chosen as $L = AC^{-1} - RC^{-1}$, where R is a diagonal matrix whose diagonal elements are of the form $R_{ii} = -\beta a_i$, where $a_i \neq a_j \geq 1$. For this choice of L , the evolution of each error state is completely decoupled from the rest of the error states, i.e., each error state evolves according to $\dot{e}_i = -\beta a_i e_i$. In this case, no peaking occurs since each error term decreases monotonically.

3.3 Model Predictive Control

We consider model predictive control of the system described by Eq.3.1, subject to the control constraints of Eq.3.3. The control action at time t is conventionally obtained by solving, on-line, a finite horizon optimal control problem [142] of the form:

$$P(x, t) : \min\{J(x, t, u(\cdot)) | u(\cdot) \in S\} \quad (3.6)$$

where $S = S(t, T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ , mapping $[t, t+T]$ into $\mathcal{U}$ and T is the specified horizon. A control $u(\cdot)$ in S is characterized by the sequence $\{u[k]\}$ where $u[k] := u(k\Delta)$. A control $u(\cdot)$ in S satisfies $u(t) = u[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. The performance index is given by:

$$J(x, t, u(\cdot)) = \int_t^{t+T} [\|x^u(s; x, t)\|_Q^2 + \|u(s)\|_R^2] ds + F(x(t+T)) \quad (3.7)$$

where R and Q are strictly positive-definite, symmetric matrices, $x^u(s; x, t)$ denotes the solution of Eq.3.1, due to control u , with initial state x at time t , and $F(\cdot)$ denotes the terminal penalty. In addition to penalties on the state and control action, the objective function may also include penalties on the rate of change of the input, reflecting limitations on actuator speed (for example, a large valve requiring few seconds to change position). The minimizing control $u^0(\cdot) \in S$ is then applied to the process over the interval $[k\Delta, (k+1)\Delta)$ and the procedure is repeated indefinitely. This defines an implicit model predictive control law:

$$M(x) := u^0(t; x, t) \quad (3.8)$$

It is well-known that the control law defined by Eqs.3.6–3.8 is not necessarily stabilizing [24]. To achieve closed-loop stability, early versions of MPC focused on tuning the horizon length, T , and/or increasing the terminal penalty (see [24] for a survey of these approaches), while more recent formulations focused on imposing stability constraints on the optimization (see [141] for surveys of different constraints proposed in the literature and the concomitant theoretical issues) or using off-line computations to come up with explicit model predictive control laws (for example, see [165]). The additional stability constraints serve either to enforce convergence of the states of the closed-loop system to the equilibrium point, or to force the states to reach some invariant terminal set at the end of the horizon.

For the sake of a concrete illustration of the hybrid predictive control strategy, we will consider the following stabilizing MPC formulation [111, 148], which includes a terminal equality constraint in the optimization problem (see Remark 3.3 below for more advanced MPC formulations that can also be used). This model predictive controller is mathematically described by:

$$J_s(x, t, u(\cdot)) = \int_t^{t+T} (x'(s)Qx(s) + u'(s)Ru(s))ds \quad (3.9)$$

$$\begin{aligned}
u(\cdot) &= \operatorname{argmin}\{J_s(x, t, u(\cdot)) | u(\cdot) \in S\} := M_s(x) \\
s.t. \quad \dot{x} &= Ax + Bu, \quad x(0) = x_0 \\
u(\cdot) &\in S \\
x(t+T) &= 0
\end{aligned} \tag{3.10}$$

Remark 3.3: It should be noted that the hybrid predictive control strategy to be proposed in this chapter is not restricted to the MPC formulation given in Eqs.3.9–3.10, or any other MPC formulation for that matter. The results apply to any MPC formulation for which *a priori* knowledge of the set of admissible initial conditions is lacking (or computationally expensive to obtain). In this sense, the above formulation is really intended as a symbolic example of stabilizing MPC formulations, with the understanding that other advanced MPC formulations that employ less restrictive constraints and/or penalties on the state at the end of the horizon can be used. Examples include formulations with terminal inequality constraints, such as contractive MPC (for example, see [167, 115]) and CLF-based receding horizon control (for example, see [170]), as well as formulations that employ a combination of terminal penalties and inequality constraints (for example, see [31, 132, 219]). The reader is referred to [141] for a more exhaustive list.

Remark 3.4: By incorporating stability conditions directly as part of the optimization problem in Eq.3.10, asymptotic stability under state feedback MPC is guaranteed provided that the initial condition is chosen so that the optimization yields a feasible solution. However, the implicit nature of the MPC control law, obtained through repeated on-line optimization, limits our ability to obtain, *a priori*, an explicit characterization of the admissible initial conditions starting from where the given model predictive controller (with fixed horizon length) is guaranteed to be feasible and enforce asymptotic stability. This set is a complex function of the constraints and the horizon length. Estimates of sufficiently large horizon lengths that ensure stability (for example, see [36]) are typically conservative and, if used, may lead to a significant computational burden due to the increased size of the optimization problem. There-

fore, in practice, the initial conditions and/or horizon lengths are usually chosen using ad hoc criteria and tested through closed-loop simulations which can add to the computational burden in implementing the model predictive controller. The difficulties encountered in characterizing the stability region under state feedback control carry over to the case of output feedback control, where the lack of state measurements requires that the control action be computed using the state estimates. Feasibility of the MPC optimization problem based on the state estimates, however, does not guarantee closed-loop stability or even the continued feasibility of the ensuing state estimates. This motivates implementing MPC within a hybrid control structure that provides a “fall-back” controller for which a region of constrained closed-loop stability can be obtained off-line. Lyapunov-based controller design techniques provide a natural framework for the design of a stabilizing “fall-back” controller for which an explicit characterization of the region of closed-loop stability can be obtained.

3.4 Lyapunov-based Bounded Control

In this section, we initially assume that measurements of all the state variables are available and design a state feedback bounded controller using Lyapunov techniques and provide an explicit characterization of the stability region of the resulting closed-loop system. Then, we couple the state feedback controller with the state observer of Eq.3.4 to design an output feedback controller and characterize the stability properties of the resulting closed-loop system including the set of initial conditions for which closed-loop stability is guaranteed.

3.4.1 State feedback controller design

Consider the Lyapunov function candidate $V = x'Px$, where P is a positive-definite symmetric matrix that satisfies the Riccati equation:

$$A'P + PA - PBB'P = -\bar{Q} \quad (3.11)$$

for some positive-definite matrix $\bar{Q}$. Using this Lyapunov function, we can construct, using a modification of Sontag's formula for bounded controls proposed in [127] (see also [69]), the following bounded nonlinear controller:

$$u(x) = -2k(x)B'Px := b(x) \quad (3.12)$$

where

$$k(x) = \left(\frac{L_f^*V + \sqrt{(L_f^*V)^2 + (u_{max}\|(L_gV)'\|)^4}}{\|(L_gV)'\|^2 \left[1 + \sqrt{1 + (u_{max}\|(L_gV)'\|)^2}\right]} \right) \quad (3.13)$$

with $L_f^*V = x'(A'P + PA)x + \rho x'Px$, $(L_gV)' = 2B'Px$ and $\rho > 0$. This controller is continuous everywhere in the state-space and smooth away from the origin. Using a Lyapunov argument, one can show that whenever the closed-loop state trajectory evolves within the state-space region described by the set:

$$\Phi(u_{max}) = \{x \in \mathbb{R}^n : L_f^*V < u_{max}\|(L_gV)'\|\} \quad (3.14)$$

the resulting control action respects the constraints (i.e., $\|u\| \leq u_{max}$) and enforces, simultaneously, the negative-definiteness of the time-derivative of V along the trajectories of the closed-loop system. Note that the size of the set Φ depends on the magnitude of the constraints in a way such that the tighter the constraints, the smaller the region described by this set. Starting from any initial state within $\Phi(u_{max})$, asymptotic stability of the constrained closed-loop system can be guaranteed, provided that the closed-loop trajectory remains within the region described by $\Phi(u_{max})$. To ensure this, we consider initial conditions that belong to an invariant subset, preferably the largest, which we denote by $\Omega(u_{max})$. A similar idea was used in [66, 69] in the context of bounded robust control of constrained nonlinear systems. One way of constructing such a subset is using the level sets of V (see chapter 4 in [112] for details), i.e.,

$$\Omega(u_{max}) = \{x \in \mathbb{R}^n : x'Px \leq c_{max}\} \quad (3.15)$$

where $c_{max} > 0$ is the largest number for which all nonzero elements of $\Omega(u_{max})$ are contained within $\Phi(u_{max})$. The invariant region described by the set $\Omega(u_{max})$ provides

an estimate of the stability region, starting from where the origin of the constrained closed-loop system under the control law of Eqs.3.12–3.13 is guaranteed to be asymptotically stable. To simplify notation, we will suppress the dependence of the sets Ω and Φ on u_{max} in the remainder of the chapter.

Remark 3.5: The stability region under a given stabilizing control law is typically only a subset of the null controllable region (the exact computation of which remains an open research problem) and the bounded controller of Eqs.3.12–3.13 is only one choice, out of many others, that provides an explicit characterization of the stability region. Furthermore, while the level sets of V provide only an estimate of the stability region under the bounded controller, less conservative estimates that capture larger portions of the null controllable region can be obtained using, for example, a combination of several Lyapunov functions (for example, see [97, 73]). In general, well-tuned bounded control laws provide larger estimates of the stability region than controllers designed without taking the constraints into account (see [66] for further details on this issue).

Remark 3.6: Referring to the optimality properties of the bounded controller, it should be noted that, within a well-defined subset of Ω , this class of controllers can be shown to be inverse optimal with respect to some meaningful, yet unspecified *a priori*, cost functional that imposes meaningful penalties on the state and control action (see [66] for a proof and further details). However, unless this cost functional (which is determined only after controller design) coincides with the actual cost functional considered by the control system designer (this conclusion cannot be ascertained *a priori*), the performance of the bounded controller will not be optimal with respect to the actual cost functional. An explicit quantitative characterization of the cost incurred by implementing the bounded controller in this case is difficult to obtain. Furthermore, given an arbitrary cost functional, there is no transparent way of “tuning” the bounded controller to achieve optimality since, by design, the goal of the bounded controller is to achieve stabilization for an explicitly-defined (*a priori*) set of initial conditions.

Remark 3.7: When comparing the optimality properties of the bounded and predictive controllers, it is important to realize that, in general, it is possible that the total cost of steering the system from the initial state to the origin under the bounded controller could be less than the total cost incurred under MPC. Keep in mind that MPC formulations typically consider minimizing a given cost over some finite horizon. Therefore, for a given objective function of the form of Eq.3.7 for example, by solving the optimization problem once and implementing the resulting input trajectory in its entirety, the cost of taking the system from the initial state to the state at the end of the horizon under MPC is guaranteed to be smaller than the corresponding cost for the bounded controller. However, since MPC implementation involves repeated optimizations (i.e., only the first control move of the input trajectory is implemented each time), one could in some cases end up with a higher total cost for MPC. This possibility, however, cannot be ascertained *a priori* (i.e., before implementing both controllers and computing the total cost).

3.4.2 Stability properties under output feedback bounded control

The lack of state measurements necessitates the use of a state estimator to obtain estimates of the state variables. When a state estimator of the form of Eq.3.4 is used, the resulting closed-loop system is composed of a cascade, between the error system and the plant, of the form:

$$\begin{aligned}\dot{x} &= Ax + Bu(x - e) \\ \dot{e} &= (A - LC)e\end{aligned}\tag{3.16}$$

Note that the values of the states used in the controller contain errors. The state feedback stability region, therefore, is not exactly preserved under output feedback. However, by exploiting the error dynamics of Eq.3.5, it is possible to recover arbitrarily large compact subsets of the state feedback stability region, provided that the poles of the observer are placed sufficiently far in the left half of the complex plane (which can be accomplished by choosing the observer gain parameter β sufficiently large). This

idea is consistent with earlier results on semi-global output feedback stabilization of unconstrained systems using high-gain observers (for example, see [222, 133, 39]).

By viewing the estimation error as a dynamic perturbation to the state feedback problem, the basic idea, in constructing the output feedback controller and characterizing its stability region, is to design the observer in a way that exploits the robustness of the state feedback controller with respect to bounded estimation errors. To this end, given the state feedback stability region, Ω , we initially quantify the robustness margin by deriving a bound on the estimation error, $e_m > 0$, that can be tolerated by the state feedback controller, in the sense that the closed-loop state trajectory does not escape Ω for all $\|e\| \leq e_m$. Once this bound is computed, the idea is to initialize the states and the state estimates within any subset $\Omega_b \in \Omega$ and choose a consistent observer gain matrix L (parameterized by β) to ensure that, before the states reach the boundary of Ω , the norm of the estimation error has decayed to a value less than e_m (see Figure 3.2 for a pictorial representation of this idea). These arguments are formalized in Propositions 1 and 2 below (the proofs are given in the after the propositions).

Proposition 3.1: *Consider the constrained LTI system of Eqs.3.1–3.3 under the bounded control law of Eqs.3.12–3.13 with $u = u(x - e)$, where e is the state measurement error. Then, there exists a positive real number, e_m , such that if $x(0) \in \Omega$ and $\|e(t)\| \leq e_m \forall t \geq 0$, then $x(t) \in \Omega \forall t \geq 0$.*

Proof of Proposition 3.1: The proof consists of two parts. In the first part, we establish that the bounded state feedback control law of Eqs.3.12–3.13 enforces asymptotic stability for all initial conditions within Ω . In the second part, we compute an estimate of the tolerable measurement error, e_m , which guarantees that a state trajectory starting within Ω remains within it for all $\|e\| \leq e_m$, for all $t \geq 0$.

Part 1: Substituting the state feedback control law of Eqs.3.12–3.13 into the system of Eq.3.1 and evaluating the time-derivative of the Lyapunov function along the closed-

loop trajectories, it can be shown that:

$$\begin{aligned}\dot{V}(x) &= L_f V(x) + L_g V(x) u(x) \\ &\leq \frac{-\rho x' P x}{\left[1 + \sqrt{1 + (2u_{max} \|B' P x\|)^2}\right]}\end{aligned}\quad (3.17)$$

for all $x \in \Phi$, and hence for all $x \in \Omega$, where Φ and Ω were defined in Eqs.3.14–3.15, respectively. Since the denominator term in Eq.3.17 is bounded on Ω , there exists a positive real number, ρ^* , such that

$$\dot{V} \leq -\rho^* x' P x \quad (3.18)$$

for all $x \in \Omega$, which implies that the origin of the closed-loop system, under the control law of Eqs.3.12–3.13, is asymptotically stable, with Ω as an estimate of the domain of attraction.

Part 2: In this part, we analyze the behavior of $\dot{V}$ on the boundary of Ω (i.e., the level surface described by $V(x) = c_{max}$) under bounded measurement errors, $\|e\| \leq e_m$. To this end, we have:

$$\begin{aligned}\dot{V}(x) &= L_f V(x) + L_g V(x) u(x - e) \\ &= L_f V(x) + L_g V(x) u(x) + L_g V(x) [u(x - e) - u(x)] \\ &\leq -\rho^* c_{max} + \|L_g V\| \|u(x - e) - u(x)\| \\ &\leq -\rho^* c_{max} + M \|u(x - e) - u(x)\|\end{aligned}\quad (3.19)$$

for all $x = V^{-1}(c_{max})$, where $M = \max_{V(x)=c_{max}} (\|L_g V(x)\|)$ (note that M exists since $\|L_g V(\cdot)\|$ is continuous and the maximization is considered on a closed set). Since $u(\cdot)$ is continuous, then given any positive real number r such that $\mu = \frac{\rho^* c_{max} - r}{M} > 0$, there exists $e_m > 0$ such that if $\|(x - e) - x\| = \|e\| \leq e_m$, then $\|u(x - e) - u(x)\| \leq \mu$ and, consequently,

$$\dot{V}(x) \leq -\rho^* c_{max} + M\mu = -r < 0 \quad (3.20)$$

for all $x = V^{-1}(c_{max})$. This implies that for all measurement errors such that $\|e\| \leq e_m$, we have $\dot{V} < 0$ on the boundary of Ω . Therefore, under the bounded controller,

any closed-loop state trajectory, starting within Ω , cannot escape this region, i.e., $x(t) \in \Omega \forall t \geq 0$. This completes the proof of the proposition.

Remark 3.8: The bounded control law of Eqs.3.12–3.13 will be used in the next section to illustrate the basic idea of the proposed hybrid control scheme. Our choice of using this particular design is motivated by its explicit structure and well-defined region of stability, which allows for an estimation of the robustness margin of the state feedback controller with respect to bounded measurement errors. However, the results of this chapter are not restricted to this particular choice of bounded controllers. Any other analytical bounded control law, with an explicit structure and well-defined region of robust stability with respect to bounded measurement errors, can also be used in implementing the proposed control strategy.

Proposition 3.2: *Consider the constrained LTI system of Eqs.3.1–3.3, the state observer of Eq.3.4 and the bounded control law of Eqs.3.12–3.13. Then, given any positive real number, δ_b , such that $\Omega_b = \{x \in \mathbb{R}^n : \|x\|_P^2 \leq \delta_b\} \subset \Omega$, there exists a positive real number β^* such that if $\|x(0)\|_P^2 \leq \delta_b$, $\|\hat{x}(0)\|_P^2 \leq \delta_b$ and $\beta \geq \beta^*$, the origin of the constrained closed-loop system is asymptotically stable.*

Proof of Proposition 3.2: The proof is divided into two parts. In the first part, we show that given $\delta_b > 0$ (the size of the output feedback stability region) such that $\Omega_b \subset \Omega$, there exists a choice of β such that the closed-loop trajectories are bounded for all $x(0)$ and $\hat{x}(0)$ belonging in Ω_b . In the second part, we use a Lyapunov argument, together with boundedness of the states of the closed-loop system and of the estimation error, to show that the state is ultimately bounded with a bound that depends on the norm of the estimation error; and, therefore, converges to zero as the error tends to zero.

Part 1: From Eq.3.5, we have that the error dynamics are given by $\dot{e} = (A - LC)e$, where $A - LC$ is Hurwitz, and all the eigenvalues of the matrix $A - LC$ satisfy $\lambda \leq -\beta$. It then follows that an estimate of the form $\|e(t)\| \leq \kappa(\beta)\|e(0)\|\exp(-\beta t)$

holds for some $\kappa(\beta) > 0$, for all $t \geq 0$. Given any positive real number, δ_b , such that $\Omega_b = \{x \in \mathbb{R}^n : \|x\|_p^2 \leq \delta_b\} \subset \Omega$, let $T_{min} = \min\{t \geq 0 : V(x(0)) = \delta_b, V(x(t)) = c_{max}, u(t) \in \mathcal{U}\}$ (i.e., T_{min} is the shortest time during which the closed-loop state trajectory can reach the boundary of Ω starting from the boundary of Ω_b using *any* admissible control action). Furthermore, let $e_{max}(0) = \max_{x, \hat{x} \in \Omega_b} \|x(0) - \hat{x}(0)\|$ ($e_{max}(0)$ therefore is the largest possible initial error given that both the states and state estimates are initialized within Ω_b). Choose T_d such that $0 < T_d < T_{min}$ and let β^* be such that $e_m \leq \kappa(\beta^*)e_{max}(0)\exp(-\beta^*T_d)$ (the existence of such a β^* follows from the fact that $\kappa(\beta)$ is polynomial in β). For any choice of $\beta \geq \beta^*$, therefore, it follows that $\|e(T_{min})\| \leq e_m$ (since $\|e(T_d)\| \leq e_m$, $T_{min} \geq T_d$ and the bound on the norm of the estimation error decreases monotonically with time for all $t \geq 0$). This implies that the norm of the estimation error decays to a value less than e_m before the closed-loop state trajectory, starting within Ω_b , could reach the boundary of Ω . It then follows from Proposition 3.1 that the closed-loop state trajectory cannot escape Ω for all $t \geq 0$, i.e., the trajectories are bounded and $\|x(t)\|_p^2 \leq c_{max} \forall t \geq 0$.

Part 2: To prove asymptotic stability, we note that for all $x \in \Omega$:

$$\begin{aligned} \dot{V}(x) &= L_f V(x) + L_g V(x)u(x - e) \\ &= L_f V(x) + L_g V(x)u(x) + \|L_g V(x)\| \|u(x - e) - u(x)\| \\ &\leq -\rho^* \|x\|_p^2 + M \|u(x - e) - u(x)\| \end{aligned} \quad (3.21)$$

The term $\|u(x - e) - u(x)\|$ is continuous and vanishes when $e = 0$. Therefore, since both x and e are bounded, there exists a positive real number φ such that $\|u(x - e) - u(x)\| \leq \varphi \|e\|$ for all $\|x\|_p^2 \leq c_{max}$, $\|e\| \leq e_m$. Substituting this estimate into Eq.3.21 yields:

$$\begin{aligned} \dot{V}(x) &\leq -\rho^* \|x\|_p^2 + M\varphi \|e\| \\ &\leq -\frac{\rho^*}{2} \|x\|_p^2 \quad \forall \|x\|_p \geq \sqrt{\frac{2M\varphi \|e\|}{\rho^*}} := \gamma_1(\|e\|) \end{aligned} \quad (3.22)$$

where $\gamma_1(\cdot)$ is a class $\mathcal{K}$ function (a continuous function $\alpha(\cdot)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$; see also [112]). The above inequality implies that $\dot{V}$ is negative outside some residual set whose size depends on $\|e\|$. Using the

result of Theorem 5.1-Corollary 5.2 in [112], this implies that, for any $x(0) \in \Omega_b$, there exists a class $\mathcal{KL}$ function $\bar{\beta}(\cdot, \cdot)$ and a class $\mathcal{K}$ function $\gamma_2(\cdot)$, such that

$$\|x(t)\| \leq \bar{\beta}(\|x(0)\|, t) + \gamma_2(\sup_{\tau \geq 0} \|e(\tau)\|) \quad \forall t \geq 0 \quad (3.23)$$

implying that the x subsystem of Eq.3.16, with e as input, is input-to-state stable (recall, from Proposition 3.1, that $x(t) \in \Omega_b \quad \forall t \geq 0$). Noting also that the e subsystem of Eq.3.16 is asymptotically stable ($\lim_{t \rightarrow \infty} \|e(t)\| = 0$), and using Lemma 5.6 in [112], we get that the interconnected system of Eq.3.16 is asymptotically stable. This completes the proof of the proposition.

Remark 3.9: Since the only assumption on C is that the pair (C, A) is observable, the observer states may “peak” before they converge to the true state values. This, however, does not pose a problem in our design because: (a) the physical constraints on the manipulated input eliminate the occurrence of instability due to peaking of the state estimates (i.e., they prevent transmission of peaking to the process), and (b) by “stepping back” from the boundary of the state feedback stability region and choosing an appropriate value for β , the design ensures that the system states cannot leave the stability region of the bounded controller before the estimation errors have gone below the permissible value.

Remark 3.10: In principle, the stability region under output feedback, Ω_b , can be chosen as close as desired to Ω by increasing the observer gain parameter β . However, it is well-known that large observer gains can amplify measurement noise and induce poor performance (see the simulation studies section for how this issue is addressed in observer implementation). This points to a fundamental tradeoff that cannot be resolved by simply changing the estimation scheme. For example, while one could replace the high-gain observer design with other observer designs (for example, a moving horizon estimator) to get a better handle on measurement noise, it is difficult in such schemes to obtain an explicit relationship between the observer tuning parameters and the output feedback stability region.

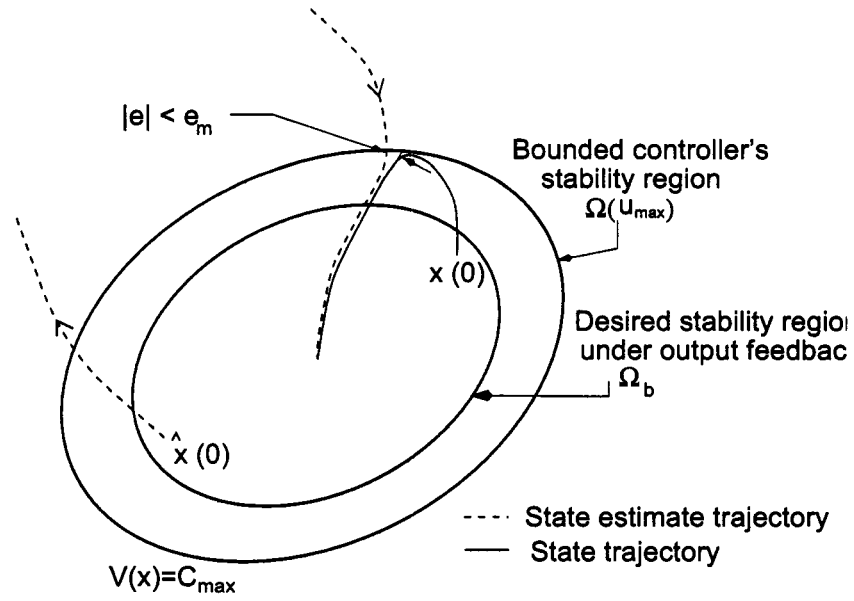


Figure 3.2: Schematic representation of the evolution of the trajectories of the closed-loop state and state estimate. Starting from within Ω_b , the observer gain ensures that, before the state of the closed-loop system reaches the boundary of Ω , the state estimate has converged close enough to the true value of the state, so that the norm of the estimation error is below the allowable error tolerated by the controller within Ω .

Remark 3.11: Note that, for the bounded controller, initializing x and $\hat{x}$ within Ω_b guarantees that x cannot escape Ω because of the inherent robustness of the bounded controller design with respect to bounded estimation errors (see the proof of Proposition 3.2). For MPC, however, there is no guarantee that x will stay in Ω , even if both x and $\hat{x}$ are initialized within Ω_b , and, therefore, whenever MPC is used, it will be necessary to rely on the evolution of the estimate, $\hat{x}(t)$, to deduce whether $x(t)$ is or is not within Ω . As we discuss in greater detail in the next section, the ability to reliably deduce the position of the true state at any given time is essential to the ability of the bounded controller to “step in” to preserve closed-loop stability in the event of failure of MPC (see Remark 3.14). One way of ascertaining bounds on the position of the true states, by looking only at the values of the estimates, is described in Proposition 3.3 below. The proof of this proposition is given below.

Proposition 3.3: *Consider the constrained system of Eqs.3.1–3.3, the state observer of Eq.3.4 and the bounded control law of Eqs.3.12–3.13. Let $T_d^* := \frac{1}{\beta} \ln \left(\frac{\kappa(\beta)e_{max}(0)}{\epsilon \sqrt{c_{max}/\lambda_{max}(P)}} \right)$ for some $0 < \epsilon < 1$ where $e_{max}(0) = \max_{\hat{x}(0), x(0) \in \Omega_b} \|\hat{x}(0) - x(0)\|$. Then, there exists a positive real number $\delta_s^* < c_{max}$ such that for all $\delta_s \leq \delta_s^*$, and for all $t \geq T_d^*$, $\hat{x}'(t)P\hat{x}(t) \leq \delta_s \implies x'(t)Px(t) \leq c_{max}$.*

Proof of Proposition 3.3: Since the error dynamics obey a bound of the form $\|e(t)\| \leq \kappa(\beta)e_{max}(0) \exp(-\beta t)$, substituting T_d^* into this expression yields $\|e(T_d^*)\| := e^* \leq \epsilon \sqrt{c_{max}/\lambda_{max}(P)}$. Then, for all $t \geq T_d^*$, if $\hat{x}'(t)P\hat{x}(t) \leq \delta_s$ for some $\delta_s > 0$, we can write:

$$\begin{aligned}
x'(t)Px(t) &= (\hat{x}(t) + e(t))'P(\hat{x}(t) + e(t)) \\
&= \hat{x}'(t)P\hat{x}(t) + 2\hat{x}'(t)Pe(t) + e'(t)Pe(t) \\
&\leq \delta_s + 2\|\hat{x}(t)\|\|e(t)\| + \|e(t)\|_p^2 \\
&\leq \delta_s + 2\sqrt{\frac{\lambda_{max}(P^2)\delta_s}{\lambda_{min}(P)}} e^* + \lambda_{max}(P)e^{*2} \\
&:= f(\delta_s)
\end{aligned} \tag{3.24}$$

Note that $f(\cdot)$ is a continuous, monotonically increasing function of $\delta_s \geq 0$, with $f(0) = \epsilon^2 c_{max} < c_{max}$ and $f(c_{max}) > c_{max}$. This implies that there exists $0 < \delta_s^* < c_{max}$ such that, for all $\delta_s \leq \delta_s^*$, the relation $f(\delta_s) \leq c_{max}$ holds, i.e., $x'(t)Px(t) \leq c_{max}$. This completes the proof of the proposition.

Remark 3.12: The above proposition establishes the existence of a set, $\Omega_s := \{x \in \mathbb{R}^m : x'Px \leq \delta_s\}$, that allows us to ascertain bounds on the position of the true state, by looking only at whether the estimate is within this set. Specifically, for a given bound on the norm of the estimation error, at any given time, if the state estimate is within Ω_s , then we can conclude with certainty that the true state cannot be outside of Ω , irrespective of the controller being used (see the proof after the proposition for the mathematical details of this argument). Note also that the size of Ω_s depends on the size of the error, and, therefore, as time evolves and the estimation error keeps decreasing, the size of Ω_s approaches that of Ω .

3.5 Hybrid Predictive Control

While the bounded controller possesses a well-defined region of initial conditions that guarantee closed-loop stability in the presence of constraints, the performance of this controller may not be optimal with respect to a given performance criterion. On the other hand, the model predictive controller is well-suited for handling constraints within an optimal control setting; however, the analytical characterization of its set of initial conditions, for which closed-loop stability is guaranteed, is a more difficult task than it is through bounded control. The lack of state measurements introduces an additional layer of complexity in implementing the controllers designed under the assumption of state feedback. In this section, we show how to reconcile the two control approaches by means of a hybrid control structure that combines the desirable properties of both approaches.

To this end, we consider the LTI system of Eqs.3.1–3.3, for which the bounded

controller of Eqs.3.12–3.13, the state estimator of Eq.3.4 and the model predictive controller of Eqs.3.9–3.10 have been designed. The control problem is formulated as the one of designing a set of switching laws that orchestrate the transition between the predictive controller and the bounded controller under output feedback in a way that: (1) respects input constraints, (2) guarantees asymptotic stability of the origin of the closed-loop system starting from any initial condition within arbitrarily large compact subsets of the state feedback stability region of the bounded controller, and (3) accommodates the optimality requirements of MPC whenever possible (i.e., when closed-loop stability is guaranteed). For a precise statement of the problem, we first cast the system of Eqs.3.1–3.3 as a switched linear system of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu_{i(t)} \\ y &= Cx \\ \|u_i\| &\leq u_{max} \\ i(t) &\in \{1, 2\} \end{aligned} \tag{3.25}$$

where $i : [0, \infty) \rightarrow \{1, 2\}$ is the switching signal which is assumed to be a piecewise continuous (from the right) function of time, implying that only a finite number of switches is allowed on any finite interval of time. The index $i(t)$, which takes values in the finite set $\{1, 2\}$, represents a discrete state that indexes the control input, $u(\cdot)$, with the understanding that $i(t) = 1$ if and only if $u_i(\hat{x}(t)) = b(\hat{x}(t))$ (i.e., the bounded controller is implemented in the closed-loop system), and $i(t) = 2$ if and only if $u_i(\hat{x}(t)) = M_s(\hat{x}(t))$ (i.e., MPC is implemented in the closed-loop system). Our goal, is to construct a switching law, based on the available state estimates:

$$i(t) = \psi(\hat{x}(t), t) \tag{3.26}$$

that provides the set of switching times that ensure asymptotically stabilizing transitions between the predictive and bounded controllers, in the event that the predictive controller is unable to enforce closed-loop stability for a given initial condition within Ω . In the remainder of this section, we present a switching scheme that addresses this problem. Theorem 3.1 below summarizes the main result (the proof is given below).

Theorem 3.1: Consider the constrained system of Eq.3.25, the bounded controller of Eqs.3.12–3.13, the state estimator of Eq.3.4, and the MPC law of Eqs.3.9–3.10. Let $x(0) \in \Omega_b$, $\hat{x}(0) \in \Omega_b$, $\beta \geq \beta^*$, $\delta_s \leq \delta_s^*$, $\Omega_s(T_d^*) = \{x \in \mathbb{R}^n : x'Px \leq \delta_s(T_d^*)\}$, and $0 < T_d < T_{\min} := \min\{t \geq 0 : V(x(0)) = \delta_b, V(x(t)) = c_{\max}, u(t) \in \mathcal{U}\}$, where Ω_b , β^* were defined in Proposition 3.2, and δ_s^* , T_d^* were defined in Proposition 3.3. Let $T_m \geq \max\{T_d, T_d^*\}$ be the earliest time for which $\hat{x}(T_m) \in \Omega_s$ and the MPC law prescribes a feasible solution, $M_s(\hat{x}(T_m))$. Also, let $T_f > T_m$ be the earliest time for which $\dot{V}(\hat{x}(T_f)) \geq 0$. Then, the following switching rule:

$$i(t) = \begin{cases} 1, & 0 \leq t < T_m \\ 2, & T_m \leq t < T_f \\ 1, & t \geq T_f \end{cases} \quad (3.27)$$

where $i(t) = 1$ if and only if $u_i(\hat{x}(t)) = b(\hat{x}(t))$ and $i(t) = 2$ if and only if $u_i(\hat{x}(t)) = M_s(\hat{x}(t))$, asymptotically stabilizes the origin of the closed-loop system.

Proof of Theorem 3.1: Given $x(0) \in \Omega_b$, $\hat{x}(0) \in \Omega_b$, $\beta \geq \beta^*$ (see Proposition 3.2), and $\delta_s \leq \delta_s^*$ (see Proposition 3.3), we consider the following cases:

Case 1: Consider first the case when $T_m = \infty$ (i.e., MPC is never feasible). Then, we have from Eq.3.27 that $i(t) = 1$ and $u(\hat{x}(t)) = b(\hat{x}(t))$ for all $t \geq 0$. It then follows from Proposition 3.2 that the origin of the closed-loop system is asymptotically stable.

Case 2: Now, consider the case when $T_m < \infty$ and $T_f = \infty$. Since $T_m \geq T_d$ and $\|e(T_d)\| \leq e_m$ (see part 1 of the proof of Proposition 3.2), we have that $\|e(T_m)\| \leq e_m$. Since only the bounded controller is implemented, i.e. $i(t) = 1$, for $0 \leq t < T_m$, it follows from Proposition 3.1 that $x(t) \in \Omega$ for all $0 \leq t < T_m$ (or that $x(T_m^-) \in \Omega$). This fact, together with the continuity of the solution of the switched closed-loop system – which follows from the fact that the right hand side of Eq.3.25 is continuous in x and piecewise continuous in time since only a finite number of switches is allowed over any finite time interval – implies that, upon switching (instantaneously) to the model predictive controller at $t = T_m$, we have $x(T_m) \in \Omega$. Since $T_f = \infty$, then from the

definition of T_f in Theorem 3.1 it follows that $u(\hat{x}(t)) = M_s(\hat{x}(t))$ and $\dot{V}(\hat{x}(t)) < 0$ for all $t \geq T_m$. This implies that $\hat{x}(t) \in \Omega_s$ for all $t \geq T_m$ and, consequently from the definition of T_m (note that $T_m \geq T_d^*$ where T_d^* was defined in Proposition 3.3), that $x(t) \in \Omega$ for all $t \geq T_m$ (i.e., the closed-loop state trajectory is bounded). Furthermore, since $\hat{x}(t) = x(t) - e(t)$ and $\lim_{t \rightarrow \infty} \|e(t)\| = 0$, it follows that $\lim_{t \rightarrow \infty} \hat{x}'(t)P\hat{x}(t) = 0$; and, hence, $\lim_{t \rightarrow \infty} x'(t)Px(t) = 0$. The origin of the switched closed-loop system is therefore asymptotically stable.

Case 3: Finally, consider the case when $T_m < \infty$ and $T_f < \infty$. From the analysis in case 2 above, we have that $x(t) \in \Omega \quad \forall 0 \leq t < T_f$ (or that $x(T_f^-) \in \Omega$). This fact, together with the continuity of the solution of the switched closed-loop system, implies that, upon switching (instantaneously) to the bounded controller at $t = T_f$, we have $x(T_f) \in \Omega$ and $u(t) = b(\hat{x}(t))$ for all $t \geq T_f$. We also have $\|e(t)\| \leq e_m$ for all $t \geq T_f$ since $T_f > T_d$. Therefore, from Proposition 3.1 it is guaranteed that $x(t) \in \Omega$ for all $t \geq T_f$. Finally, following the same arguments presented in part 2 of the proof of Proposition 3.2, it can be shown that $\|x(t)\| \rightarrow 0$ as $t \rightarrow \infty$, which together with boundedness of the state trajectory, establishes asymptotic stability of the origin of the closed-loop system. This completes the proof of the theorem.

Remark 3.13: Figure 3.3 depicts a schematic representation of the hybrid predictive control structure proposed in Theorem 3.1. The four main components of this structure include the state estimator, the bounded controller, the predictive controller, and a high-level supervisor that orchestrates the switching between the two controllers. The implementation of the control strategy is best understood through the following stepwise procedure (see Figure 3.4):

1. Given the system of Eq.3.1 and the constraints of Eq.3.3, design the bounded controller using Eqs.3.12–3.13. Given the performance objective, design the predictive controller.
2. Compute the stability region estimate for the bounded controller under state

feedback, Ω , using Eqs.3.14–3.15 and, for a choice of the output feedback stability region, $\Omega_b \subset \Omega$, compute e_m , β and T_d defined in Propositions 1 and 2.

3. Compute the region Ω_s and T_d^* (see Proposition 3.3) which ensure that $\hat{x} \in \Omega_s \Rightarrow x \in \Omega$ for all times greater than T_d^* .
4. Initialize the closed-loop system at any initial condition, $x(0)$, within Ω_b , under the bounded controller using an initial guess for the state, $\hat{x}(0)$, within Ω_b . Keep the bounded controller active for a period of time, $[0, T_m)$.
5. At $t = T_m$ (by which time $\hat{x} \in \Omega_s$), test the feasibility of MPC using values of the estimates generated by the state observer. If MPC yields no feasible solution, keep implementing the bounded controller.
6. If MPC is feasible, disengage the bounded controller from the closed-loop system and implement the predictive controller instead. Keep MPC active for as long as $\hat{x} \in \Omega_s$ and $V(\hat{x})$ keeps decaying.
7. At the first instance that either $V(\hat{x})$ begins to increase under MPC, or MPC runs into infeasibilities, switch back to the bounded controller and implement it for all future times; else keep the predictive controller active.

Remark 3.14: An important feature that distinguishes the switching logic of Theorem 3.1 from the state feedback logic in [75] is the fact that, in the output feedback case, MPC implementation does not commence until some period of time, $[0, T_m)$, has elapsed, even if MPC is initially feasible. The rationale for this delay (during which only bounded control is implemented) is the need to ensure that, by the time MPC is implemented, the norm of the estimation error has already decayed to sufficiently low values that would allow the bounded controller to preserve closed-loop stability in the event that MPC needs to be switched out after its implementation. More specifically, before MPC can be safely activated, the supervisor needs to make sure that: (1) the

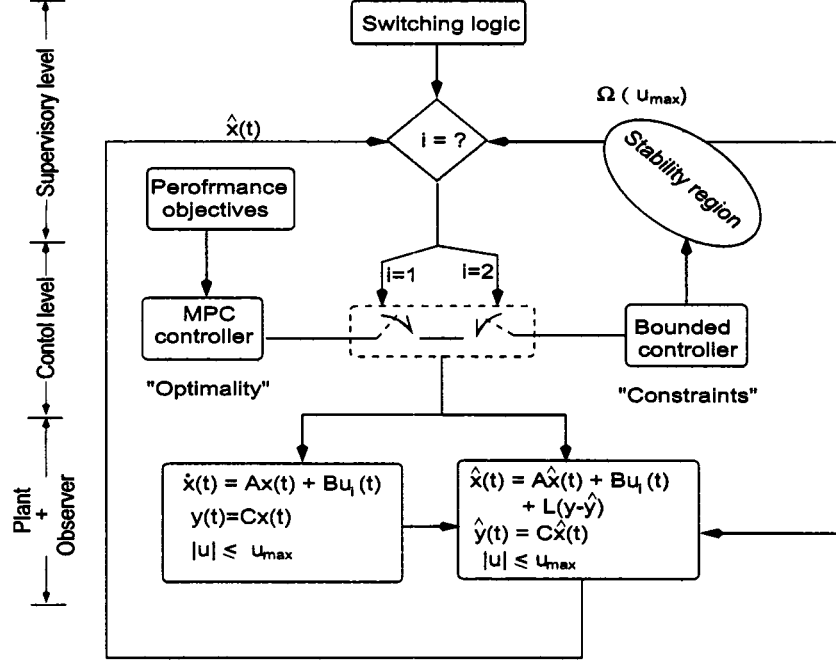


Figure 3.3: Schematic representation of the hierarchical hybrid predictive control structure merging the bounded and model predictive controllers and a state observer.

estimation error has decayed to levels below e_m , and (2) the state, x , will be contained within Ω at any time that MPC could be switched out after its implementation. The first requirement ensures that the bounded controller, when (and if) reactivated, is able to tolerate the estimation error and still be stabilizing within Ω (see Proposition 3.1). This requirement also implies that T_m must be greater than T_d , which is the time beyond which the observer design guarantees $\|e\| \leq e_m$ (see part 1 of the proof of Proposition 3.2). The second requirement, on the other hand, implies that MPC cannot be safely switched in before the estimate enters Ω_s , and the norm of the error is such that $\hat{x} \in \Omega_s \implies x \in \Omega$ (see Remarks 11–12). Recall from Proposition 3.3 that this property holds for all $t \geq T_d^*$; hence T_m must also be greater than T_d^* , i.e., $T_m \geq \max\{T_d, T_d^*\}$. For times greater than T_m , the implementation of MPC can be safely tested without fear of closed-loop instability, since even if $\hat{x}$ tries to escape Ω_s (possibly indicating that x under MPC is about to escape Ω), the switching rule allows

the supervisor to switch back immediately to the bounded controller (before $\hat{x}$ leaves Ω_s), thus guaranteeing that at the switching time $\hat{x} \in \Omega_s$, $x \in \Omega$ and $\|e\| \leq e_m$, which altogether imply that the bounded controller, once switched in, will be asymptotically stabilizing (see the proof of Theorem 3.1 for the mathematical details of this argument).

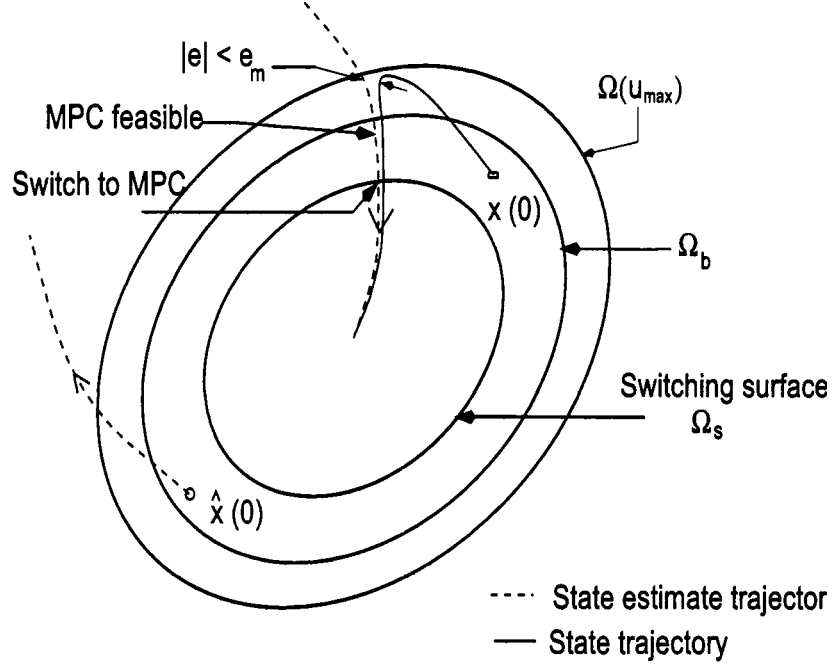


Figure 3.4: Schematic representation of the evolution of the closed-loop system under the controller switching strategy proposed in Theorem 3.1. Note that MPC is not activated (even if feasible) before $\hat{x}$ enters Ω_s .

Remark 3.15: The proposed approach does not require or provide any information as to whether the model predictive controller itself is asymptotically stabilizing or not, starting from any initial condition within Ω_b . In other words, the approach does not turn Ω_b into a stability region for MPC. What the approach does, however, is turn Ω_b into a stability region for the switched closed-loop system. The value of this can be understood in light of the difficulty in obtaining, *a priori*, an analytical characterization of the set of admissible initial conditions that the model predictive controller can steer to the origin in the presence of input constraints. Given this difficulty, by

embedding MPC implementation within Ω_b and using the bounded controller as a fall-back controller, the switching scheme of Theorem 3.1 allows us to safely initialize the closed-loop system anywhere within Ω_b with the guarantee that the bounded controller can always intervene to preserve closed-loop stability in case the MPC is infeasible or destabilizing (due, for example, to a poor choice of the initial condition and/or improper tuning of the horizon length). This safety feature distinguishes the bounded controller from other fall-back controllers that could be used, such as PID controllers, which do not provide *a priori* knowledge of the constrained stability region and, therefore, do not guarantee a safe transition in the case of unstable plants. For these controllers, a safe transition is critically dependent on issues such as whether or not the operator is able to properly tune the controller on-line, which, if not achieved, can possibly result in instability and/or performance degradation. The transition from the MPC to the bounded controller, on the other hand, is not fraught with such uncertainty and is always safe because of the *a priori* knowledge of the stability region. This aspect helps make plant operation safer and smoother.

Remark 3.16: Note that, once MPC is switched in, if $V(\hat{x})$ continues to decrease monotonically, then the predictive controller will be implemented for all $t \geq T_m$. In this case, the optimal performance of the predictive controller is practically recovered. Note also, that in this approach, the state-feedback predictive controller design is not required to be robust with respect to state measurement errors (see [88] for examples when MPC is not robust) because even if it is not robust, closed-loop stability can always be guaranteed by switching back to the bounded controller (within its associated stability region) which provides the desired robustness with respect to the measurement errors.

Remark 3.17: Even though the result of Theorem 3.1 was derived using the MPC formulation of Eqs.3.9–3.10, it is important to point out that the same result applies to MPC formulations that employ different types of stability constraints (see Remark 3.3).

The basic commonality between the formulation of Eqs.3.9–3.10 and other stability-handling formulations is the fact that, in all of these formulations, the implementation of MPC depends on whether the optimization problem, subject to the stability constraints, is feasible or not. This issue of feasibility is therefore accommodated in the switching logic by requiring the supervisor to check the feasibility of MPC in addition to monitoring the evolution of V . On the other hand, when a conventional MPC formulation (i.e., one with no stability constraints) is used, potential infeasibility of the optimization problem is no longer a concern, and, therefore, in this case the supervisory switching logic rests only on monitoring the evolution of V .

Remark 3.18: In conventional MPC formulations (with no stability constraints), the objective function is typically a tuning parameter that can be manipulated, by arbitrarily changing the weighting matrices, to ensure stability. In the hybrid predictive control framework, on the other hand, since closed-loop stability is guaranteed by the bounded controller, the objective function and the weights, instead of being arbitrarily tuned to guarantee stability, can be chosen to reflect actual physical costs and satisfy desirable performance objectives (for example, fast closed-loop response).

Remark 3.19: The switching strategy of Theorem 3.1 can be generalized to allow for multiple switchings between MPC and the bounded controller. According to Theorem 3.1, once MPC is terminated, the bounded controller is implemented for all future times. In a more general scheme, however, the bounded controller need not stay in the closed-loop system for all future times. Instead, it can be employed only until it brings the closed-loop state trajectory sufficiently closer to the origin (for example, to a pre-determined inner level set of V) at which point MPC is re-activated (if feasible) and its behavior monitored once again. Potential chattering problems due to undesirable back-and-forth switching, are eliminated by allowing only a finite number of switches between the two controllers over any finite time-interval. This scheme offers the possibility of further enhancement in the closed-loop performance (over that

obtained from the bounded controller) by implementing MPC for longer periods of time (as permitted by stability considerations).

Remark 3.20: The hybrid predictive control structure proposed in this chapter can be further extended to deal with the case when both input and state constraints are present. In one possible extension, state constraints would be incorporated directly as part of the constrained optimization problem that yields the model predictive control law. In addition, an estimate of the stability region for the bounded controller would be obtained by intersecting the region described by Eq.3.14 with the region described by the state constraints (presumed closed, convex and containing the origin in its interior) and computing the largest invariant subset within the intersection. Using this stability region (which now accounts for both input and state constraints) in place of Ω , the implementation of the hybrid predictive control strategy can proceed following the same methodology outlined in Theorem 3.1.

Remark 3.21: The proposed switching scheme differs, both in its objective and implementation, from other MPC formulations that involve switching. For example, in dual mode MPC [149], the strategy involves switching from MPC to a locally stabilizing controller once the state is brought near the origin by MPC. The purpose of switching in this approach is to relax the terminal equality constraint whose implementation is computationally burdensome for nonlinear systems. However, the set of initial conditions for which MPC is guaranteed to steer the state close to the origin is not explicitly known *a priori*. In contrast, switching from MPC to the bounded controller is used in our work only to prevent any potential closed-loop instability arising from implementing MPC without the *a priori* knowledge of the admissible initial conditions. Therefore, depending on the stability properties of the chosen MPC, switching may or may not occur, and if it occurs, it can take place near or far from the origin.

3.6 Simulation Studies

In this section, two simulation studies are presented to demonstrate the implementation, and evaluate the effectiveness, of the proposed hybrid predictive control strategy as well as test its robustness with respect to modeling errors and measurement noise.

3.6.1 Illustrative example

In this section, we demonstrate an application of the proposed hybrid predictive control strategy to a three dimensional linear system where only two of the states are measured. Specifically, we consider an exponentially unstable linear system of the form of Eqs.3.1–3.2 with

$$A = \begin{bmatrix} 0.55 & 0.15 & 0.05 \\ 0.15 & 0.40 & 0.20 \\ 0.10 & 0.15 & 0.45 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where both inputs u_1, u_2 are constrained in the interval $[-1, 1]$. Using Eqs.3.12–3.13, we initially designed the bounded controller and constructed its stability region, Ω , via Eqs.3.14–3.15. The matrix P was chosen as:

$$P = \begin{bmatrix} 6.5843 & 4.2398 & -3.830 \\ 4.2398 & 3.6091 & -2.667 \\ -3.830 & -2.667 & 2.8033 \end{bmatrix}$$

and the observer gain parameter was chosen to be $\beta = 500$ to ensure closed-loop stability for all initial conditions within a set $\Omega_b \subset \Omega$. For the model predictive controller, the parameters in the objective function of Eq.3.9 were chosen as $Q = qI$, with $q = 1$ and $R = rI$, with $r = 0.1$. We also chose a horizon length of $T = 1.5$ in implementing the predictive controller of Eqs.3.9–3.10. The resulting quadratic program was solved using the MATLAB subroutine QuadProg, and the set of ODEs integrated using the MATLAB solver ODE45.

In the first simulation run (solid lines in Figures 3.5–3.6), the states were initialized at $x_0 = [0.75 \ -0.5 \ 1.0]'$, while the observer states were initialized at $\hat{x}_0 = [0 \ 0 \ 0]'$ (which belong to the stability region of the bounded controller, Ω_b). The supervisor employs

the bounded controller, while continuously checking the feasibility of MPC. At $t = 5.45$, MPC (based on the state estimates) becomes feasible and is implemented in the closed-loop system to achieve asymptotic stability. The total cost of stabilization achieved by the switching strategy (as measured by Eq.3.9) is $J = 80.12$. The switching scheme leads to better performance when compared with the scenario where the bounded controller is implemented for all times (the cost in this case is $J = 121.07$), and also better than the model predictive controller, because, starting from this initial condition, the predictive controller prescribes no feasible control move (equivalent to an infinite value for the objective function). Note that even though feasibility of MPC could have been achieved earlier by increasing the horizon length to $T = 3.5$ (dashed lines in Figures 3.5–3.6), this conclusion could not be reached *a priori*, i.e., before running the closed-loop simulation in its entirety to check whether the choice $T = 3.5$ is appropriate. In the absence of any *a priori* knowledge of the necessary horizon length, for a given initial condition, the proposed hybrid predictive control can be used to guarantee closed-loop stability regardless of the choice of the horizon length.

In the second set of simulation results, we demonstrate the need for a choice of an observer gain consistent with the choice of Ω_b . To this end, we consider an observer design with a low gain ($\beta = 0.5$) placing the observer poles at $-0.5, -1.0, -1.5$. With the low observer gain, the estimates take a long time to converge to the true state values, resulting in the implementation of “incorrect” control action for a large period of time, by the end of which the states have already escaped out of Ω (even though the states and state estimates were initiated within Ω_b), thus resulting in closed-loop instability (dashed lines in Figures 3.7–3.8). Recall that stability was achieved when the observer gain was chosen to be $\beta = 500$ in the first set of simulation runs (the state and input profiles for this case are reproduced by the solid lines in Figures 3.7–3.8 for convenience of comparison).

To recover, as closely as possible, the state feedback stability region, large values of

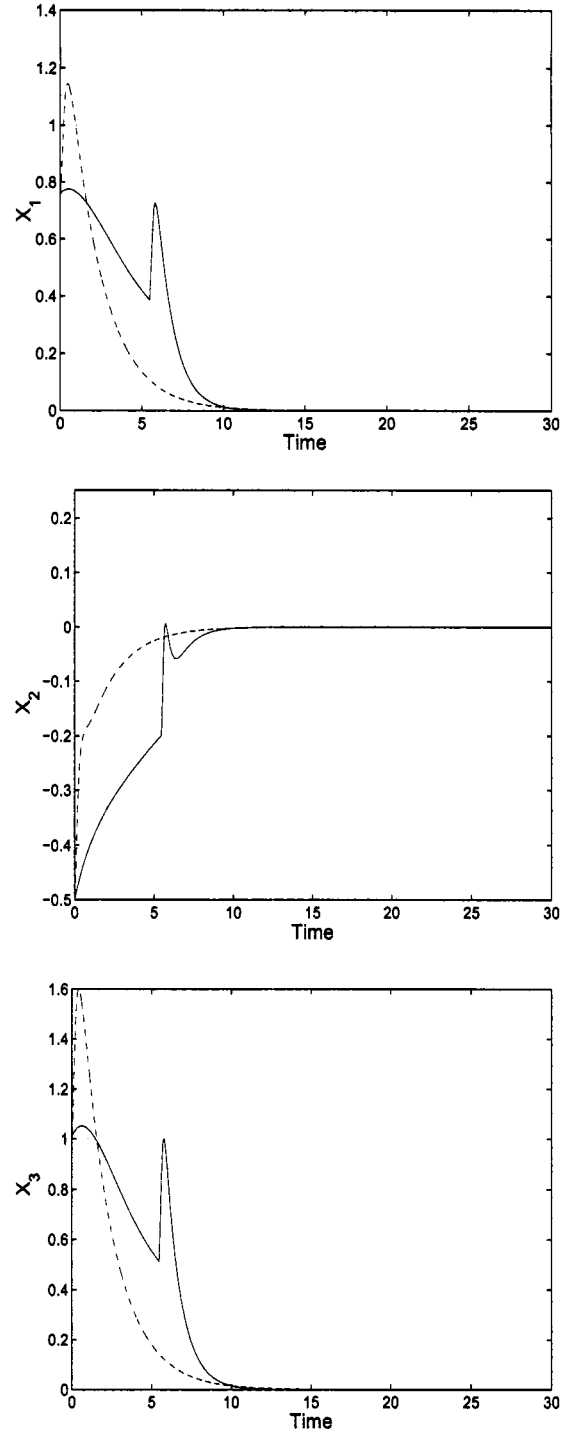


Figure 3.5: States of the closed-loop system under the proposed hybrid predictive control strategy using two different horizon lengths of $T = 1.5$ (solid) and $T = 3.5$ (dashed) in MPC implementation, and an observer gain parameter of $\beta = 500$.

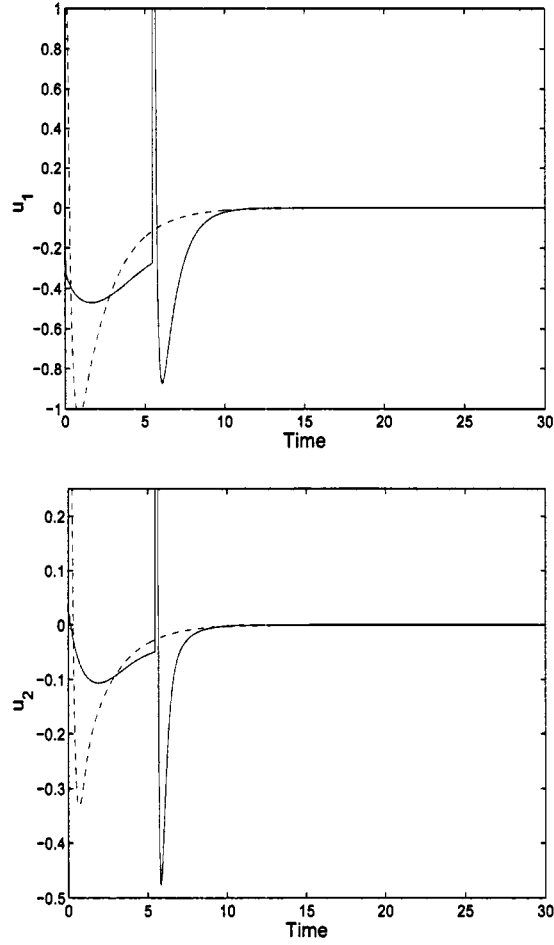


Figure 3.6: Manipulated input profiles under the proposed hybrid predictive control strategy using two different horizon lengths of $T = 1.5$ (solid) and $T = 3.5$ (dashed) in MPC implementation, and an observer gain parameter $\beta = 500$.

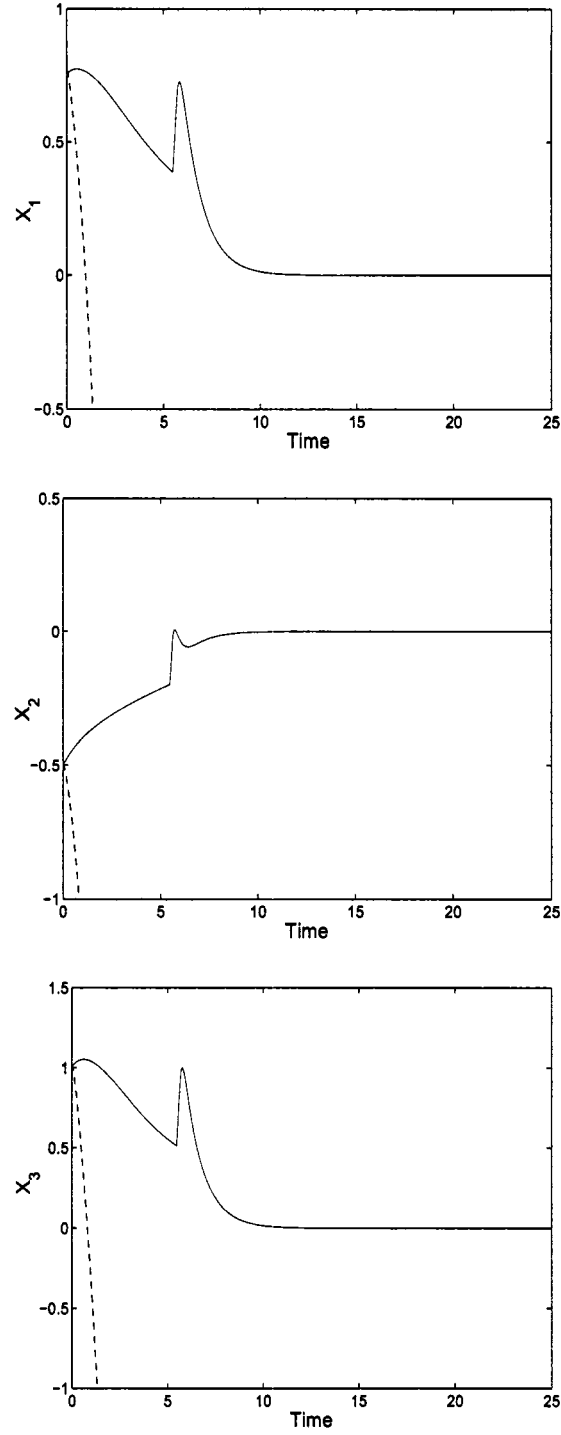


Figure 3.7: States of the closed-loop system under the proposed hybrid predictive control strategy using two different gain parameters of $\beta = 500$ (solid) and $\beta = 0.5$ (dashed) in observer implementation, and horizon length of $T = 1.5$ in the predictive controller.

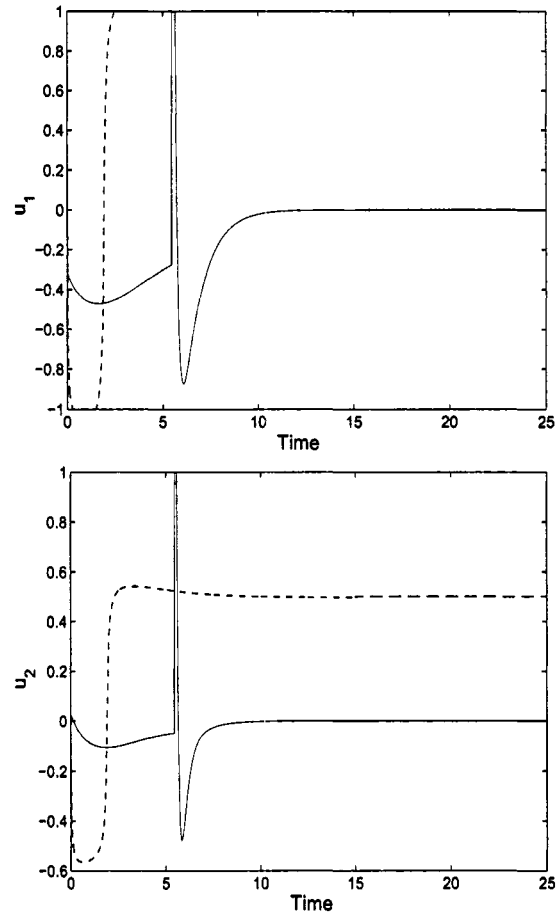


Figure 3.8: Manipulated input profiles under the proposed hybrid predictive control strategy using two different gain parameters of $\beta = 500$ (solid) and $\beta = 0.5$ (dashed) in observer implementation, and horizon length of $T = 1.5$ in the predictive controller.

the observer gain are needed. However, it is well-known that high observer gains can amplify measurement noise and induce poor performance. These observations point to a fundamental tradeoff that cannot be resolved by simply changing the estimation scheme (see Remark 3.10). For example, if the observer gain consistent with the choice of the output feedback stability region is abandoned, the noise problem may disappear, but then stability cannot be guaranteed. One approach to avoid this problem in practice is to initially use a large observer gain that ensures quick decay of the initial estimation error, and then switch to a low observer gain. In the following simulation, we demonstrate how this idea, in conjunction with switching between the controllers, can be used to mitigate the undesirable effects of measurement noise. To illustrate this point, we switch between the high and low observer gains used in the first two simulation runs and demonstrate the attenuation of noise. Specifically, we consider the nominal system described by Eqs.3.1–3.2 (see the first paragraph of this subsection for the values of A , B and C), together with model uncertainty and measurement noise. The model matrix A_m (used for controller and observer design) is assumed to be within five percent error of the system matrix A and the sensors are assumed to introduce noise in the measured outputs as $y(t) = Cx(t) + \delta(t)$ where $\delta(t)$ is a random gaussian noise with zero mean and a variance of 0.01. As seen by the solid lines in Figure 3.9, starting from the initial condition, $x_0 = [0.75 \ -0.5 \ 1.0]'$, using a high observer gain followed by a switch to a low observer gain at $t = 1.0$, and a switch from bounded control to MPC at $t = 3.5$, the supervisor is still able to preserve closed-loop stability, while at the same time resulting in a smoother control action (solid lines in Figure 3.10) when compared to the case where a high observer gain is used for the entire duration of the simulation (dotted lines in Figures 3.9–3.10).

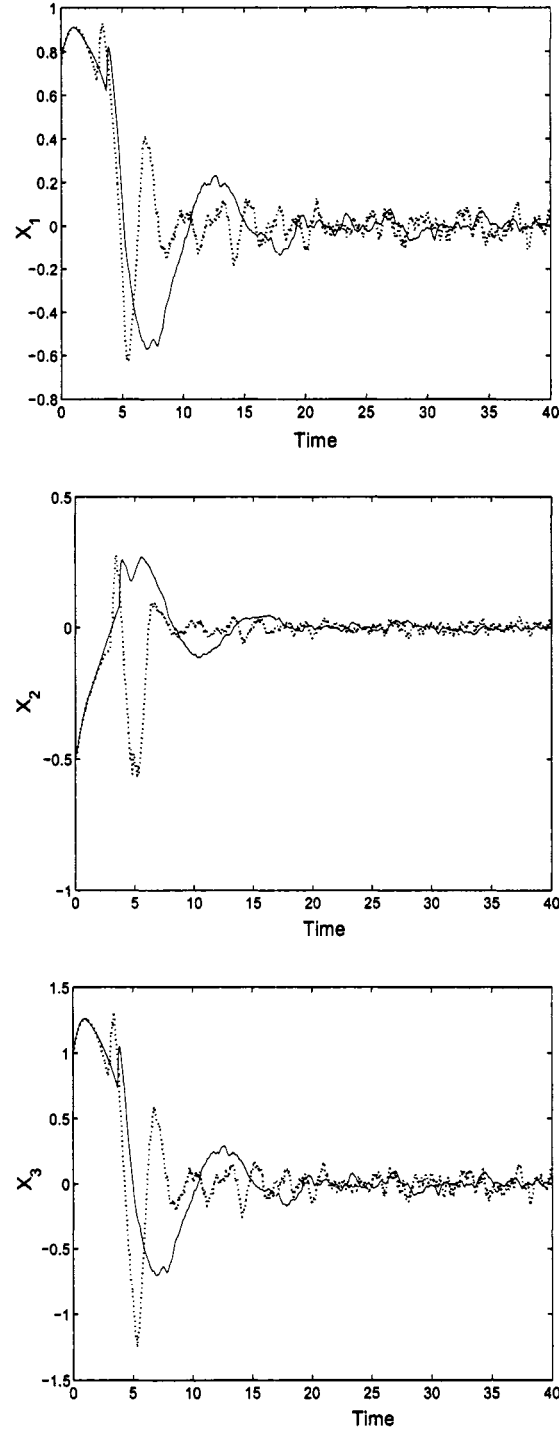


Figure 3.9: States of the closed-loop system under the proposed hybrid predictive control strategy in the presence of measurement noise and modeling errors. The solid profiles depict the case when the state observer is initially implemented using $\beta = 500$ and then switched to $\beta = 0.5$ at $t = 1.0$. The dotted profiles depict the case when the state observer is implemented using $\beta = 500$ for all times. The horizon length for the predictive controller is $T = 1.5$.

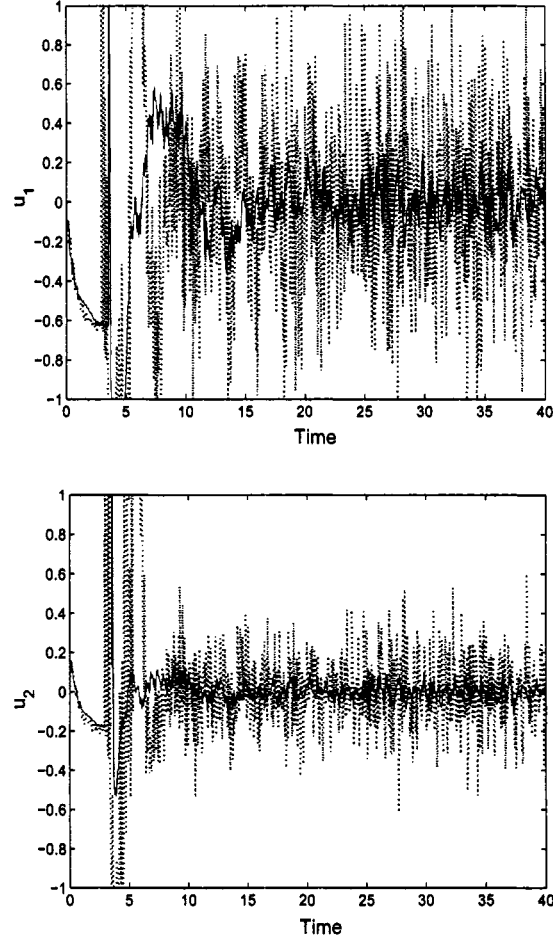


Figure 3.10: Manipulated input profiles under the proposed hybrid predictive control strategy in the presence of measurement noise and modeling errors. The solid profiles depict the case when the state observer is initially implemented using $\beta = 500$ and then switched to $\beta = 0.5$ at $t = 1.0$. The dotted profiles depict the case when the state observer is implemented using $\beta = 500$ for all times. The horizon length for the model predictive controller is $T = 1.5$.

3.6.2 Application to a chemical reactor example

In this subsection, we consider a nonlinear chemical reactor example to test the robustness of the proposed hybrid predictive control structure (designed on the basis of the linearization of the nonlinear model around the operating steady-state) with respect to modeling errors arising due to the presence of nonlinear terms. Specifically, we consider a well-mixed continuous stirred tank reactor where an irreversible elementary exothermic reaction of the form $A \xrightarrow{k} B$, takes place, where A is the reactant species and B is the product. The feed to the reactor consists of pure A at flow rate F , molar concentration C_{A0} and temperature T_{A0} . A jacket is used to remove/provide heat to the reactor. Under standard modelling assumptions, a mathematical model of the process can be derived from material and energy balances and takes the following form:

$$\begin{aligned} V \frac{dT_R}{dt} &= F(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 \exp\left(\frac{-E}{RT_R}\right) C_A V + \frac{Q}{\rho c_p} \\ V \frac{dC_A}{dt} &= F(C_{A0} - C_A) - k_0 \exp\left(\frac{-E}{RT_R}\right) C_A V \end{aligned} \quad (3.28)$$

where C_A denotes the concentrations of the species A , T_R denotes the temperature of the reactor, Q denotes rate of heat input/removal from the reactor, V denotes the volume of the reactor, ΔH , k , E denote the enthalpy, pre-exponential constant and activation energy of the reaction, c_p and ρ denote the heat capacity and density of the reactor, respectively. The values of the process parameters are given in Table 3.1. It was verified that these conditions correspond to three steady-states, one unstable (given in Table 1) and two locally asymptotically stable.

The control problem is to regulate both the outlet concentration of the reactant, C_A , and the reactor temperature, T_R , at the unstable steady-state by manipulating the inlet reactant concentration, C_{A0} , and the rate of heat input, Q , provided by the jacket. The control objective is to be accomplished in the presence of constraints on the manipulated inputs ($|Q| \leq 1$ KJ/min, $|\Delta C_{A0}| = |C_{A0} - C_{A0s}| \leq 1$ mol/L)

Table 3.1: Process parameters and steady-state values for the reactor of Eq.3.28.

V	$=$	0.1	m^3
R	$=$	8.314	$kJ/kmol \cdot K$
C_{A0s}	$=$	1.0	$kmol/m^3$
T_{A0s}	$=$	310.0	K
ΔH	$=$	-4.78×10^4	$kJ/kmol$
k_0	$=$	1.2×10^9	s^{-1}
E	$=$	8.314×10^4	$kJ/kmol$
c_p	$=$	0.239	$kJ/kg \cdot K$
ρ	$=$	1000.0	kg/m^3
F	$=$	1.67×10^{-3}	m^3/s
T_{Rs}	$=$	395.33	K
C_{As}	$=$	0.57	$kmol/m^3$

using only measurements of the reactor temperature. For the purpose of implementing the control scheme, the process model of Eq.3.28 was linearized around the unstable steady-state, and the resulting linear model was used for the design of the hybrid predictive control structure. Using Eqs.3.12–3.13, we initially designed the bounded controller and constructed its stability region via Eqs.3.14–3.15. The matrix P was chosen as:

$$P = \begin{bmatrix} 0.027 & 0.47 \\ 0.47 & 15.13 \end{bmatrix}$$

and the observer gain parameter was chosen to be $\beta = 5$. It should be noted here that, because of the linearization effect, the largest invariant region, Ω , computed using Eqs.3.14–3.15 applied to the linear model, includes physically meaningless initial conditions ($C_A < 0$) and, therefore, a smaller level set, $\Omega' \subset \Omega$, that includes only physically meaningful initial conditions, was chosen and used as the stability region estimate for the bounded controller (see Figure 3.11). For the predictive controller, the parameters in the objective function of Eq.3.9 were chosen as $Q = qI$, with $q = 1$ and $R = rI$, with $r = 1$. We also chose a horizon length of $T = 2$ in implementing the predictive controller of Eqs.3.9–3.10. The resulting quadratic program was solved using the MATLAB subroutine QuadProg, and the set of ODEs integrated using the

MATLAB solver ODE45. In all simulations, the controllers, designed on the basis of the linearization around the unstable steady-state, were implemented on the nonlinear system of Eq.3.28.

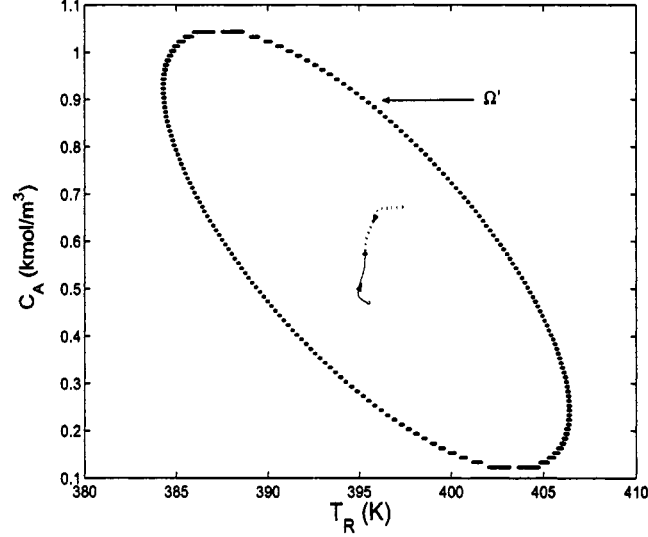


Figure 3.11: Stability region estimate, Ω' , based on the linearization of the nonlinear model of Eq.3.28. The solid line depicts the closed-loop state trajectory starting from an initial condition for which MPC (designed based on the linear model with $T = 2$) is feasible and stabilizes the nonlinear closed-loop system, while the dotted line depicts the state trajectory starting from an initial condition for which MPC is infeasible, and the hybrid predictive control strategy is implemented instead to enforce closed-loop stability.

As seen by the solid lines in Figures 3.12–3.13, starting from the point $(T_R, C_A) = (395.53 \text{ K}, 0.47 \text{ kmol/m}^3)$, the model predictive controller is able to stabilize the process at the desired steady-state, $(T_{RS}, C_{AS}) = (395.33 \text{ K}, 0.57 \text{ kmol/m}^3)$. From the initial condition, $(T_R, C_A) = (397.33 \text{ K}, 0.67 \text{ kmol/m}^3)$, however, the model predictive controller yields an infeasible solution. Therefore, using the proposed hybrid control strategy (dotted lines in Figures 3.12–3.13), the supervisor implements the bounded controller in the closed-loop system, while continuously checking the feasibility of MPC. At $t = 0.9$ seconds, MPC is found to be feasible and the supervisor switches to the

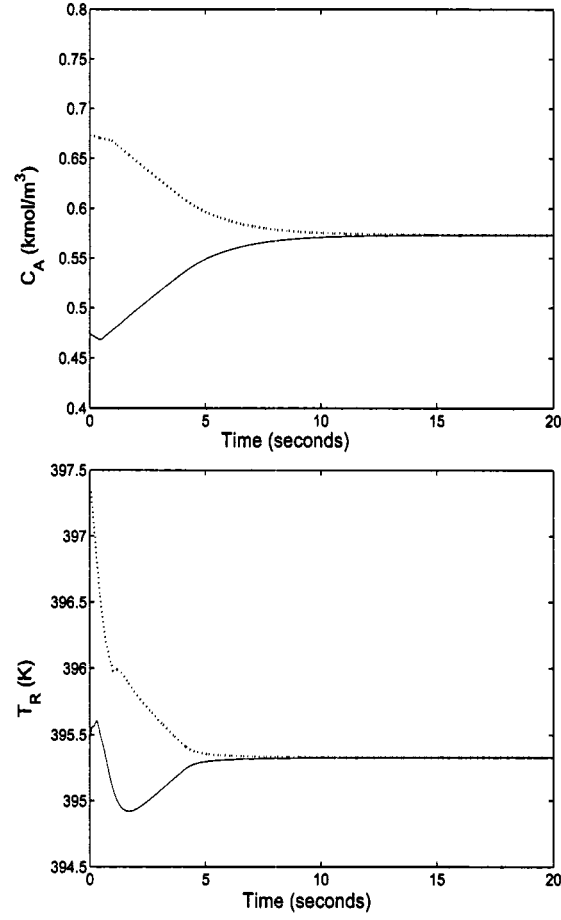


Figure 3.12: Closed-loop reactant concentration (top) and reactor temperature (bottom) starting from an initial condition for which MPC (designed based on the linear model with $T = 2$) is feasible and stabilizes the nonlinear closed-loop system (solid), and starting from an initial condition for which MPC is infeasible, and the hybrid predictive control strategy is implemented instead to enforce closed-loop stability (dotted).

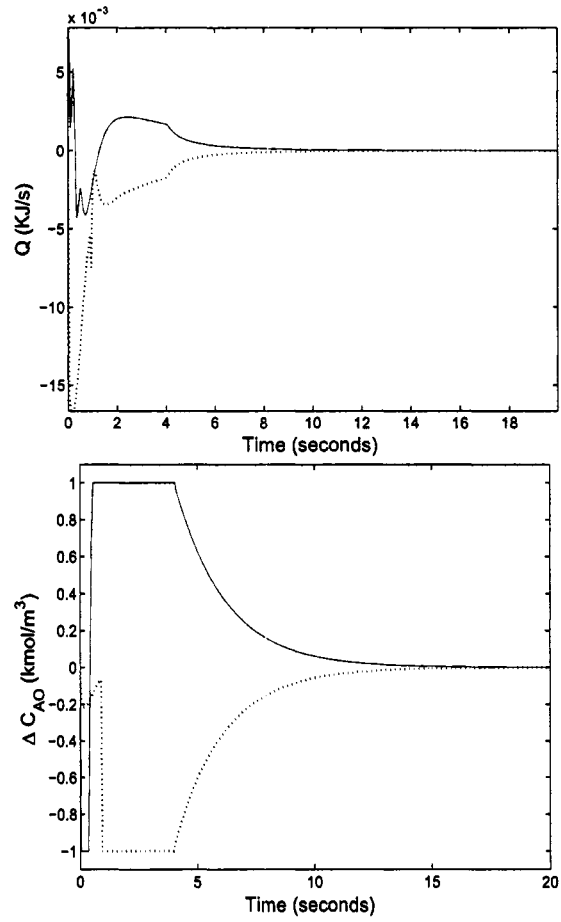


Figure 3.13: Rate of heat input (top) and inlet reactant concentration in deviation variable form (bottom) when MPC is implemented (solid) and when the hybrid predictive control strategy is implemented (dotted).

model predictive controller, which in turn stabilizes the nonlinear closed-loop system at the desired steady-state. These results clearly demonstrate a certain degree of robustness that the proposed hybrid predictive control structure possesses with respect to modeling errors (arising in this case due to neglecting the nonlinearities).

Remark 3.22: In both simulation studies presented above, we have treated robustness as a practical implementation issue and performed simulations that test the performance of the control strategy in the event of some plant-model mismatch to show that the hybrid predictive controller does possess some robustness margins that make it suited for practical implementation. We note that our objective here is not to incorporate robustness explicitly into the controller design, but rather to investigate the robustness margins of the controllers. The issue of incorporating robustness explicitly into the hybrid predictive control strategy is beyond the scope of the current work and is the subject of other research work (see [72]). Furthermore, within predictive control, a common approach to the robust controller design problem is through the min-max type formulations. However, these formulations are well-known to be computationally demanding and, more importantly, suffer from similar difficulties in identifying the set of initial conditions from which robust closed-loop stability can be guaranteed *a priori*.

3.7 Conclusions

In this chapter, a hybrid predictive control structure, which seamlessly unites MPC and bounded control, was proposed for the output feedback stabilization of linear systems with input constraints. The hybrid control structure consists of four components, including a bounded controller, a model predictive controller, a state estimator and a higher-level supervisor that orchestrates the switching between the two controllers. The output feedback bounded controller, provides a well-defined stability region (but no optimal performance) while the model predictive controller provides the desired optimal performance (but no explicit characterization of the stability region). Switch-

ing laws were derived to reconcile the two approaches in a way that guarantees: (a) closed-loop stability for all initial conditions within the stability region of the bounded controller, and (b) achievement of the MPC optimal performance whenever the pertinent stability criteria are met. The hybrid predictive control structure was shown to provide a safety net for the practical implementation of output feedback MPC by providing, through off-line computations, *a priori* knowledge of a large set of initial conditions for which stability of the switched closed-loop system is guaranteed. Finally, simulation studies were presented to demonstrate the implementation and evaluate the effectiveness of the proposed hybrid predictive control strategy, as well as test its robustness with respect to modeling errors and measurement noise.

Chapter 4

Hybrid Predictive Control of Nonlinear Systems: Method And Applications to Chemical Processes

The stabilization of dynamical systems using constrained control is an important problem that has been the subject of significant research work in control theory. Model predictive control (MPC) is a popular method for handling constraints within an optimal control setting. In MPC, the control action is obtained by solving repeatedly, on-line, a finite-horizon constrained open-loop optimal control problem. When the system is linear, the cost quadratic, and the constraints convex, the MPC optimization problem reduces to a quadratic program for which efficient software exists and, consequently, a number of control-relevant issues have been explored, including issues of closed-loop stability, performance, implementation and constraint satisfaction (e.g., see the tutorial paper [179]).

Many systems of practical interest, however, exhibit highly nonlinear behavior that must be accounted for when designing the controller. In the literature, several nonlinear model predictive control (NMPC) schemes have been developed (e.g., see [140, 149, 31, 115, 217, 132, 219]) that focus on the issues of stability, constraint satisfaction and

performance optimization for nonlinear systems. A common idea of these approaches is to enforce stability by means of suitable penalties and constraints on the state at the end of the finite optimization horizon. Because of the system nonlinearities, however, the resulting optimization problem is non-convex and, therefore, much harder to solve, even if the cost functional and constraints are convex. The computational burden is more pronounced for NMPC algorithms that employ terminal stability constraints (e.g., equality constraints) whose enforcement requires intensive computations that typically cannot be performed within a limited time-window.

In addition to the computational difficulties of solving a nonlinear optimization problem at each time step, one of the key challenges that impact on the practical implementation of NMPC is the inherent difficulty of characterizing, *a priori*, the set of initial conditions starting from where a given NMPC controller is guaranteed to stabilize the closed-loop system. For finite-horizon MPC, an adequate characterization of the stability region requires an explicit characterization of the complex interplay between several factors, such as the initial condition, the size of the constraints, the horizon length, the penalty weights, etc. Use of conservatively large horizon lengths to address stability only increases the size and complexity of the nonlinear optimization problem and could make it intractable. Furthermore, since feasibility of NMPC is determined through on-line optimization, unless a NMPC controller is exhaustively tested by simulation over the whole range of potential initial states, doubt will always remain as to whether or not a state will be encountered for which an acceptable solution to the finite horizon problem can be found.

The above host of theoretical and computational issues continue to motivate research efforts in this area. Most available predictive control formulations for nonlinear systems, however, either do not explicitly characterize the stability region, or provide estimates of this region based on linearized models, used as part of some scheduling scheme between a set of local predictive controllers. The idea of scheduling of a set

of local controllers to enlarge the operating region was proposed earlier in the context of analytic control of nonlinear systems (e.g., see [129, 2]) and requires an estimate of the region of stability for the local controller designed at each scheduling point. Then by designing a set of local controllers with their estimated regions of stability overlapping each other, supervisory scheduling of the local controllers can move the state through the intersections of the estimated regions of stability of different controllers to the desired operating point. Similar ideas were used in [33, 227] for scheduled predictive control of nonlinear systems. All of these approaches require the existence of an equilibrium surface that connects the scheduling points, and the resulting stability region estimate is the union of the local stability regions, which typically forms an envelope around the equilibrium surface. Stability region estimates based on linearization, however, are inherently conservative.

In contrast to this approach, progress within the area of analytic nonlinear controller design has made available bounded controllers with well-characterized stability properties (e.g., [127, 134, 126]). The class of Lyapunov-based bounded controllers in [127] was extended in [66, 68] to handle input constraints and model uncertainty within an inverse optimal control setting, and provide at the same time an explicit characterization of the constrained stability region. Since the analytic controllers are designed on the basis of the nonlinear model, and also account explicitly for the constraints in the design of the controller, they yield a substantially larger region of stability than controllers based on linearized models. However, the resulting closed-loop performance is not necessarily optimal with respect to arbitrary cost functionals.

In a previous work [75], we developed a hybrid control scheme, uniting bounded control with MPC, for the stabilization of linear systems with input constraints. The scheme was predicated upon the idea of switching between a bounded controller, for which the region of constrained closed-loop stability is explicitly characterized, and a model predictive controller that minimizes a given performance objective subject to

constraints. Switching laws were derived to orchestrate the transition between the two controllers in a way that reconciles the tradeoffs between their respective stability and optimality properties, and guarantees asymptotic closed-loop stability for all initial conditions within the stability region of the bounded controller. The hybrid scheme was shown to provide a safety net for the implementation of MPC under state feedback.

In this chapter, we propose a hybrid control structure for the stabilization of nonlinear systems with input constraints. The central idea is to use a family of bounded nonlinear controllers, each with an explicitly characterized stability region, as fall-back controllers, and embed the operation of MPC within the union of these regions. In the event that the given predictive controller (which can be linear, nonlinear, or even scheduled) is unable to stabilize the closed-loop system (e.g., due to failure of the optimization algorithm, poor choice of the initial condition, insufficient horizon length, etc.), supervisory switching from MPC to any of the bounded controllers, whose stability region contains the trajectory, guarantees closed-loop stability. The rest of the paper is organized as follows. In section 2, some mathematical preliminaries are presented to describe the class of systems under consideration and briefly review how the constrained control problem is addressed within the bounded control and MPC frameworks. In section 3, the controller switching problem is formulated first and then its solution is presented in the form of a hybrid control structure that employs switching between MPC and bounded control. Two representative switching schemes, that address (with varying degrees of flexibility) stability and performance objectives, are described to highlight the theoretical underpinnings and practical implications of the proposed hybrid control structure. Possible extensions of the supervisory switching logic, to address a variety of practical implementation issues, are also discussed. Finally in section 4, the efficacy of the proposed approach is demonstrated through applications to chemical reactor and crystallization process examples.

4.1 Preliminaries

In this chapter, we consider the problem of asymptotic stabilization of continuous-time nonlinear systems with input constraints, with the following state-space description

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \quad (4.1)$$

$$\|u\| \leq u_{max} \quad (4.2)$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables, $u = [u_1 \cdots u_m]'$ is the vector of manipulated inputs, $u_{max} \geq 0$ denotes the bound on the manipulated inputs, $f(\cdot)$ is a sufficiently smooth $n \times 1$ nonlinear vector function, and $g(\cdot)$ is a sufficiently smooth $n \times m$ nonlinear matrix functions. Without loss of generality, it is assumed that the origin is the equilibrium point of the unforced system (i.e. $f(0) = 0$). Throughout the paper, the notation $\|\cdot\|$ will be used to denote the standard Euclidean norm of a vector, while the notation $\|\cdot\|_Q$ refers to the weighted norm, defined by $\|x\|_Q^2 = x'Qx$ for all $x \in \mathbb{R}^n$, where Q is a positive-definite symmetric matrix and x' denotes the transpose of x . In order to provide the necessary background for our main results in section 3, we will briefly review in the remainder of this section the design procedure for, and the stability properties of, both the bounded and model predictive controllers, which constitute the basic components of our hybrid control scheme. For clarity of presentation, we will focus only on the state feedback problem where measurements of $x(t)$ are assumed to be available for all t (see Remark 4.16 below for a discussion on the issue of measurement sampling and how it can be handled).

4.1.1 Bounded Lyapunov-based control

Consider the system of Eqs.4.1-4.2, for which a family of control Lyapunov functions (CLFs), $V_k(x)$, $k \in \mathcal{K} \equiv \{1, \dots, p\}$ has been found (see Remark 4.1 below for a discussion on the construction of CLFs). Using each control Lyapunov function, we

construct, using the results in [127] (see also [66, 68]), the following continuous bounded control law

$$u_k(x) = -k_k(x)(L_g V_k)'(x) \equiv b_k(x) \quad (4.3)$$

where

$$k_k(x) = \frac{L_f V_k(x) + \sqrt{(L_f V_k(x))^2 + (u_{max} \|(L_g V_k)'(x)\|)^4}}{\|(L_g V_k)'(x)\|^2 \left[1 + \sqrt{1 + (u_{max} \|(L_g V_k)'(x)\|)^2} \right]} \quad (4.4)$$

$L_f V_k(x) = \frac{\partial V_k(x)}{\partial x} f(x)$, $L_g V_k(x) = [L_{g_1} V_k(x) \cdots L_{g_m} V_k(x)]'$ and $g_i(x)$ is the i -th column of the matrix $g(x)$. For the above controller, it can be shown, using standard Lyapunov arguments, that whenever the closed-loop state trajectory, x , evolves within the state-space region described by the set

$$\Phi_k(u_{max}) = \{x \in \mathbb{R}^n : L_f V_k(x) < u_{max} \|(L_g V_k)'(x)\|\} \quad (4.5)$$

then the controller satisfies the constraints, and the time-derivative of the Lyapunov function is negative-definite. Therefore, starting from any initial state within the set $\Phi_k(u_{max})$, asymptotic stability of the origin of the constrained closed-loop system can be guaranteed, provided that the state trajectory remains within region described by $\Phi_k(u_{max})$ whenever $x \neq 0$. To ensure this, we consider initial conditions that belong to an invariant subset (preferably the largest), $\Omega_k(u_{max})$ (this idea was also used in [66, 68] in the context of bounded robust inverse optimal control of constrained nonlinear systems). One way to construct such a subset is using the level sets of V_k , i.e.

$$\Omega_k(u_{max}) = \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\} \quad (4.6)$$

where $c_k^{max} > 0$ is the largest number for which $\Phi_k(u_{max}) \supset \Omega_k(u_{max}) \setminus \{0\}$. The union of the invariant regions described by the set

$$\Omega(u_{max}) = \bigcup_{k=1}^p \Omega_k(u_{max}) \quad (4.7)$$

then provides an estimate of the stability region, starting from where the origin of the constrained closed-loop system, under the appropriate control law from the family of

Eqs.4.3-4.4, is guaranteed to be asymptotically stable.

Remark 4.1: CLF-based stabilization of nonlinear systems has been studied extensively in the nonlinear control literature (e.g., see [9, 127, 80, 207]). The construction of constrained CLFs (i.e. CLFs that take the constraints into account) remains a difficult problem (especially for nonlinear systems) that is the subject of ongoing research. For several classes of nonlinear systems that arise commonly in the modeling of practical systems, systematic and computationally feasible methods are available for constructing unconstrained CLFs (CLFs for the unconstrained system) by exploiting the system structure. Examples include the use of quadratic functions for feedback linearizable systems and the use of back-stepping techniques to construct CLFs for systems in strict feedback form. In this chapter, the bounded controllers in Eqs.4.3-4.4 are designed using unconstrained CLFs, which are also used to explicitly characterize the associated regions of stability via Eqs.4.5-4.6. While the resulting estimates do not necessarily capture the entire domain of attraction, we will use them throughout the paper only for a concrete illustration of the basic ideas of the results. It is possible to obtain substantially improved (i.e. less conservative) estimates by using, for example, a combination of several CLFs (see section 4 for examples).

4.1.2 Model predictive control

In this section, we consider model predictive control of the system described by Eq.4.1, subject to the control constraints of Eq.4.2. In the literature, several MPC formulations are currently available, each with its own merits and limitations. While the results of this chapter are not restricted to any particular MPC formulation (see Remark 4.6 below), we will briefly describe here the “traditional” formulation for the purpose of highlighting some of the theoretical and computational issues involved in the nonlinear setting. For this case, MPC at state x and time t is conventionally obtained by solving,

on-line, a finite horizon optimal control problem [142] of the form

$$P(x, t) : \min\{J(x, t, u(\cdot)) | u(\cdot) \in S\} \quad (4.8)$$

$$s.t. \dot{x} = f(x) + g(x)u \quad (4.9)$$

where $S = S(t, T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ , mapping $[t, t+T]$ into $\mathcal{U} := \{u \in \mathbb{R}^m : \|u\| \leq u_{max}\}$ and T is the specified horizon. Eq.4.9 is a nonlinear model describing the time evolution of the states x . A control $u(\cdot)$ in S is characterized by the sequence $\{u[k]\}$ where $u[k] := u(k\Delta)$. A control $u(\cdot)$ in S satisfies $u(t) = u[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. The performance index is given by

$$J(x, t, u(\cdot)) = \int_t^{t+T} [\|x^u(s; x, t)\|_Q^2 + \|u(s)\|_R^2] ds + F(x(t+T)) \quad (4.10)$$

where R and Q are strictly positive-definite, symmetric matrices and $x^u(s; x, t)$ denotes the solution of Eq.4.1, due to control u , with initial state x at time t and $F(\cdot)$ denotes the terminal penalty. In addition to penalties on the state and control action, the objective function may also include penalties on the rate of input change reflecting limitations on actuator speed (e.g., a large valve requiring few seconds to change position). The minimizing control $u^0(\cdot) \in S$ is then applied to the plant over the interval $[k\Delta, (k+1)\Delta)$ and the procedure is repeated indefinitely. This defines an implicit model predictive control law

$$M(x) = \operatorname{argmin}(J(x, t, u(\cdot))) = u^0(t; x, t) \quad (4.11)$$

While the use of a nonlinear model as part of the optimization problem is desirable to account for the system's nonlinear behavior, it also raises a number of well-known theoretical and computational issues [142] that impact on the practical implementation of MPC. For example, in the nonlinear setting, the optimization problem is non-convex and, therefore, harder to solve than in the linear case. Furthermore, the issue of closed-loop stability is typically addressed by introducing penalties and constraints on the

state at the end of the finite optimization horizon (see [141] for a survey of different constraints proposed in the literature). Imposing constraints adds to the computational complexity of the nonlinear optimization problem which must be solved at each time instance.

Even if the optimization problem could be solved in a reasonable time, any guarantee of closed-loop stability remains critically dependent upon making the appropriate choice of the initial condition, which must belong to the predictive controller's region of stability (or feasibility), which, in turn, is a complex function of the constraints, the performance objective, and the horizon length. However, the implicit nature of the nonlinear MPC law, obtained through repeated on-line optimization, limits our ability to obtain, *a priori* (i.e. before controller implementation), an explicit characterization of the admissible initial conditions starting from where a given MPC controller (with a fixed performance index and horizon length) is guaranteed to asymptotically stabilize the nonlinear closed-loop system. Therefore, the initial conditions and the horizon lengths are usually tested through closed-loop simulations, which can add to the computational burden prior to the implementation of MPC.

4.2 The hybrid control strategy: uniting bounded control and MPC

By comparing the bounded controller and MPC designs presented in the previous section, some tradeoffs with respect to their stability and optimality properties are evident. The bounded controller, for example, possesses a well-defined region of admissible initial conditions that guarantee constrained closed-loop stability. However, its performance may not be optimal with respect to an arbitrary performance criterion. MPC, on the other hand, provides the desired optimality requirement, but poses implementation difficulties and lacks an explicit characterization of the stability region. In this section, we show how to reconcile the two approaches by means of a switching

scheme that provides a safety net for the implementation of MPC to nonlinear systems.

4.2.1 Formulation of the switching problem

Consider the constrained nonlinear system of Eqs.4.1-4.2, for which the bounded controllers of Eqs.4.3-4.4 and predictive controller of Eqs.4.8-4.11 have been designed. The control problem is formulated as the one of designing a set of switching laws that orchestrate the transition between MPC and the bounded controllers in a way that guarantees asymptotic stability of the origin of the closed-loop system starting from any initial condition in the set $\Omega(u_{max})$ defined in Eq.4.7, respects input constraints, and accommodates the optimality requirements whenever possible. For a precise statement of the problem, the system of Eq.4.1 is first cast as a switched system of the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u_{i(t)} \\ \|u_i\| &\leq u_{max} \\ i(t) &\in \{1, 2\} \end{aligned} \tag{4.12}$$

where $i : [0, \infty) \rightarrow \{1, 2\}$ is the switching signal, which is assumed to be a piecewise continuous (from the right) function of time, implying that only a finite number of switches, between the predictive and bounded controllers, is allowed on any finite interval of time. The index, $i(t)$, which takes values in the set $\{1, 2\}$, represents a discrete state that indexes the control input $u(\cdot)$, with the understanding that $i(t) = 1$ if and only if $u_i(x(t)) = M(x(t))$ and $i(t) = 2$ if and only if $u_i(x(t)) = b_k(x(t))$ for some $k \in \mathcal{K}$. Our goal is to construct a switching law

$$i(t) = \psi(x(t), t) \tag{4.13}$$

that provides the set of switching times that ensure stabilizing transitions between the predictive and bounded controllers, in the event that the predictive controller is unable to enforce closed-loop stability. This in turn determines the time-course of the discrete state $i(t)$.

In the remainder of this section, two switching schemes that address the above

problem are presented. The first scheme is given in Theorem 4.1 (section 4.3.2) and focuses primarily on the issue of closed-loop stability, while the second scheme, given in Theorem 2 (section 4.3.3), provides more flexible switching rules that guarantee closed-loop stability and, simultaneously, enhance the overall closed-loop performance beyond that obtained from the first scheme. The proofs of both theorems are also given.

4.2.2 Stability-based controller switching

Theorem 4.1: *Consider the constrained nonlinear system of Eq.4.12, with any initial condition $x(0) \equiv x_0 \in \Omega_k(u_{max})$, for some $k \in \mathcal{K} \equiv \{1, \dots, p\}$, where Ω_k was defined in Eq.4.6, under the model predictive controller of Eqs.4.8-4.11. Also let $\bar{T} \geq 0$ be the earliest time for which either the closed-loop state, under MPC, satisfies*

$$L_f V_k(x(\bar{T})) + L_g V_k(x(\bar{T}))M(x(\bar{T})) \geq 0 \quad (4.14)$$

or the MPC algorithm fails to prescribe any control move. Then, the switching rule given by

$$i(t) = \begin{cases} 1, & 0 \leq t < \bar{T} \\ 2, & t \geq \bar{T} \end{cases} \quad (4.15)$$

where $i(t) = 1 \Leftrightarrow u_i(x(t)) = M(x(t))$ and $i(t) = 2 \Leftrightarrow u_i(x(t)) = b_k(x(t))$, guarantees that the origin of the switched closed-loop system is asymptotically stable.

Proof of Theorem 4.1:

Step 1: Substituting the control law of Eqs.4.3-4.4 into the system of Eq.4.1 and evaluating the time-derivative of the Lyapunov function along the trajectories of the closed-loop system, it can be shown that $\dot{V}_k < 0$ for all $x \in \Phi_k(u_{max})$ (where $\Phi_k(u_{max})$ was defined in Eq.4.5). Since $\Omega_k(u_{max})$ (defined in Eq.4.6) is an invariant subset of $\Phi_k(u_{max}) \cup \{0\}$, it follows that for any $x(0) \in \Omega_k(u_{max})$, the origin of the closed-loop system, under the control law of Eqs.4.3-4.4, is asymptotically stable.

Step 2: Consider the switched closed-loop system of Eq.4.12, subject to the switching

rule of Eqs.4.14-4.15, with any initial state $x(0) \in \Omega_k(u_{max})$. From the definition of $\bar{T}$ given in Theorem 4.1, it is clear that if $\bar{T}$ is a finite number, then $\dot{V}_k(x^M(t)) < 0 \ \forall 0 \leq t < \bar{T}$, where the notation $x^M(t)$ denotes the closed-loop state under MPC at time t , which implies that $x(t) \in \Omega_k(u_{max}) \ \forall 0 \leq t < \bar{T}$ (or that $x(\bar{T}^-) \in \Omega_k(u_{max})$). This fact, together with the continuity of the solution of the switched system, $x(t)$, (following from the fact that the right hand side of Eq.4.12 is continuous in x and piecewise continuous in time) implies that, upon switching (instantaneously) to the bounded controller at $t = \bar{T}$, we have $x(\bar{T}) \in \Omega_k(u_{max})$ and $u(t) = b_k(x(t))$ for all $t \geq \bar{T}$. Therefore, from our analysis in step 1, we conclude that $\dot{V}_k(x^{b_k}(t)) < 0 \ \forall t \geq \bar{T}$. In summary, the switching rule of Eqs.4.14-4.15 guarantees that, starting from any $x(0) \in \Omega_k(u_{max})$, $\dot{V}_k(x(t)) < 0 \ \forall x \neq 0, x \in \Omega_k(u_{max}), \forall t \geq 0$, which implies that the origin of the switched closed-loop system is asymptotically stable. Note that if no such $\bar{T}$ exists, then we simply have from Eqs.4.14-4.15 that $\dot{V}_k(x^M(t)) < 0 \ \forall t \geq 0$, and the origin of the closed-loop system is also asymptotically stable. This completes the proof of the theorem.

Remark 4.2: Theorem 4.1 describes a stability-based switching strategy for control of nonlinear systems with input constraints. The main components of this strategy include4. the predictive controller, a family of bounded nonlinear controllers, with their estimated regions of constrained stability, and a high-level supervisor that orchestrates the switching between the controllers. A schematic representation of the hybrid control structure is shown in Figure 4.1. The implementation procedure of this hybrid control strategy is outlined below:

- Given the system model of Eq.4.1, the constraints on the input and the family of CLFs, design the bounded controllers using Eqs.4.3-4.4. Given the performance objective, set up the MPC optimization problem.
- Compute the stability region estimate for each of the bounded controllers, $\Omega_k(u_{max})$,

using Eqs.4.5-4.6, for $k = 1, \dots, p$, and $\Omega(u_{max}) = \bigcup_{k=1}^p \Omega_k(u_{max})$.

- Initialize the closed-loop system under MPC, at any initial condition, x_0 within Ω , and identify a CLF, $V_k(x)$, for which the initial condition is within the corresponding stability region estimate, Ω_k .
- Monitor the temporal evolution of the closed-loop trajectory (by checking Eq.4.14 at each time) until the earliest time that either Eq.4.14 holds or the MPC algorithm prescribes no solution, $\bar{T}$.
- If such a $\bar{T}$ exists, discontinue MPC implementation, switch to the k -th bounded controller (whose stability region contains x_0) and implement it for all future times.

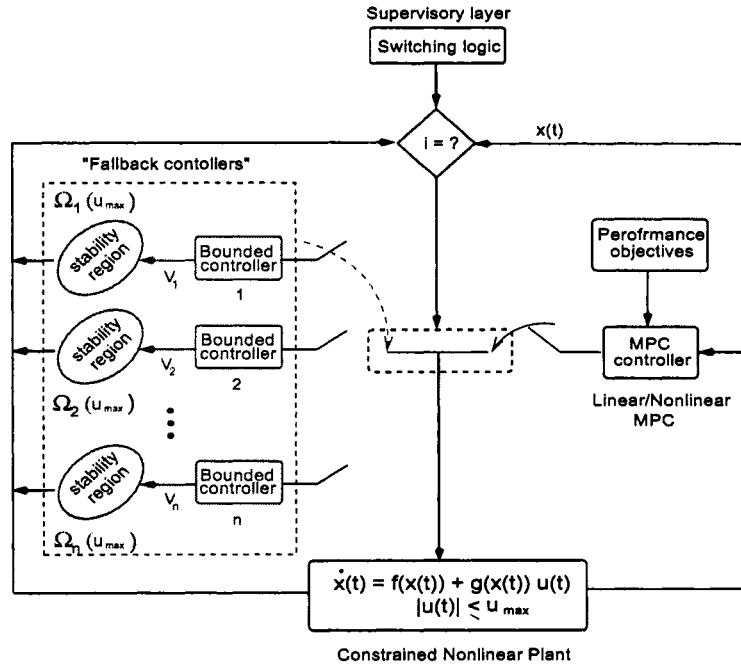


Figure 4.1: Schematic representation of the hybrid control structure merging MPC and a family of fall-back bounded controllers with their stability regions.

Remark 4.3: The use of multiple CLFs to design a family of bounded controllers, with their estimated regions of stability, allows us to initialize the closed-loop system

from a larger set of initial conditions than in the case when a single CLF is used. Note, however, that once the initial condition is fixed, this determines both the region where MPC operation will be confined (and monitored) and the corresponding fall-back bounded controller to be used in the event of MPC failure. If the initial condition falls within the intersection of several stability regions, then any of the corresponding bounded controllers can be used as the fall-back controller.

Remark 4.4: The relation of Eqs.4.14-4.15 represents the switching rule that the supervisor observes when contemplating whether a switch, between MPC and any of the bounded controllers at a given time, is needed. The left hand-side of Eq.4.14 is the rate at which the Lyapunov function grows or decays along the trajectories of the closed-loop system, under MPC, at time $\bar{T}$. By observing this rule, the supervisor tracks the temporal evolution of V_k , under MPC, such that whenever an increase in V_k is detected after the initial implementation of MPC (or if the MPC algorithm fails to prescribe any control move, e.g., due to optimization failure), the predictive controller is disengaged from the closed-loop system, and the appropriate bounded controller is switched in, thus steering the closed-loop trajectory to the origin asymptotically. This switching rule, together with the choice of the initial condition, guarantee that the closed-loop trajectory, under MPC, never escapes $\Omega_k(u_{max})$ before the corresponding bounded controller can be activated. The idea of designing the switching logic based on monitoring the state's temporal evolution with respect to the stability region was introduced in [67] in the context of control of switched (multi-modal) nonlinear systems with input constraints.

Remark 4.5: In the case when the condition of Eq.4.14 is never fulfilled (i.e. MPC continues to be feasible for all times and the Lyapunov function continues to decay monotonically for all times ($\bar{T} = \infty$)), the switching rule of Eq.4.15 ensures that only MPC is implemented for all times and that no switching to the fall-back controllers takes place. In this case, MPC is stabilizing and its optimal performance is fully recov-

ered by the hybrid control structure. This particular feature underscores the central objective of the hybrid control structure, which is not to replace or subsume MPC but, instead, to provide a safe environment for the implementation of any optimal predictive control policy for which *a priori* guarantees of stability are not available. Note also that, to the extent that stability under MPC is captured by the given Lyapunov function, the notion of switching, as described in Theorem 4.1, does not result in loss of performance, since the transition to the bounded controller takes place only if MPC is infeasible or destabilizing. Clearly, under these circumstances the issue of optimality is not very meaningful for the predictive controller.

Remark 4.6: The fact that closed-loop stability is guaranteed, for all $x_0 \in \Omega$ (through supervisory switching), independently of whether MPC itself is stabilizing or not, allows us to use any desired MPC formulation within the switching scheme (and not just the one mentioned in Theorem 4.1), whether linear or nonlinear, and with or without terminal stability constraints or terminal penalties, without concern for loss of stability. This flexibility of using any desired MPC formulation has important practical implications for reducing the computational complexities that arise in implementing predictive control algorithms to nonlinear systems by allowing, for example, the use of less-computationally demanding NMPC formulations, or even the use of linear MPC algorithms instead (based on linearized models of the plant), with guaranteed stability (see subsections 4.1 and 4.2 for examples). In all cases, by embedding the operation of the chosen MPC algorithm within the large and well-defined stability regions of the bounded controllers, the switching scheme provides a safe fall-back mechanism that can be called upon at any moment to preserve closed-loop stability should MPC become unable to achieve closed-loop stability. Finally, we note that the Lyapunov functions used by the bounded controllers can also be used as a guide to design and tune the MPC in case a Lyapunov-based constraint is used at the end of the prediction horizon.

Remark 4.7: Note that no assumption is made regarding how fast the MPC optimization needs to be solved since, even if this time is relatively significant and the plant dynamics are unstable, the implementation of the switching rule in Theorem 4.1 guarantees an “instantaneous” switch to the appropriate stabilizing bounded controller before such delays can adversely affect closed-loop stability (the times needed to check Eqs.4.14-4.15 and compute the control action of the bounded controller are insignificant as they involve only algebraic computations). This feature is valuable when computationally-intensive NMPC formulations fail, in the course of the online optimization, to provide an acceptable solution in a reasonable time. In this case, switching to the bounded controller allows us to safely abort the optimization without loss of stability.

Remark 4.8: One of the important issues in the practical implementation of finite-horizon MPC is the selection of the horizon length. It is well known that this selection can have a profound effect on nominal closed-loop stability as well as the size and complexity of the optimization problem. For NMPC, however, *a priori* knowledge of the shortest horizon length that guarantees closed-loop stability, from a given set of initial conditions (alternatively, the set of feasible initial conditions for a given horizon length) is not available. Therefore, in practice the horizon length is typically chosen using *ad hoc* selection criteria, tested through extensive closed-loop simulations, and varied, if necessary, to achieve stability. In the switching scheme of Theorem 4.1, closed-loop stability is maintained independently of the horizon length. This allows the predictive control designer to choose the horizon length solely on the basis of what is computationally practical for the size of the optimization problem, and without increasing, unnecessarily, the horizon length (and consequently the computational load) out of concern for stability. Furthermore, even if a conservative estimate of the necessary horizon length were to be ascertained for a small set of initial conditions before MPC implementation (say through extensive closed-loop simulations and testing), then if some disturbance were to drive the state outside of this set during the on-line imple-

mentation of MPC, this estimate may not be sufficient to stabilize the closed-loop system from the new state. Clearly, in this case, running extensive closed-loop simulations on-line to try to find the new horizon length needed is not a feasible option, considering the computational difficulties of NMPC as well as the significant delays that would be introduced until (and if) the new horizon could be determined. In contrast to the on-line re-tuning of MPC, stability can be preserved by switching to the fall-back controllers (provided that the disturbed state still lies within Ω).

Remark 4.9 It is important to note that one can compute estimates of stability regions for closed-loop systems resulting from the application of various classes of linear/nonlinear controllers to linear/nonlinear systems without constraints (see [80, 66, 65, 32]). Such stability regions can be used for the constrained closed-loop system where the constraints on the control action and states are not violated, however, they may be unnecessarily conservative because they are computed without taking the constraints directly into account. Within these sets, the constrained control problem becomes equivalent to the unconstrained control problem, allowing for dropping the constraints and consequently the computational complexity of the control action calculation when optimization-based control algorithms are used. For nonlinear model predictive control, however, *a priori* knowledge of the shortest horizon length that guarantees closed-loop stability, from a given set of initial conditions (alternatively, the set of feasible initial conditions for a given horizon length) is not available.

Remark 4.10: When compared with other MPC-based approaches, the proposed hybrid control scheme is conceptually aligned with Lyapunov-based approaches in the sense that it, too, employs a Lyapunov stability condition to guarantee asymptotic closed-loop stability. However, this condition is enforced at the supervisory level, via continuous monitoring of the temporal evolution of V_k and explicitly switching between two controllers, rather than being incorporated in the optimization problem, as is customarily done in Lyapunov-based MPC approaches, whether as a terminal inequality

constraint (e.g., contractive MPC [167, 115], CLF-based RHC for unconstrained nonlinear systems [170]) or through a CLF-based terminal penalty (e.g., [217, 219]). The methods proposed in these works do not provide an explicit characterization of the set of states starting from where feasibility and/or stability is guaranteed *a priori*. Furthermore, the idea of switching to a fall-back controller with a well-characterized stability region, in the event that the MPC controller does not yield a feasible solution, is not considered in these approaches.

Remark 4.11: The idea of switching between different controllers for the purpose of achieving some objective that either cannot be achieved or is more difficult to achieve using a single controller has been widely used in the literature, and in a variety of contexts (see, for example, [197, 92, 2, 214]). In most of these works however, switching takes place between multiple structurally-similar controllers. In this chapter, switching is employed between two structurally-different, though complementary, control approaches as a tool for reconciling the objectives of optimal stabilization of the constrained closed-loop system (through MPC) and the *a priori* (off-line) determination of set of initial conditions for which closed-loop stability is guaranteed (through bounded control). The proposed switching schemes also differ, both in their objective and implementation, from other MPC formulations that involve switching. For example, in dual mode MPC [149] the strategy includes switching from MPC to a locally stabilizing controller once the state is brought near the origin by MPC. The purpose of switching in this approach is to relax the terminal equality constraint whose implementation is computationally burdensome for nonlinear systems. However, the set of initial conditions for which MPC is guaranteed to steer the state close to the origin is not explicitly known *a priori*. In contrast, switching from MPC to the bounded controller is used in our work only to prevent any potential closed-loop instability arising from implementing MPC without the *a priori* knowledge of the admissible initial conditions. Therefore, depending on the stability properties of the chosen MPC, switching may or may not occur, and if it occurs, it can take place near or far from the origin.

4.2.3 Enhancing closed-loop performance

The switching rule in Theorem 4.1 requires monitoring only one of the CLFs (any one for which $x_0 \in \Omega_k$) and does not permit any transient increase in this CLF under MPC (by switching immediately to the appropriate bounded controller). While this condition is sufficient to guarantee closed-loop stability, it does not take full advantage of the behavior of other CLFs at the switching time. For example, even though a given CLF, say V_1 , may start increasing at time $\bar{T}$ under MPC, another CLF, say V_2 , (for which $x(\bar{T})$ lies inside the corresponding stability region) may still be decreasing (see Figure 4.2). In this case, it would be desirable to keep MPC in the closed-loop (rather than switch to the bounded controller) and start monitoring the growth of V_2 instead of V_1 , because if V_2 continues to decay, then MPC can be kept active for all times and its optimal performance fully recovered. To allow for such flexibility, we extend in this section the switching strategy of Theorem 4.1 by relaxing the switching rule. This is formalized in the following theorem. The proof is given below.

Theorem 4.2: *Consider the constrained nonlinear system of Eq.4.12, with any initial condition $x(0) \equiv x_0 \in \Omega(u_{max})$, where $\Omega(u_{max})$ was defined in Eq.4.7, under the model predictive controller of Eqs.4.8-4.11. Let $T_k \geq 0$ be the earliest time for which*

$$\begin{aligned} V_k(x(T_k)) &\leq c_k^{max} \\ L_f V_k(x(T_k)) + L_g V_k(x(T_k)) M(x(T_k)) &\geq 0, \quad k \in \mathcal{K} \end{aligned} \quad (4.16)$$

and define

$$T_k^*(t) = \begin{cases} T_k, & 0 \leq t \leq T_k \\ 0, & t > T_k \end{cases} \quad (4.17)$$

Then, the switching rule given by

$$i(t) = \begin{cases} 1, & 0 \leq t < T^* \\ 2, & t \geq T^* \end{cases} \quad (4.18)$$

where $T^* = \min\{T^i, T^f\}$, $T^i \geq 0$ is the earliest time for which the MPC algorithm fails to prescribe any control move, and $T^f = \max_j \{T_j^*\}$ for all j such that $x(t) \in \Omega_j(u_{max})$, guarantees that the origin of the closed-loop system is asymptotically stable.

Proof of Theorem 4.2: The proof of this theorem, for the case when only one bounded controller is used as the fall-back controller (i.e. $p = 1$), is same as that of Theorem 4.1. To simplify the proof, we prove the result only for the case when the family of fall-back controllers consists of 2 controllers (i.e. $p = 2$). Generalization of the proof to the case of a family of p controllers, where $2 < p < \infty$, is conceptually straightforward. Also, without loss of generality, we consider the case when the optimization problem in MPC is feasible for all times (i.e. $T^i = \infty$), since if it is not (i.e. $T^i < \infty$), then the switching time is simply taken to be the minimum of $\{T^i, T^f\}$, as stated in Theorem 4.2.

Without loss of generality, let the closed-loop system be initialized within the stability region of the first bounded controller, $\Omega_1(u_{max})$, under MPC. Then one of the following scenarios will take place:

Case 1: $\dot{V}_1(x^M(t)) < 0$ for all $t \geq 0$. The switching law of Eqs.4.16-4.18 dictates in this case that MPC be implemented for all times. Since $x(0) \in \Omega_1(u_{max})$, where Ω_1 is a level set of V_1 , and $\dot{V}_1 < 0$, then the state of the closed-loop system is bounded and converges to the origin as $t \rightarrow \infty$.

Case 2: $\dot{V}_1(x^M(T_1)) \geq 0$ and $x^M(T_1) \in \Omega_1(u_{max})$, for some finite $T_1 > 0$. In this case, one of the following scenarios will occur:

(a) If $x(T_1)$ is outside of $\Omega_2(u_{max})$, then it follows from Eqs.4.16-4.18 that the supervisor will set $T^* = T_f = T_1$ and switch to the first bounded controller at $t = T_1$, using V_1 as the CLF, which enforces asymptotic stability as discussed in step 2 of the proof of Theorem 4.1.

(b) If $x(T_1) \in \Omega_2(u_{max})$ and $\dot{V}_2(x(T_1)) < 0$ (i.e. $0 < T_1 < T_2$), then the supervisor will keep MPC in the closed-loop system at T_1 . If T_2 is a finite number, then the supervisor will set $T^* = T_f = T_2$ (since $T_1^* = 0$ for all $t > T_1$ from Eq.4.17) at which time it will switch to the second bounded controller, using V_2 as the CLF. Since $\dot{V}_2(x^M(t)) < 0$ for all $T_1 \leq t < T_2$, and $x(T_1) \in \Omega_2(u_{max})$, then $x(T^{*-}) \in$

$\Omega_2(u_{max})$. By continuity of the solution of the closed-loop system, it follows that $x(T^*) \in \Omega_2(u_{max})$; and since $\Omega_2(u_{max})$ is the stability region corresponding to V_2 , then implies that upon implementation of the corresponding bounded controller for all future times, asymptotic stability is achieved. Note that if T_2 does not exist (or $T_2 = \infty$), then we simply have $x(T_1) \in \Omega_2(u_{max})$ and $\dot{V}_2(x(T_1)) < 0$ for all $t \geq T_1$, which implies that the origin of the closed-loop system is again asymptotically stable.

(c) If $0 < T_2 < T_1$ (i.e. $x(T_2) \in \Omega_2(u_{max})$, $V_1(x(T_2)) < 0$ and $V_2(x(T_2)) \geq 0$), then it follows from Eqs.4.16-4.18 that the supervisor will set $T^* = T_f = T_1$ (since $T_2^* = 0$ for all $t > T_2$ from Eq.4.17) and switch to the first bounded controller, using V_1 as the CLF. Since $\dot{V}_1(x^M(t)) < 0$ for all $t < T_1$, and $x(0) \in \Omega_1(u_{max})$, then $x(T^*) \in \Omega_1(u_{max})$. By continuity of the solution of the closed-loop system, we have $x(T^*) \in \Omega_1(u_{max})$, and since $\Omega_1(u_{max})$ is the stability region corresponding to V_1 , this implies that upon implementation of the first bounded controller for the remaining time, asymptotic closed-loop stability is achieved. This completes the proof of the theorem.

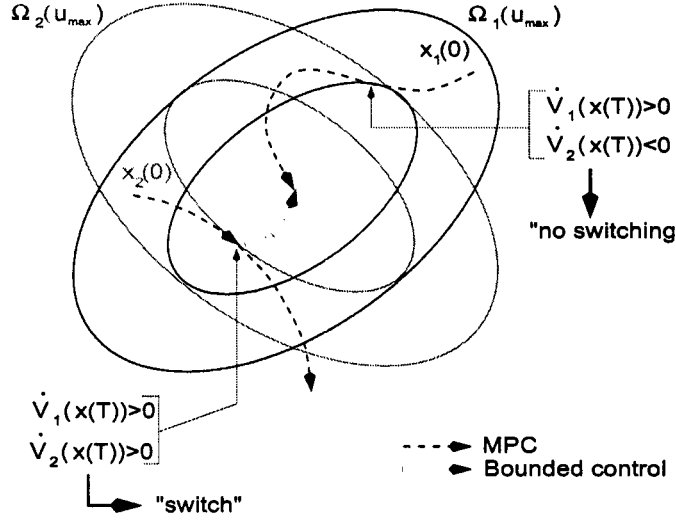


Figure 4.2: Schematic representation illustrating the main idea of the controller switching scheme proposed in Theorems 1 and 2.

Remark 4.12: The implementation of the switching scheme proposed in Theorem 4.2

can be understood as follows:

- Initialize the closed-loop system using MPC, at any initial condition x_0 within Ω , and start monitoring the growth of all the Lyapunov functions whose corresponding stability regions contain the state.
- The supervisor disregards (i.e., permanently stops monitoring) any CLF whose value ceases to decay (i.e. any CLF for which the condition of Eq.4.16 holds) by setting the corresponding $T_k^*(t)$ to zero for all future times.
- As the closed-loop trajectory continues to evolve, the supervisor includes in the pool of Lyapunov functions it monitors any CLF for which the state enters the corresponding stability region, provided that this CLF has not been discarded at some earlier time.
- Continue implementing MPC as long as at least one of the CLFs being monitored is decreasing and the MPC algorithm yields a solution.
- If at any time, there is no active CLF (this corresponds to T^f), or if the MPC algorithm fails to prescribe any control move (this corresponds to T^i), switch to the bounded controller whose stability region contains the state at this time, else the MPC controller stays in the closed-loop system.

Remark 4.13: The purpose of permanently disregarding a given CLF, once its value begins to increase, is to avoid (or “break”) a potential perpetual cycle in which the value of such a CLF increases (at times when other CLFs are decreasing) and then decreases (at times when other CLFs are increasing) and keeps repeating this pattern without decaying to zero. The inclusion of such a CLF among the pool of active CLFs used to decide whether MPC should be kept active, would result in MPC staying in the closed-loop indefinitely, but without actually converging to the origin (i.e. only boundedness of the closed-loop state can be established but not asymptotic stability). Instead of

disregarding, for all future times, a CLF that begins to increase at some point, however, an alternative strategy is to consider such a CLF in the supervisory decision-making only if its value, at the current state, falls below the value from which it began to increase. This idea is similar to the one used in the stability analysis of switched systems using multiple Lyapunov functions (MLFs) (e.g., see [50, 67]). The greater CLF-monitoring flexibility resulting from this policy can increase the likelihood of MPC implementation and enhanced closed-loop performance, while still guaranteeing asymptotic stability.

Remark 4.14: The switching schemes proposed in Theorems 1 and 2 can be further generalized to allow for multiple switchings between MPC and the bounded controllers. For example, if a given MPC is initially found infeasible (e.g., due to some terminal equality constraints), the bounded controller can be activated initially, but need not stay in the closed-loop system for all future times. Instead, it could be employed only until it brings the closed-loop trajectory to a point where MPC becomes feasible, at which time MPC can take over (see section 4 for illustrations of this scenario). This scheme offers the possibility of further enhancement in the closed-loop performance by implementing MPC for all the times that it is feasible (instead of using the bounded controller). If MPC runs into any feasibility or stability problems, then the bounded controller can be re-activated. Chattering problems, due to back and forth switching, are avoided by allowing only a finite number of switches over any finite time-interval.

Remark 4.15: Note that the implementation of the switching schemes proposed in Theorems 1 and 2 can be easily adapted to the case when actuator dynamics are not negligible and, consequently, a sudden change in the control action (resulting from switching between controllers) may not be achieved instantaneously. A new estimate of the stability region under the bounded controllers can be generated by sufficiently “stepping back” from the boundary of the original stability region estimate, in order to prevent the escape of the closed-loop trajectory as a result of the implementation

of possibly incorrect control action for some time (due to the delay introduced by the actuator dynamics). The switching schemes can then be applied as described in Theorems 1 and 2, using the revised estimate of the stability region for the family of bounded controllers.

Remark 4.16: When the proposed hybrid control schemes are applied to a process on-line, state measurements are typically available only at discrete sampling instants (and not continuously). In this case, the restricted access that the supervisor has to the state evolution between sampling times can lead to the possibility that the closed-loop state trajectory under MPC may leave the stability region without being detected, particularly if the initial condition is close to the boundary of the stability region and/or the sampling period is too large. In such an event, switching to the bounded controller at the next sampling time may be too late to recover from the instability of MPC. To guard against this possibility, the switching rules in Theorems 1 and 2 can be modified by restricting the implementation of MPC within a subset of the stability region, computed such that, starting from this subset, the system trajectory is guaranteed to remain within the stability region after one sampling period. Explicit estimates of this subset, which is parameterized by the sampling period, can be readily obtained off-line by computing (or estimating) the time-derivative of the Lyapunov function under the maximum allowable control action and then integrating both sides of the resulting inequality over one sampling period. Note that this computation of a “worst-case” estimate does not require knowledge of the solution of the closed-loop system.

Remark 4.17: The hybrid control schemes proposed in this paper can be extended to deal with the case when both input and state constraints are present. In one possible extension, state constraints would be incorporated directly as part of the constrained optimization problem that yields the model predictive control law. In addition, estimates of the stability regions for the bounded controllers would be obtained

by intersecting the region described by Eq.4.5 with the region described by the state constraints, and computing the largest invariant subset within the intersection. Using these estimates (which now account for both input and state constraints), implementation of the switching schemes can proceed following the same logic outlined for each case.

4.3 Application to chemical process examples

In this section, we present two simulation studies of chemical process examples to demonstrate the implementation of the proposed hybrid predictive control structure and evaluate its effectiveness.

4.3.1 Application to a chemical reactor example

We consider a continuous stirred tank reactor where an irreversible, first-order exothermic reaction of the form $A \xrightarrow{k} B$ takes place. The inlet stream consists of pure A at flow rate F , concentration C_{A0} and temperature T_{A0} . Under standard modeling assumptions, the mathematical model for the process takes the form

$$\begin{aligned}\dot{C}_A &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A \\ \dot{T}_R &= \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{UA}{\rho c_p V}(T_c - T)\end{aligned}\tag{4.19}$$

where C_A denotes the concentration of the species A , T_R denotes the temperature of the reactor, T_c is the temperature of the coolant in the surrounding jacket, U is the heat-transfer coefficient, A is the jacket area, V is the volume of the reactor, k_0 , E , ΔH are the pre-exponential constant, the activation energy, and the enthalpy of the reaction, c_p and ρ , are the heat capacity and fluid density in the reactor. The values of all process parameters can be found in Table 4.1. At the nominal operating condition of $T_c^{nom} = 302 \text{ K}$, the system has three equilibrium points, one of which is

unstable. The control objective is to stabilize the reactor at the unstable equilibrium point $(C_A^s, T_R^s) = (0.52, 398.9)$ using the coolant temperature, T_c , as the manipulated input with constraints: $275\text{ K} \leq T_c \leq 370\text{ K}$.

Table 4.1: Process parameters and steady-state values for the reactor of Eq.4.19.

V	$=$	100.0	L
E/R	$=$	8000	K
C_{A0}	$=$	1.0	mol/L
T_{A0}	$=$	400.0	K
ΔH	$=$	-2.0×10^5	J/mol
k_0	$=$	4.71×10^8	min^{-1}
c_p	$=$	1.0	$J/g.K$
ρ	$=$	1000.0	g/L
UA	$=$	1.0×10^5	$J/min.K$
F	$=$	100.0	L/min
C_A^s	$=$	0.52	mol/L
T_R^s	$=$	398.97	K
T_c^{nom}	$=$	302	K

Defining $x = [x_1 \ x_2]' = [(C_A - C_A^s) \ (T_R - T_R^s)]'$ and $u = T_c - T_c^{nom}$, the process model of Eq.4.19 can be written in the form of Eq.4.1. Defining an auxiliary output, $y = h(x) = x_1$ (for the purpose of designing the controller), and using the invertible coordinate transformation: $\xi = [\xi_1 \ \xi_2]' = T(x) = [x_1 \ f_1(x)]'$, where $f_1(x) = \dot{x}_1$, the system of Eq.4.19 can be transformed into the following partially linear form

$$\dot{\xi} = A\xi + bl(\xi) + b\alpha(\xi)u \quad (4.20)$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b = [0 \ 1]'$, $l(\xi) = L_f^2 h(T^{-1}(\xi))$ is the second-order Lie derivative of $h(\cdot)$ along the vector field $f(\cdot)$, $\alpha(\xi) = L_g L_f h(T^{-1}(\xi))$ is the mixed order Lie derivative. The system of Eq.4.20 will be used to design the bounded controllers and compute their estimated regions of stability. A common choice of CLFs for this system is quadratic functions of the form, $V_k = \xi' P_k \xi$, where the positive-definite matrix P_k is chosen to satisfy the Riccati matrix inequality: $A' P_k + P_k A - P_k b b' P_k < 0$.

The following matrices

$$P_1 = \begin{bmatrix} 1.45 & 1.0 \\ 1.0 & 1.45 \end{bmatrix}, P_2 = \begin{bmatrix} 0.55 & 0.1 \\ 0.1 & 0.55 \end{bmatrix}, P_3 = \begin{bmatrix} 8.02 & 3.16 \\ 3.16 & 2.53 \end{bmatrix} \quad (4.21)$$

were used to construct a family of three CLFs and three bounded controllers, and compute their stability region estimates, Ω_k , $k = 1, 2, 3$, in the ξ -coordinate system. The corresponding stability regions in the (C_A, T_R) coordinate system are then computed using the transformation $\xi = T(x)$ defined earlier. The union of these regions, Ω' , is shown in Figure 4.3.

For the design of the predictive controller, a linear MPC formulation (based on the linearization of the process model around the unstable equilibrium point) with terminal equality constraints, $x(t+T) = 0$, is chosen for the sake of illustration (other MPC formulations that use terminal penalties, instead of terminal equality constraints, could also be used). The parameters in the objective function of Eq.4.10 are chosen as $Q = qI$, with $q = 1$, $R = rI$, with $r = 1.0$, and $F = 0$. We also choose a horizon length of $T = 0.25$ in implementing the MPC controller. The resulting quadratic program is solved using the MATLAB subroutine QuadProg, and the set of nonlinear ODEs is integrated using the MATLAB solver ODE45.

As shown by the solid trajectory in Figure 4.3, starting from the initial condition $[C_A(0) \ T_R(0)]' = [0.46 \ 407.0]'$, MPC using a horizon length of $T = 0.25$ yields a feasible solution, and when implemented in the closed-loop, stabilizes the nonlinear closed-loop system. The corresponding state and input profiles are shown in Figures 4.4(a)-(c). Starting from the initial condition $[C_A(0) \ T_R(0)]' = [0.45 \ 385.0]'$ (dashed lines in Figures 4.3-4.4), however, the linear MPC controller is infeasible. Recognizing that the initial condition is within the stability region estimate, Ω'_3 , the supervisor implements the third bounded controller, while continuously checking feasibility of MPC. At $t = 0.4$, the predictive controller becomes feasible and, therefore, the supervisor switches to MPC and keeps monitoring the evolution of V_3 . In this case, the value of V_3 keeps decreasing and MPC stays in the closed-loop for all future times, thus asymp-

totically stabilizing the nonlinear plant. Note that, from the same initial condition, if the horizon length in the MPC is increased to $T = 0.5$, MPC yields a feasible solution and, when implemented, asymptotically stabilizes the closed-loop system (dotted lines in Figures 4.3-4.4). However, the initial feasibility, based on the linearized model, does not necessarily imply that the predictive controller will be stabilizing, or for that matter even feasible at future times. Furthermore, this value of the horizon length (which yields a feasible solution) could not be determined *a priori* (without actually solving the optimization problem with the given initial condition), and stability of the closed-loop could not be ascertained, *a priori*, without running the closed-loop simulation in its entirety.

While our particular choice of implementing linear MPC (using the linearized model) facilitates implementation by making the optimization problem more easily solvable, the stabilizability of a given initial condition is limited, not only by the possibility of insufficient horizon length, but also by the linearization procedure itself. To demonstrate this, we consider an initial condition, $[C_A(0) \ T_R(0)]' = [0.028 \ 484.0]'$, belonging to the stability region Ω' , (dash-dotted lines in Figures 4.3-4.4). For this initial condition, linear MPC is found infeasible, no matter how large T is chosen to be, suggesting that this initial condition is outside the feasibility region based on the linear model. Therefore, using the Lyapunov function, V_2 , (since the initial condition belongs to Ω'_2), the supervisor activates the second bounded controller which brings the state trajectory closer to the desired equilibrium point, while continuously checking feasibility of linear MPC. At $t = 1.925$, the predictive controller with $T = 0.5$ is found to be feasible and is, therefore, employed in the closed-loop system to asymptotically stabilize the closed-loop system.

To demonstrate some of the performance benefits of using the more flexible switching rules in Theorem 4.2, we consider the same control problem, described above, with relaxed constraints on the manipulated input: $250 \text{ K} \leq T_c \leq 500 \text{ K}$. Using these

constraints, a new set of four bounded controllers are designed using a family of four CLFs of the form $V_k = \xi' P_k \xi$, $k = 1, 2, 3, 4$, where

$$P_1 = \begin{bmatrix} 1.03 & 0.32 \\ 0.32 & 3.26 \end{bmatrix}, P_2 = \begin{bmatrix} 0.4 & 0.32 \\ 0.32 & 1.28 \end{bmatrix}, P_3 = \begin{bmatrix} 1.45 & 1.0 \\ 1.0 & 1.45 \end{bmatrix}, P_4 = \begin{bmatrix} 4.78 & 2.24 \\ 2.24 & 2.14 \end{bmatrix} \quad (4.22)$$

The stability region estimates of the controllers, Ω_k , $k = 1, 2, 3, 4$, are depicted in Figure 4.5. The relaxed input constraints are also incorporated in the design of the predictive controller, using the same MPC formulation employed in the preceding simulations. Starting from the initial condition $[C_A(0) \ T_R(0)]' = [0.75 \ 361.0]'$ (Figures 4.6(a)-(c)), the predictive controller, with $T = 0.1$, does not yield a feasible solution and, therefore, the supervisor implements the first bounded controller, using V_1 , instead. At $t = 0.95$, however, MPC yields a feasible solution and, therefore, the supervisor switches to MPC. Even though $\dot{V}_1 > 0$ at this time, recognizing the fact that the state at this time ($[0.84 \ 363.3]'$, denoted by $\triangle$ in Figure 4.5) belongs to Ω'_2 and that $\dot{V}_2 < 0$, the supervisor continues to implement MPC while monitoring $\dot{V}_2$ (instead of V_1). At $t = 1.1$, the supervisor detects that $\dot{V}_2 > 0$. However, the state at this time ($[0.84 \ 378.0]'$, denoted by $\diamond$ in Figure 4.5) is within Ω'_3 where $\dot{V}_3 < 0$. Therefore, the supervisor continues the implementation of MPC while monitoring V_3 . From this point onwards, V_3 continues to decay monotonically and MPC is implemented in the closed-loop system for all future times to achieve asymptotic stability. Note that the switching scheme of Theorem 1 would have dictated a switch back to the first bounded controller at $t = 0.95$ and would not have allowed for MPC to be implemented in closed-loop, leading to a total cost of $J = 1.81 \times 10^6$ for the objective function of Eq.4.10. The switching rules in Theorem 2, on the other hand, allow the implementation of MPC for all the times that it is feasible, leading to a lower total cost of $J = 1.64 \times 10^5$, while guaranteeing, at the same time, closed-loop stability.

4.3.2 Application to a continuous crystallizer example

We consider a continuous crystallizer described by a fifth-order moment model of the following form:

$$\begin{aligned}
 \dot{x}_0 &= -x_0 + (1 - x_3)Da \exp\left(\frac{-F}{y^2}\right) \\
 \dot{x}_1 &= -x_1 + yx_0 \\
 \dot{x}_2 &= -x_2 + yx_1 \\
 \dot{x}_3 &= -x_3 + yx_2 \\
 \dot{y} &= \frac{1 - y - (\alpha - y)yx_2}{1 - x_3} + \frac{u}{1 - x_3}
 \end{aligned} \tag{4.23}$$

where x_i , $i = 0, 1, 2, 3$, are dimensionless moments of the crystal size distribution, y is a dimensionless concentration of the solute in the crystallizer, and u is a dimensionless concentration of the solute in the feed (the reader may refer to [34, 64] for a detailed process description, population balance modeling of the crystal size distribution and derivation of the moments model, and to [40] for further results and references in this area). The values of the dimensionless process parameters are chosen to be: $F = 3.0$, $\alpha = 40.0$ and $Da = 200.0$. For these values, and at the nominal operating condition of $u^{nom} = 0$, the above system has an unstable equilibrium point, surrounded by a stable limit cycle. The control objective is to stabilize the system at the unstable equilibrium point, $x^s = [x_0^s \ x_1^s \ x_2^s \ x_3^s \ y^s]' = [0.0471, \ 0.0283, \ 0.0169, \ 0.0102, \ 0.5996]'$, where the superscript s denotes the desired steady-state, by manipulating the dimensionless solute feed concentration, u , subject to the constraints: $-1 \leq u \leq 1$.

To facilitate the design of the bounded controller, we initially transform the system of Eq.4.23 into the normal form. To this end, we define the auxiliary output variable, $\bar{y} = h(x) = x_0$, and introduce the invertible coordinate transformation: $[\xi' \ \eta']' = T(x) = [x_0 \ f_1(x) \ x_1 \ x_2 \ x_3]'$, where $\xi = [\xi_1 \ \xi_2]' = [x_0 \ f_1(x)]'$, $\bar{y} = \xi_1$, $f_1(x) = -x_0 + (1 - x_3)Da \exp(-F/y^2)$, and $\eta = [\eta_1 \ \eta_2 \ \eta_3]' = [x_1 \ x_2 \ x_3]'$. The state-space description of the system in the transformed coordinates takes the form:

$$\begin{aligned}
 \dot{\xi} &= A\xi + bl(\xi, \eta) + b\alpha(\xi, \eta)u \\
 \dot{\eta} &= \Psi(\eta, \xi)
 \end{aligned} \tag{4.24}$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b = [0 \ 1]'$, $l(\xi, \eta) = L_f^2 h(T^{-1}(\xi, \eta))$ is the second-order Lie derivative of the scalar function, $h(\cdot)$, along the vector field $f(\cdot)$, and $\alpha(\xi, \eta) = L_g L_f h(T^{-1}(\xi, \eta))$ is the mixed Lie derivative. The forms of $f(\cdot)$ and $g(\cdot)$ can be obtained by re-writing the system of Eq.4.23 in the form of Eq.4.1, and are omitted for brevity.

The partially-linear ξ -subsystem in Eq.4.24 is used to design a bounded controller that stabilizes the full interconnected system of Eq.4.24 and, consequently, the original system of Eq.4.23. For this purpose, a quadratic function of the form, $V_\xi = \xi' P \xi$, is used as a CLF in the controller synthesis formula of Eq.4.3, where the positive-definite matrix, P , is chosen to satisfy the Riccati matrix equality: $A'P + PA - Pbb'P = -Q$ where Q is a positive-definite matrix. An estimate of the region of constrained closed-loop stability for the full system is obtained by defining a composite Lyapunov function of the form $V_c = V_\xi + V_\eta$, where $V_\eta = \eta' P_\eta \eta$ and P_η is a positive-definite matrix, and choosing a level set of V_c , Ω_c , for which $\dot{V}_c < 0$ for all x in Ω_c . The two-dimensional projections of the stability region are shown in Figure 4.7 for all possible combinations of the system states.

In designing the predictive controller, a linear MPC formulation, with a terminal equality constraint of the form $x(t + T) = 0$, is chosen (based on the linearization of the process model of Eq.4.23 around the unstable equilibrium point). The parameters in the objective function of Eq.4.10 are taken to be: $Q = qI$, with $q = 1$, $R = rI$, with $r = 1.0$, and $F = 0$. We also choose a horizon length of $T = 0.5$ in implementing the predictive controller. The resulting quadratic program is solved using the MATLAB subroutine QuadProg, and the nonlinear closed-loop system is integrated using the MATLAB solver ODE45.

In the first set of simulation runs, we test the ability of the predictive controller to stabilize the closed-loop system starting from the initial condition, $x(0) = 0.046, 0.0277, 0.0166, 0.01, 0.58$. The result is shown by the solid lines in Figure 4.8(a)–(e) where it is seen that the predictive controller, with a horizon length of $T = 0.5$, is

able to stabilize the closed-loop system at the desired equilibrium point. Starting from the initial condition $x(0) = [0.032 \ 0.035 \ 0.010 \ 0.009 \ 0.58]'$, however, the predictive controller yields no feasible solution. If the terminal equality constraint is removed, to make MPC yield a feasible solution, we see from the dashed lines in Figure 4.8(a)–(e) that the resulting control action cannot stabilize the closed-loop system and sends the system states into a limit cycle. On the other hand, when the switching scheme of Theorem 1 is employed, the supervisor immediately switches to the bounded controller which in turn stabilizes the closed-loop system at the desired equilibrium point. This is depicted by the dotted lines in Figure 4.8(a)–(e)). The manipulated input profiles for the three scenarios are shown in Figure 4.8(f).

4.4 Concluding remarks

In this chapter, a hybrid control structure, uniting bounded control with MPC, was proposed for the stabilization of nonlinear systems with input constraints. The structure consists of: (1) a high-performance model predictive controller, (2) a family of fall-back Lyapunov-based bounded nonlinear controllers, each with a well-defined stability region, and (3) a high-level supervisor that orchestrates switching between MPC and the bounded controllers in a way that safeguards closed-loop stability in the event of MPC instability or infeasibility. The basic idea was to embed the implementation of MPC within the stability regions of the bounded controllers, and derive a set of supervisory switching rules that monitor the evolution of the closed-loop trajectory and place appropriate restrictions on the growth of the Lyapunov functions in a way that guarantees asymptotic stability for all initial conditions within the union of all stability regions of the bounded controllers. By tailoring the switching logic appropriately, the hybrid control structure was shown to provide, irrespective of the chosen MPC formulation, a safety net for the implementation of predictive control algorithms to constrained nonlinear systems. Finally, the implementation of the switching schemes

was demonstrated through applications to chemical reactor and crystallization process examples.

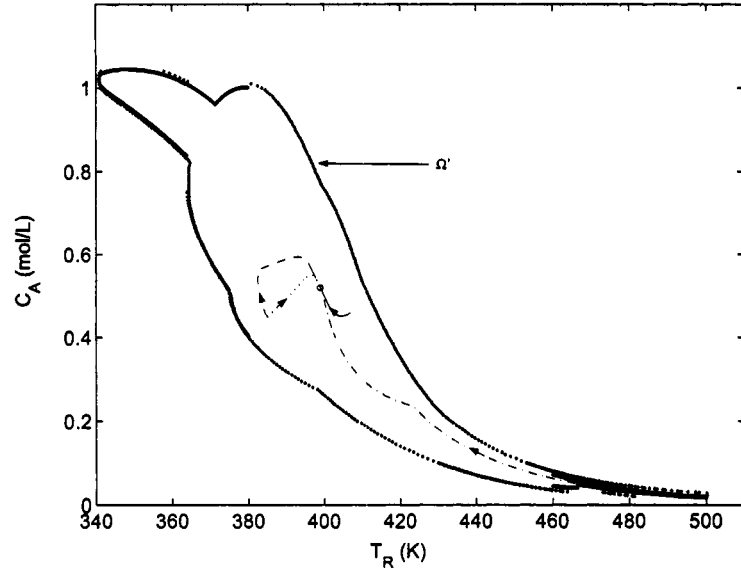


Figure 4.3: Implementation of the proposed hybrid control structure: Closed-loop state trajectory under MPC with $T = 0.25$ (solid trajectory), under the switched MPC/bounded controller(3) with $T = 0.25$ (dashed trajectory), under MPC with $T = 0.5$ (dotted trajectory), and under the switched bounded controller(2)/MPC with $T = 0.5$ (dash-dotted line).

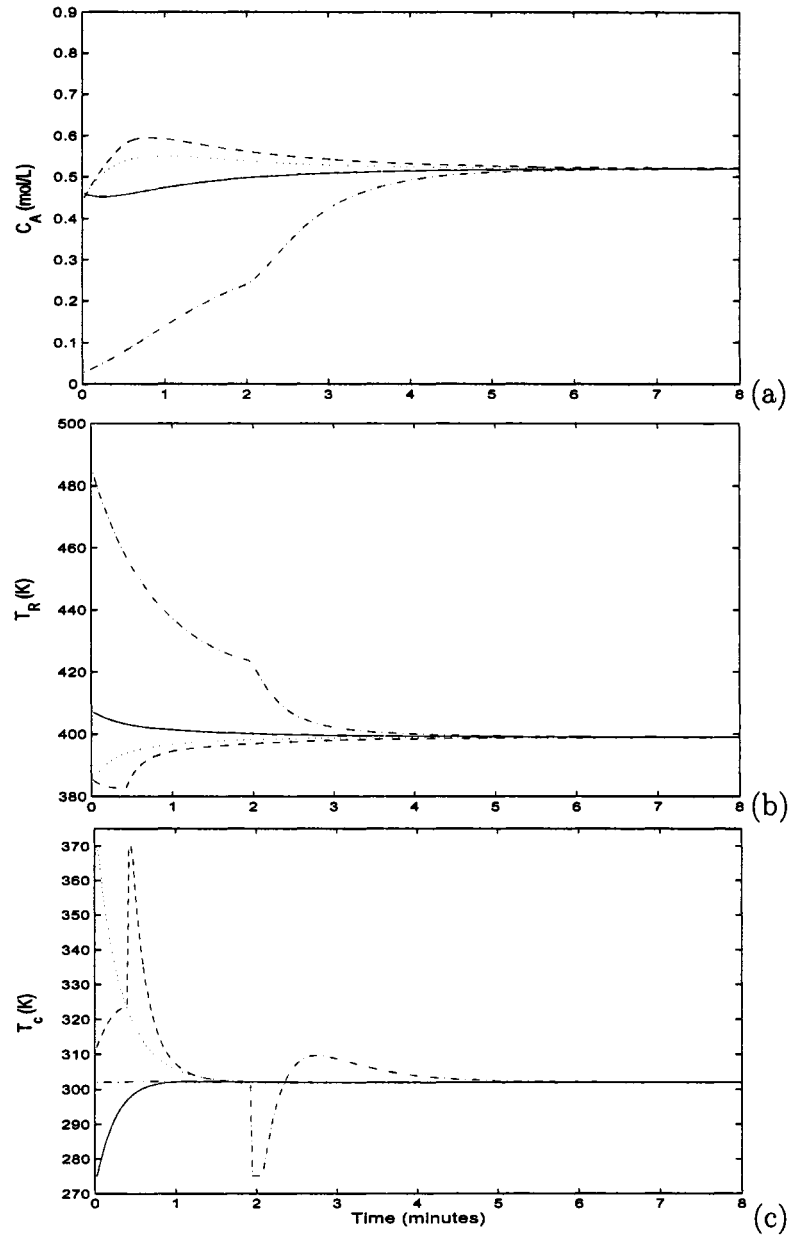


Figure 4.4: Closed-loop reactant concentration profile (a), reactor temperature profile (b) and coolant temperature profile (c) under MPC with $T = 0.25$ (solid trajectory), under the switched MPC/bounded controller(3) with $T = 0.25$ (dashed trajectory), under MPC with $T = 0.5$ (dotted trajectory), and under the switched bounded controller(2)/MPC with $T = 0.5$ (dash-dotted line).

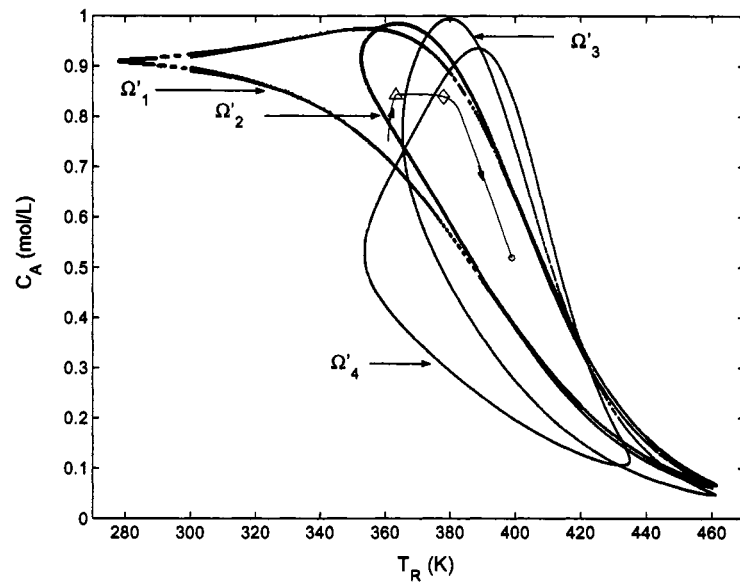


Figure 4.5: Closed-loop state trajectory under the switching rules of Theorem 2 with $T = 0.1$ for MPC.

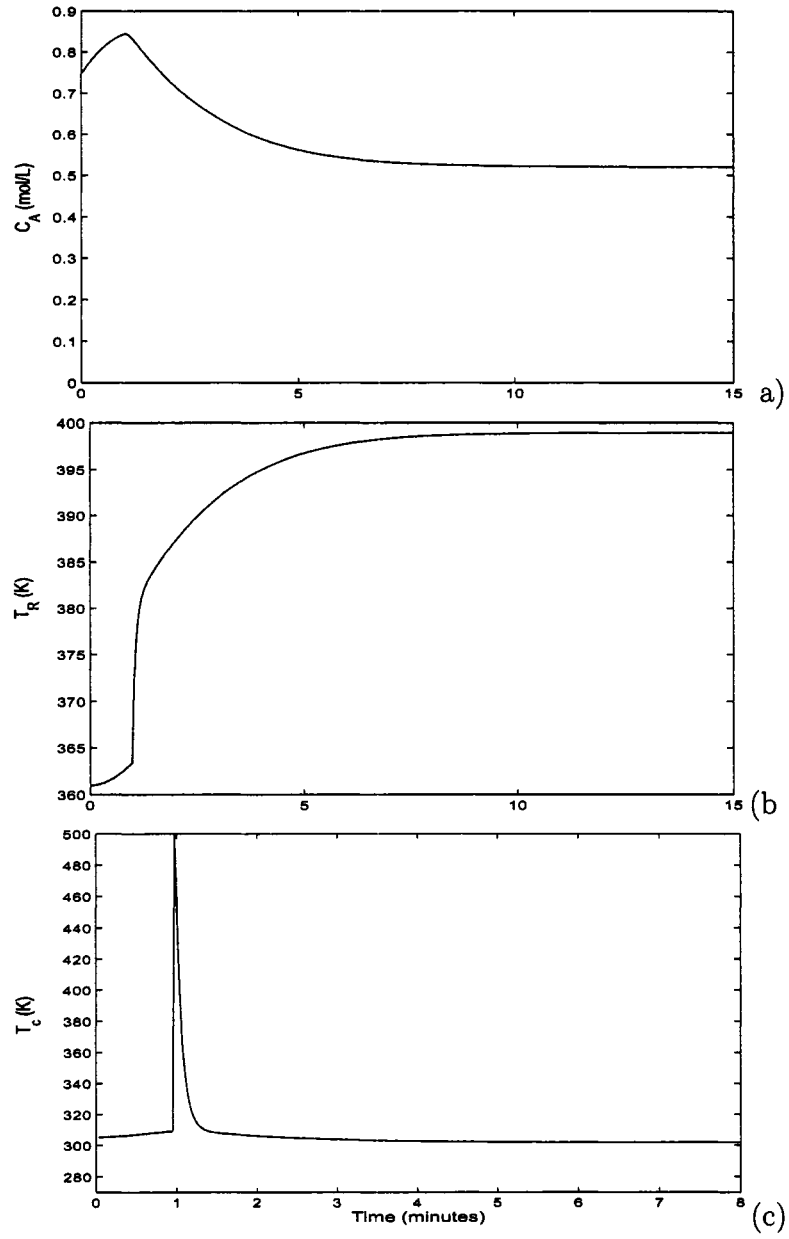


Figure 4.6: Closed-loop reactant concentration profile (a), reactor temperature profile (b) and coolant temperature profile (c) under the switching scheme of Theorem 2 with $T = 0.1$ for MPC.

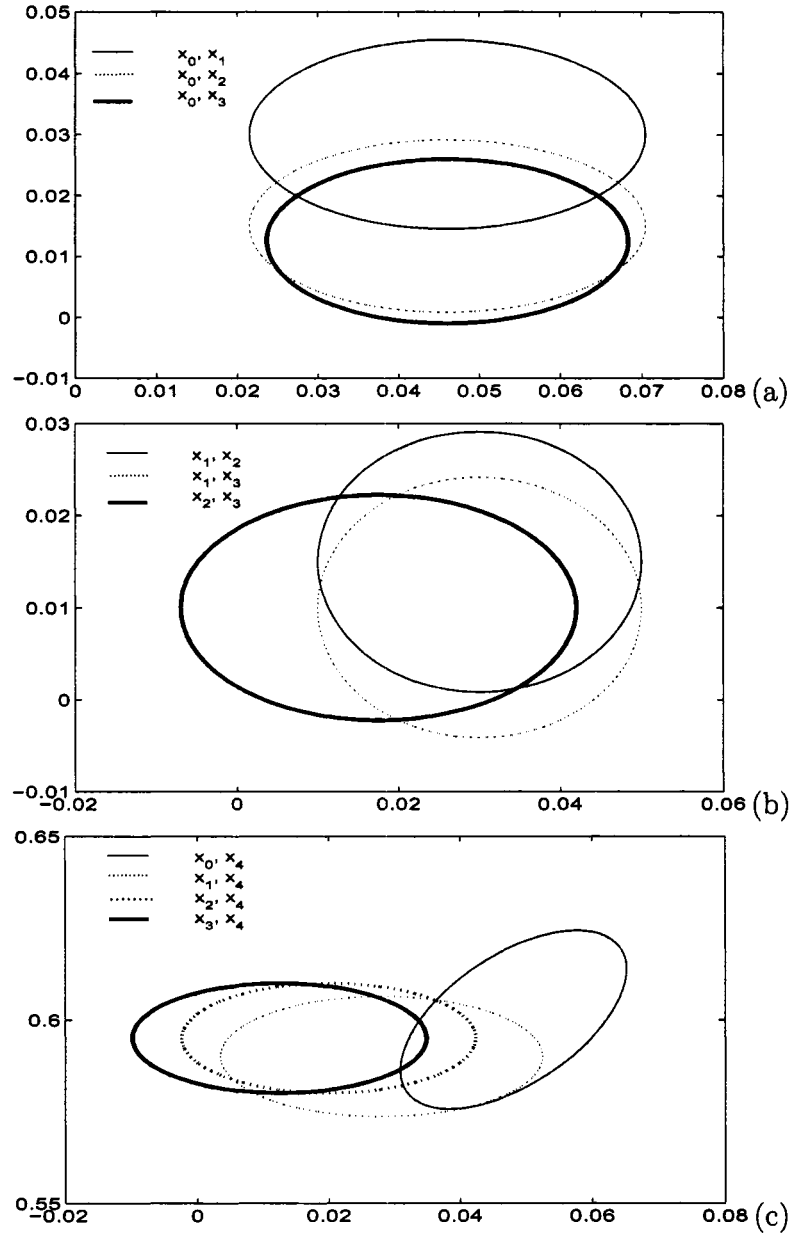


Figure 4.7: Implementation of the proposed hybrid control structure to a continuous crystallizer: two-dimensional projections of the stability region for the 10 distinct combinations of the process states.

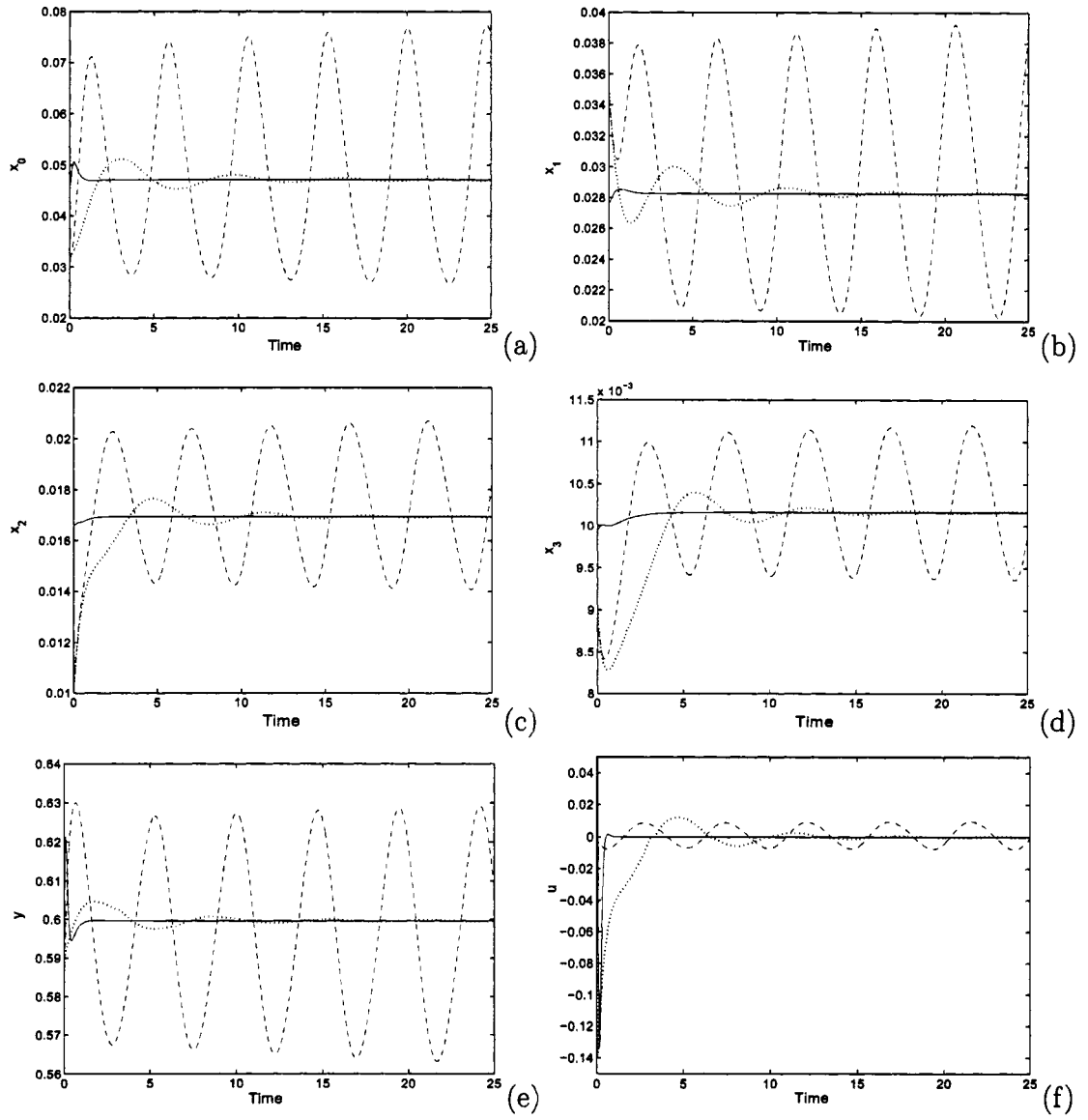


Figure 4.8: Closed-loop profiles of the dimensionless crystallizer moments (a)-(d), the solute concentration in the crystallizer (e) and the manipulated input (f) under MPC with stability constraints (solid line), under MPC without terminal constraints (dashed line), and using the switching scheme of Theorem 1 (dotted line).

Chapter 5

Robust Hybrid Predictive Control of Nonlinear Systems

5.1 Introduction

Stabilization of nonlinear systems subject to uncertainty and manipulated input constraints is a fundamental control problem that has been the subject of significant research work. One of the control methods suited for handling constraints within an optimal control setting is model predictive control (MPC). Numerous research studies have investigated the stability properties of model predictive controllers for systems without uncertainty (see, for example, the review paper [141]).

The problem of analysis and design of predictive controllers for uncertain linear systems has been extensively investigated (see [19, 141] for surveys of results in this area). For uncertain nonlinear systems, the problem of robust MPC design continues to be an area of ongoing research (see, for example, [149, 131]). While min-max formulations provide a natural setting within which to address this problem, computational problems with these approaches are well known, and stem in part from the nonlinearity of the model which typically makes the optimization problem non-convex and in part from performing the min-max optimization over the non-convex problem. Furthermore

the robust stability guarantee in various MPC formulations (with or without stability conditions, and with or without robustness considerations) is contingent upon the assumption of initial feasibility, and the set of initial conditions starting from where feasibility and stability is guaranteed is not explicitly characterized.

Stabilizing control laws that provide explicitly-defined regions of attraction for the closed-loop system have been developed using Lyapunov techniques; the reader may refer to [114] for a survey of results in this area. In [69, 70], a class of Lyapunov-based bounded robust nonlinear controllers, inspired by the results on bounded control originally presented in [127], was developed. While these Lyapunov-based controllers have well-characterized stability and constraint-handling properties, they cannot, in general, be designed to be optimal with respect to a pre-specified, arbitrary cost function.

From the above discussion, it is clear that both MPC and Lyapunov-based analytic control approaches possess, by design, their own, distinct stability and optimality properties. Recently, we proposed a hybrid predictive control structure that employs switching between bounded control and MPC for the stabilization of linear systems under state [75] and output feedback [145], and in [74] for nonlinear systems, subject to input constraints while providing *a priori* (off-line) the set of initial conditions, for which closed-loop stability is guaranteed (through bounded control).

The presence of uncertainty, however, may alter the stability region of the nominal controllers (designed without taking the uncertainty into account) or even render the closed-loop system unstable. Furthermore, simply replacing the fall-back controller by an appropriate robust controller and implementing the same switching logics proposed in [75, 74] may lead to switching that is too conservative, resulting in the implementation of the fall-back controller for almost all times. Motivated by these considerations, we consider in this chapter nonlinear systems with input constraints and uncertain variables, and develop a robust hybrid predictive control structure. The proposed method provides a safety net for the implementation of any available MPC formulation, de-

signed with or without taking uncertainty into account, and allows for an explicit characterization of the set of initial conditions starting from where the closed-loop system is guaranteed to be stable. The key idea is to use a Lyapunov-based robust controller, for which an explicit characterization of the closed-loop stability region can be obtained, to provide a stability region within which MPC can be implemented. A set of switching laws are designed that exploit the performance of MPC whenever possible, while using the bounded controller to provide the stability guarantees. The implementation and efficacy of the robust hybrid predictive control structure are demonstrated through a chemical process example.

5.2 Preliminaries

We consider nonlinear systems with uncertain variables and input constraints, described by:

$$\dot{x} = f(x) + G(x)u + W(x)\theta(t), \quad u \in \mathbf{U} \quad (5.1)$$

where $x \in \mathbb{R}^n$ denotes the vector of state variables, $u \in \mathbb{R}^m$ denotes the vector of constrained manipulated inputs, taking values in a nonempty convex subset $\mathbf{U}$ of $\mathbb{R}^m$, where $\mathbf{U} = \{u \in \mathbb{R}^m : \|u\| \leq u_{max}\}$, $\|\cdot\|$ is the Euclidean norm of a vector, $u_{max} > 0$ is the magnitude of input constraints, and $\theta(t) = [\theta^1(t) \cdots \theta^q(t)]^T \in \Theta \subset \mathbb{R}^q$ denotes the vector of uncertain (possibly time-varying) but bounded variables taking values in a nonempty compact convex subset of $\mathbb{R}^q$ and $f(0) = 0$. The vector function $f(x)$, the matrices $G(x) = [g^1(x) \cdots g^m(x)]$ and $W(x) = [w^1(x) \cdots w^q(x)]$, where $g^i(x) \in \mathbb{R}^n$, $i = 1 \cdots m$, and $w^i(x) \in \mathbb{R}^n$, $i = 1 \cdots q$, are assumed to be sufficiently smooth on their domains of definition. The notation $L_f h$ denotes the standard Lie derivative of a scalar function $h(\cdot)$ with respect to the vector function $f(\cdot)$, the notation $x(T^-)$ denotes the limit of the trajectory $x(t)$ as T is approached from the left, i.e., $x(T^-) = \lim_{t \rightarrow T^-} x(t)$ and the notation $\partial\Omega$ is used to denote the boundary of a closed set, Ω . Throughout the chapter, we assume that for any $u \in \mathbf{U}$ the solution of the system of Eq.5.1 exists and is continuous for all t , and we focus on the state feedback problem where measurements

of the entire state, $x(t)$, are assumed to be available for all t .

5.2.1 Bounded robust Lyapunov-based control

Referring to the system of Eq.5.1, we assume that the uncertain variable term, $W(x)\theta$, is non-vanishing (in the sense that the origin is no longer the equilibrium point of the uncertain system) and that a robust control Lyapunov function (RCLF [80]), V exists. Consider also, the bounded state feedback control law (see [69, 70, 66] for details on controller design):

$$u = - \left(\frac{\alpha(x) + \sqrt{(\alpha_1(x))^2 + (u_{max}\beta(x))^4}}{(\beta(x))^2 [1 + \sqrt{1 + (u_{max}\beta(x))^2}]} \right) (L_G V)^T \quad (5.2)$$

when $L_G V \neq 0$ and $u = 0$ when $L_G V = 0$, where

$\alpha(x) = L_f V + (\rho\|x\| + \chi\theta_b\|L_W V\|) \left(\frac{\|x\|}{\|x\| + \phi} \right)$,
 $\alpha_1(x) = L_f V + \rho\|x\| + \chi\theta_b\|L_W V\|$, $\beta(x) = \|(L_G V)^T\|$, $L_G V = [L_{g^1} V \ \cdots \ L_{g^m} V]$ and $L_W V = [L_{w^1} V \ \cdots \ L_{w^q} V]$ are row vectors, θ_b is a positive real number such that $\|\theta(t)\| \leq \theta_b$, for all $t \geq 0$, and ρ , χ and ϕ are adjustable parameters that satisfy $\rho > 0$, $\chi > 1$ and $\phi > 0$. Let Π be the set defined by $\Pi(\theta_b, u_{max}) = \{x \in \mathbb{R}^n : \alpha_1(x) \leq u_{max}\beta(x)\}$ and assume that $\Omega := \{x \in \mathbb{R}^n : V(x) \leq c^{max}\} \subseteq \Pi(\theta_b, u_{max})$ for some $c^{max} > 0$. Then, given any positive real number, d , such that:

$$\mathbb{D} := \{x \in \mathbb{R}^n : \|x\| \leq d\} \subset \Omega \quad (5.3)$$

and for any initial condition $x_0 \in \Omega$, it can be shown that there exists a positive real number ϵ^* such that if $\phi/(\chi - 1) < \epsilon^*$, the states of the closed-loop system of Eqs.5.1–5.2 satisfy $x(t) \in \Omega \ \forall t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$.

Remark 5.1: Referring to the above controller design, it is important to make the following remarks. First, a general procedure for the construction of RCLFs for nonlinear systems of the form of Eq.5.1 is currently not available. Yet, for several classes of nonlinear systems that arise commonly in the modeling of engineering applications, it is possible to exploit system structure to construct RCLFs. For example, for feedback

linearizable systems, quadratic Lyapunov functions can be chosen as candidate RCLFs and can be made RCLFs with appropriate choice of the function parameters based on the process parameters (see, for example, [80]). Also, for nonlinear systems in strict feedback form, backstepping techniques can be employed for the construction of RCLFs [80]. Second, given that an RCLF, V , has been obtained for the system of Eq.5.1, it is important to clarify the essence and scope of the additional assumption that there exists a level set, Ω , of V that is contained in Π . Specifically, the assumption that the set, Π , contains an invariant subset around the origin, is necessary to guarantee the existence of a set of initial conditions for which closed-loop stability is guaranteed (note that even though $\dot{V} < 0 \forall x \in \Pi \setminus \mathcal{D}$, there is no guarantee that trajectories starting within Π remain within Π for all times). Moreover, the assumption that Ω is a level set of V is made only to simplify the construction of Ω . This assumption restricts the applicability of the proposed control method because a direct method for the construction of an RCLF with level sets contained in Π is not available. However, the proposed control method remains applicable if the invariant set Ω is not a level set of V but can be constructed in some other way (which, in general, is a difficult task).

Remark 5.2: Regarding the choice of the above controller design, we note that the problem of designing control laws that guarantee stability in the presence of input constraints has been extensively studied (see, for example, [127, 221, 125, 69, 70]). The bounded robust controller design of Eq.5.2, proposed in [69, 70] (inspired by the results on bounded control in [127] for systems without uncertainty) is an example of a controller design that (1) guarantees robust stability in the presence of constraints, and (2) allows for an explicit characterization of the closed-loop stability region. The results of this chapter are not limited to this particular choice of controllers and any other robust controller that satisfies (1) and (2) above, can be used.

5.2.2 Model Predictive Control

The model predictive control approach provides a framework with the ability to handle, among other issues, multi-variable interactions, constraints on controls, and optimization requirements, all in a consistent, systematic manner. For the purpose of illustrating our results, we describe here a symbolic MPC formulation that incorporates most existing MPC formulations as special cases. This is not a new formulation of MPC; the general description is only intended for the purpose of highlighting the fact that the robust hybrid predictive control structure (to be proposed in the next section) can incorporate any available MPC formulation. In MPC, the control action at time t is conventionally obtained by solving, on-line, a finite horizon optimal control problem. The generic form of the optimization problem can be described as:

$$\begin{aligned} u(\cdot) &= \operatorname{argmin}\{\max\{J_s(x, t, u(\cdot)) | \theta(\cdot) \in \Theta\} | u(\cdot) \in S\} \\ \text{s.t. } \dot{x}(t) &= f(x(t)) + G(x)u + W(x)\theta(t) \\ x(0) &= x_0, \quad x(t+T) \in \Omega_{MPC}(x, t, \theta) \end{aligned} \quad (5.4)$$

where $J_s(x, t, u(\cdot)) =$

$$\int_t^{t+T} (x'(s)Qx(s) + u'(s)Ru(s))ds + F(x(t+T)) \quad (5.5)$$

and $S = S(t, T)$ is the family of piecewise continuous functions, with period Δ , mapping $[t, t+T]$ into the set of admissible controls, T is the horizon length and θ is the bounded uncertainty assumed to belong to a set Θ . A control $u(\cdot)$ in S is characterized by the sequence $\{u[k]\}$ where $u[k] := u(k\Delta)$ and satisfies $u(t) = u[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. J_s is the performance index, R and Q are strictly positive definite, symmetric matrices and the function $F(x(t+T))$ represents a penalty on the states at the end of the horizon. The maximization over θ may not be carried out if the MPC version used is not a min-max type of formulation. The set $\Omega_{MPC}(x, t, \theta)$ could be a fixed, terminal set, or may represent inequality constraints (as in the case of MPC formulations that require some norm of the state, or a Lyapunov function for the system, to decrease at the end of the horizon). This stability constraint may or may not account for uncertainty. The stability guarantees in MPC formulations (with or without explicit

stability conditions, and with or without robustness considerations, and whether or not it is a min-max type of formulation) are dependent on the assumption of initial feasibility. Obtaining an explicit characterization of the closed-loop stability region of the predictive controller under uncertainty and constraints remains a difficult task.

5.3 Robust Hybrid Predictive Control Structure

We first cast the system of Eq.5.1 as a switched system of the form:

$$\dot{x} = f(x) + G(x)u_{i(t)} + W(x)\theta(t), \quad \|u_{i(t)}\| \leq u_{max} \quad (5.6)$$

where $i : [0, \infty) \rightarrow \{1, 2\}$ is the switching signal which is assumed to be a piecewise continuous (from the right) function of time, implying that only a finite number of switches between the two controllers is allowed on any finite-time interval. The index, $i(t)$, represents a discrete state that indexes the control input, u , with the understanding that $i(t) = 1$ if and only if $u_i(x(t)) = M_s(x(t))$ (i.e., MPC is used) and $i(t) = 2$ if and only if $u_i(x(t)) = b(x(t))$ (i.e., bounded control is used). Our goal is to construct a switching law that provides the supervisor with the switching times that ensure stabilizing transitions between the two controllers. We first present a scheme in Theorem 5.1 that focuses on the issue of robust closed-loop stability, and then in Theorem 5.2, provide more flexible switching rules that, in addition, enhance the overall closed-loop performance. The proofs of both theorems are given after the theorems.

5.3.1 Stability-based controller switching

Theorem 5.1: *Consider the switched nonlinear system of Eq.5.6, the model predictive controller of Eqs.5.4–5.5 and the bounded controller of Eq.5.2. Let $x(0) = x_0 \in \Omega$, and initially set $T_s = T_D = T_{inf} = \infty$. At the earliest time $t \geq 0$ for which the closed-loop state under MPC satisfies $V(x(t^-)) = c^{max}$ set $T_s = t$. At the earliest time for which the closed-loop state under MPC satisfies $\|x(t)\| \leq d$ where d was defined in Eq.5.3, set $T_D = t$. Finally, at the earliest time t that MPC is infeasible, set $T_{inf} = t$. Define $T_{switch} = \min\{T_s, T_D, T_{design}, T_{inf}\}$, where $0 \leq T_{design} < \infty$ is arbitrary. Then, the*

switching rule

$$i(t) = \begin{cases} 1, & 0 \leq t < T_{switch} \\ 2, & t \geq T_{switch} \end{cases} \quad (5.7)$$

guarantees that $x(t) \in \Omega \forall t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$.

Proof of Theorem 5.1: The proof uses the fact that if the state of the closed-loop system resides in Ω when the bounded controller is switched in, i.e., $x(T_{switch}) \in \Omega$, then $x(t) \in \Omega \forall t \geq T_{switch}$, and convergence to the set $\mathbb{D}$ under the bounded robust controller are guaranteed (for a proof, see [69, 70]). Therefore, we need only show that, under the switching scheme of Eq.5.7, the closed-loop state is always within Ω until the time the bounded controller is switched in. For this purpose, we enumerate the four possible values that T_{switch} could take, and show that $x(T_{switch}) \in \Omega$ in all cases.

- Case 1: If $T_{switch} = T_s$, then from the definition of T_s , $x(T_s^-) \in \Omega$. By continuity of the solution of the system of Eq.5.1 (which follows from the fact that the right hand side of Eq.5.6 is continuous in x and piecewise continuous in time, since only a finite number of switches is allowed), we have $x(T_s) \in \Omega$, i.e., $x(T_{switch}) \in \Omega$.
- Case 2: If $T_{switch} = T_D$, then from the definition of T_D , $x(T_D) \in \mathbb{D} \subset \Omega$; hence $x(T_{switch}) \in \mathbb{D} \subset \Omega$.
- Case 3: If $T_{switch} = T_{design}$, then from the definition of T_{switch} , $T_{design} \leq T_s$. To prove that $x(T_{design}) \in \Omega$, we proceed by contradiction. Assume $x(T_{design}) \notin \Omega$. Then $V(x(T_{design})) > c^{max}$ (from the definition of Ω). By continuity of the solution, continuity of $V(\cdot)$ and the fact that $V(x_0) \leq c^{max}$, there exists a time $0 \leq T'_s \leq T_{design}$ for which $V(x(T'^s_-)) = c^{max}$. Since T_s is the *earliest* time for which $V(x(t'^-)) = c^{max}$, then it must be that $T_s \leq T'_s \leq T_{design}$, which leads to a contradiction ($T_{design} \leq T_s$). Therefore, once again, $x(T_{switch}) \in \Omega$.
- Case 4: If $T_{switch} = T_{inf}$, the same argument as in Case 3 can be used (replacing T_{design} by T_{inf}) to show that $x(T_{switch}) \in \Omega$.

Remark 5.3: The hybrid predictive controller of Theorem 5.1 is designed and implemented as follows:

- Given the nonlinear system of Eq.5.1, θ_b and u_{max} , design the bounded robust controller of Eq.5.2, and calculate an estimate of its stability region Ω .
- Design/pick an MPC formulation (the MPC formulation could be min-max optimization based, linear or nonlinear, and with or without stability constraints). For convenience, we refer to the general MPC formulation of Eqs.5.4–5.5.
- Given any $x_0 \in \Omega$, check the feasibility of the optimization in Eqs.5.4–5.5 at $t = 0$, and if feasible, start implementing MPC (i.e., set $u(0) = M_s(x_0)$).
- If at any time, MPC becomes infeasible ($t = T_{inf}$), or the states of the closed-loop system approach the boundary of Ω ($t = T_s$) or the closed-loop states enter the set $\mathbb{ID}$ ($t = T_D$) then switch to the bounded controller, else keep MPC active in the closed-loop system until a time T_{design} .
- Switch to the bounded robust controller at T_s , T_D , T_{design} , or T_{inf} , whichever comes earliest, to achieve practical closed-loop stability.

Remark 5.4: The purpose of switching to the bounded robust controller after the time T_{design} is to ensure convergence to $\mathbb{ID}$ and avoid possible cases where the closed-loop states, under MPC, could wander inside Ω without actually converging to, and staying within, $\mathbb{ID}$. Convergence to $\mathbb{ID}$ could also be achieved (see, for example, [75, 74]), by switching to the bounded controller when $\dot{V} \geq 0$ under MPC. However, in the presence of uncertainty, such a condition might be very restrictive in the sense that it may terminate MPC implementation too early. Note that if an MPC design is used that guarantees robust stability for the uncertain nonlinear system if initially feasible, it could be implemented for all time (T_{design} can be chosen to be practically infinity) to stabilize the closed-loop system. The stability safeguards, provided by the bounded

controller, are still required because the stability of any MPC formulation, robust or otherwise, is based on the assumption of initial feasibility, which cannot be verified short of testing, via simulation, an initial condition for feasibility.

Remark 5.5: We note that while the MPC framework provides a transparent way of specifying a performance objectives, the various MPC formulations, in general, may not be optimal, and only approximate the infinite horizon optimal cost to varying degrees of success. The choice of a particular MPC design can be made entirely on the basis of the desired tradeoff between performance and computational complexity because the stability guarantees of the robust hybrid predictive controller are independent of the specific MPC formulation being used.

5.3.2 Enhancing closed-loop performance

In this section we relax the switching rules of Theorem 5.1, in order to take better advantage of the MPC performance. The relaxed switching scheme uses a family of l fall back controllers together with the MPC and allows for multiple switchings between MPC and the bounded controllers. We recast the system of Eq.5.1 as a switched system of the form:

$$\begin{aligned} \dot{x} &= f(x) + G(x)u_{i(t)} + W(x)\theta(t) \\ \|u_{i(t)}\| &\leq u_{max}, \quad i(t) \in \{1, 2, \dots, l+1\} \end{aligned} \quad (5.8)$$

where $i(t) = k$ if and only if $u_i(x(t)) = b_k(x(t))$ (i.e., the k th bounded controller is used) and $i(t) = l+1$ if and only if $u_i(x(t)) = M_s(x(t))$ (i.e., MPC is used). We also define, for the l bounded controllers, and the robust control Lyapunov functions, V_k , $k = 1, \dots, l$, $0 < \alpha < 1$, $c_{k,j}^{max} = \alpha^j c_k^{max}$, $\Omega_k^j = \{x \in \mathbb{R}^n : V_k(x) \leq c_{k,j}^{max}\}$ and $\Omega_L^j = \bigcup_{k=1}^l \Omega_k^j$, for $j = 0, \dots, N-1$ where $N < \infty$ is an integer, and $d^{max} = \max_{k=1, \dots, l} \{d_k\}$, where d_k are arbitrarily small positive numbers such that $\mathbb{D}_k = \{x \in \mathbb{R}^n : \|x\| \leq d_k\} \subset \Omega_k^{N-1}$.

Theorem 5.2: Consider the constrained switched nonlinear system of Eq.5.8, the model predictive controller of Eqs.5.4–5.5 and the l bounded controllers of the form of Eq.5.2, designed using the robust control Lyapunov functions V_k , $k = 1, \dots, l$. Consider any initial condition $x(0) \equiv x_0 \in \Omega_L^0$ and initially set $T_0 = 0$, $T_1 = T_2 = \dots = T_N = T_D = \infty$. At the earliest time, $t \geq 0$, for which the closed-loop state under MPC

satisfies $\|x(t)\| \leq d^{max}$, set $T_D = t$. At the earliest time, $t \geq 0$, under MPC such that

$$x(t) \in \partial\Omega_L^{j-1}, x(t^-) \in \Omega_L^{j-1} \text{ and } T_{j-1} < \infty \quad (5.9)$$

for some $j \in \{1, \dots, N\}$, or MPC is infeasible, set $T_j = t$. Define T_{switch}

$= \min\{T_D, T_N, T_{design}\}$, where $0 \leq T_{design} < \infty$ is arbitrary. Then the switching rule

$$i(t) = \begin{cases} l+1, & 0 \leq T_j \leq t \leq T_{j+1} \leq T_{switch}, x \in \Omega_L^j \\ a, & 0 \leq T_j \leq t \leq T_{j+1} \leq T_{switch}, x \notin \Omega_L^j \\ f, & t > T_{switch} \end{cases} \quad (5.10)$$

for some $a \in \{1, \dots, l\}$ for which $x(T_j) \in \Omega_a^{j-1}$, and for some $f \in \{1, \dots, l\}$ for which $x(T_{switch}) \in \Omega_f^0$, guarantees that $x(t) \in \Omega_L^0 \forall t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d^{max}$.

Proof of Theorem 5.2: The proof of this theorem uses the fact that if the state of the closed-loop system resides in Ω_f^0 at the time that the f^{th} bounded controller is switched in (and kept in the closed-loop for all future times), then under the bounded robust controller f , the region Ω_f^0 is invariant (for a proof, see [69, 70]), and convergence to $\mathbb{D}$ is guaranteed. We now show that for $t \leq T_{switch}$ (i.e., up until the time that the switching between the controllers is governed by Eqs.5.9-5.10), the closed-loop state, initialized within Ω_L^0 , stays within Ω_L^0 . We only need to show that the switching logic ensures that the bounded controller is switched in before the states have escaped Ω_L^0 under MPC. To this end, we first note that $c_{k,j}^{max} < c_{k,0}^{max}$, for all $k = 1, \dots, l$. Therefore $\Omega_L^j \subset \Omega_L^0$. This implies that if $x \in \Omega_L^j$ then $x \in \Omega_L^0$.

Consider any time, t , when $T_j \leq t < T_{j+1} < T_{switch}$, for some $j \geq 1$. Between T_j and T_{j+1} , we have from Eq.5.10 that MPC is switched in (and the bounded controller is switched out) only when $x(t) \in \Omega_L^j$. Now consider the time T_{j+1} at which time the bounded controller is switched back in, either due to Eq.5.9 or because MPC is infeasible. In the first case, $x(T_{j+1}) \in \Omega_L^j \subset \Omega_L^0$. In the second case, since the right hand side of the Eq.5.8 is a piecewise continuous function of time and a continuous function of its states, and the closed-loop system is initialized within Ω_L^j under MPC, any time the MPC optimization problem is infeasible, $x(t) \in \Omega_L^j \subset \Omega_L^0$. At the time $t = T_{switch}$, the state of the closed-loop system can be shown to be within Ω_L^0 following the same

line of reasoning used in the proof of Theorem 5.1, with the additional possibility of $T_{switch} = T_N$. As shown above, $x(t) \in \Omega_L^0$ for all $T_{j-1} \leq t \leq T_j$, and therefore $x \in \Omega_L^0$ for $T_{N-1} \leq t \leq T_N$ and therefore $x(T_N) \in \Omega_L^0$ implying once again that $x(T_{switch}) \in \Omega_L^0$. For all $t \geq T_{switch}$, the closed-loop state evolves under some bounded controller $u(x) = b_k(x)$, for which $x(T_{switch}) \in \Omega_k^0$. Therefore, $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$ (see [69, 70] for the proof). Note that since d^{max} is chosen to be the largest of d_k , $k = 1, \dots, l$, $\|x(t)\| \leq d_k$ implies $\|x(t)\| \leq d^{max}$; hence $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d^{max}$. This completes the proof of Theorem 5.2.

Remark 5.6: The hybrid predictive controller of Theorem 5.2 is designed and implemented as follows (see also the schematic representation in Figure 5.1 for an example when $l = N = 2$):

- For each Lyapunov function V_k , $k = 1, \dots, l$: construct a bounded controller using Eq.5.2, a set of N concentric level sets, Ω_k^j , $j = 0, \dots, N-1$, with $\Omega_k^{N-1} \subset \Omega_k^{N-2} \subset \dots \subset \Omega_k^1 \subset \Omega_k^0$, and a terminal set, $\mathbb{D}_k \subset \Omega_k^{N-1}$.
- Initialize the closed-loop system using MPC at any initial condition, x_0 , within Ω_L^0 (the union of the level sets Ω_k^0) and monitor the Lyapunov function value.
- At the earliest time that either the closed-loop state approaches the boundary of Ω_L^0 or MPC is infeasible, switch to bounded control (this corresponds to encountering T_1 ; see Figure 5.1).
- Implement any of the bounded controllers whose stability region contains the state at T_1 until the state enters Ω_L^1 (the union of the level sets Ω_k^1) after which switch back to MPC (in Figure 5.1 this happens when the state enters Ω_1^1).
- If the closed-loop state under MPC does not escape the boundary of Ω_L^1 and MPC continues to be feasible, keep MPC in the closed-loop system (this happens in Figure 5.1(a) where the state continues to stay within Ω_L^1). If the closed-loop state, however, tries to escape the boundary of Ω_L^1 (this happens in Figure 5.1(b)

when the state hits the boundary of Ω_L^1 at $t = T_2$) or MPC is infeasible again, switch to bounded control and implement it until the states are driven inside the next safety zone where MPC is re-activated and confined (note that each time MPC is switched back in, its implementation is confined within a smaller safety zone since $\Omega_L^1 \supset \Omega_L^2 \supset \dots \supset \Omega_L^{N-1}$). Repeat this process until either the state under MPC is about to escape the N -th safety zone, Ω_L^{N-1} , or the state enters the largest terminal set, $\mathbb{D}_{max}$, or a time greater than the design parameter, T_{design} , has elapsed. At the earliest time that any of these events takes place, switch permanently to any of the bounded controllers whose stability region contains the state at the time to ensure practical stability of the closed-loop system.

5.4 Application to a chemical reactor

Consider the following model of an irreversible elementary exothermic reaction of the form $A \xrightarrow{k} B$ in a well-mixed continuous stirred tank reactor:

$$\begin{aligned} V_R \frac{dC_A}{dt} &= F(C_{A0} - C_A) - k_0 \exp\left(\frac{-E}{RT}\right) C_A V \\ V_R \frac{dT}{dt} &= F(T_{A0} - T) - \frac{\Delta H}{\rho c_p} k_0 \exp\left(\frac{-E}{RT}\right) C_A V + \frac{Q}{\rho c_p} \end{aligned} \quad (5.11)$$

where C_A denotes the concentration of species A , T , V_R denote the temperature and volume of the reactor, respectively, Q denotes the rate of heat input to the reactor, k_0 , E , ΔH denote the pre-exponential constant, the activation energy, and the enthalpy of the reaction, respectively, and c_p and ρ denote the heat capacity and density of the fluid in the reactor, respectively. Table 1 lists the steady-state values and process parameters. The control objective is to regulate both the reactor temperature and reactant concentration at the (open-loop) unstable equilibrium point by manipulating both the rate of heat input/removal and the inlet reactant concentration. Defining $x_1 = C_A - C_{As}$, $x_2 = T - T_s$, $u_1 = C_{A0} - C_{A0s}$, $u_2 = Q$, $\theta_1(t) = T_{A0} - T_{A0s}$, $\theta_2(t) = \Delta H - \Delta H_n$, where the subscript s denotes the steady-state value and ΔH_n denotes the nominal value of the heat of reaction, the process model of Eq.5.12 can be cast in the form

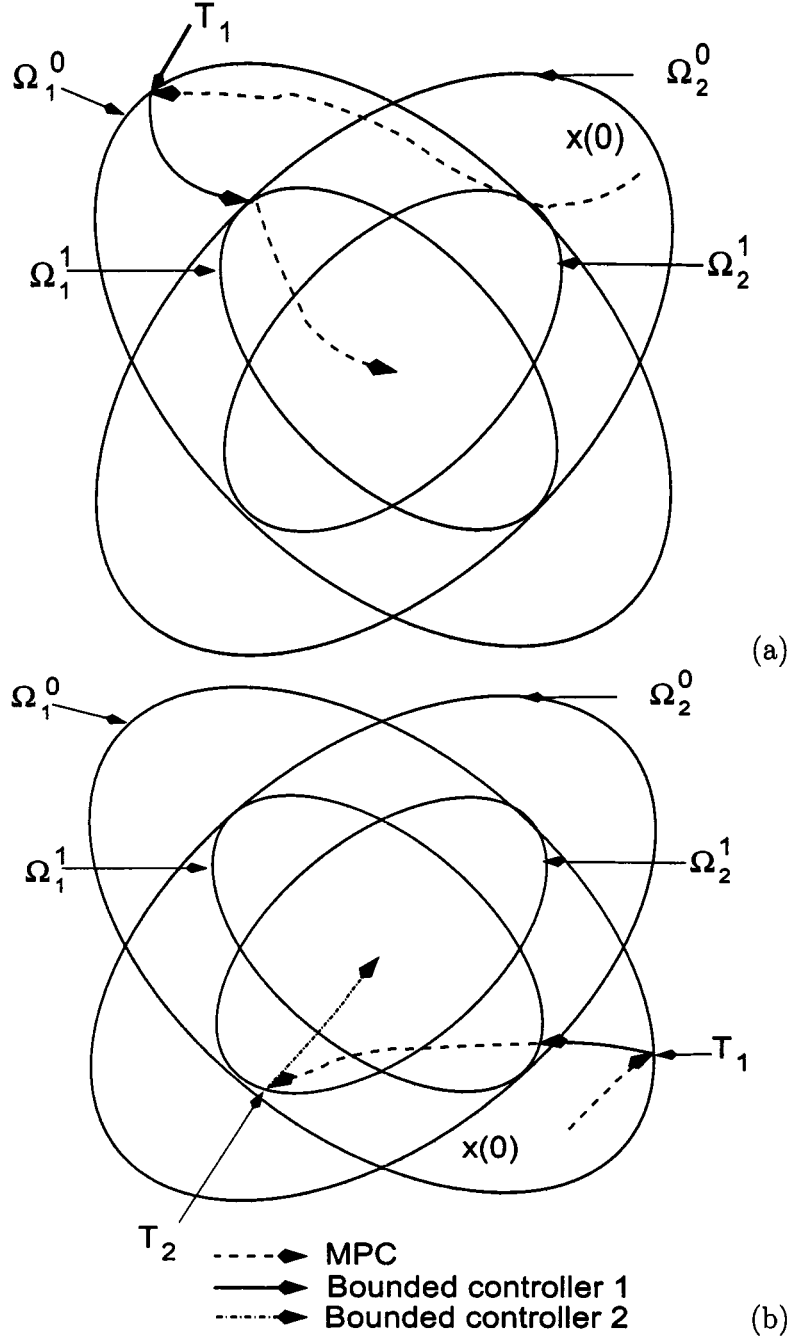


Figure 5.1: Examples of the evolution of closed-loop state trajectory under the switching scheme of Theorem 5.2.

of Eq.5.1. In all simulation runs, $\theta_1(t) = \theta_0 \sin(3t)$, where $\theta_0 = 0.08T_{A0s}$ and $\theta_2(t) = 0.5(-\Delta H_{nom})$, and the manipulated input constraints were $|u_1| \leq 1.0 \text{ Kmol/m}^3$ and $|u_2| \leq 92 \text{ KJ/s}$.

Table 5.1: Process parameters and steady-state values for the reactor of Eq.5.12.

$V_R = 0.1 \text{ m}^3$	$R = 8.314 \text{ J/mol} \cdot \text{K}$
$C_{A0s} = 1.0 \text{ Kmol/m}^3$	$T_{A0s} = 310.0 \text{ K}$
$\Delta H_n = -4.78 \times 10^4 \text{ J/mol}$	$k_0 = 72 \times 10^9 \text{ min}^{-1}$
$E = 8.314 \times 10^4 \text{ J/mol}$	$c_p = 0.239 \text{ J/g} \cdot \text{K}$
$\rho = 1000.0 \text{ Kg/m}^3$	$F = 0.1 \text{ m}^3/\text{min}$
$T_{Rs} = 395.3 \text{ K}$	$C_{As} = 0.57 \text{ Kmol/m}^3$

A quadratic Lyapunov function was used to design the bounded robust control design of Eq.5.2, using $\chi = 1.01$, $\phi = 0.0001$, and $\rho = 0.01$. The term β^2 in the denominator of the control law of Eq.5.2 was replaced by the number $\nu = 0.001$ close to the origin to alleviate chattering of the control action (note that for this example, the denominator term $\beta^2 = 0$ if and only if $x = 0$). The set of nonlinear ODEs was integrated using the MATLAB solver, ODE45, and the optimization problem in MPC was solved using the MATLAB nonlinear constrained optimization solver, *fmincon*.

The first set of simulation runs demonstrate the use of the robust hybrid predictive control strategy of Theorem 5.1, with the closed-loop system initialized from a point within the stability region $\Omega(u_{max}, \theta_b)$. In the first scenario, a nominal nonlinear MPC formulation, without stability constraints, is used as part of the robust predictive control structure (setting $\Omega_{MPC} = \mathbb{R}^n$, $F(x(t+T)) = 0$, $T = 0.02$ minutes in Eqs.5.4–5.5) and with the design parameter $T_{design} = 10$ minutes. Shortly after the initial implementation of MPC, the supervisor detects, at $t = 0.6$ seconds, that the closed-loop states are close to the boundary of $\Omega(u_{max}, \theta_b)$ and therefore switches to the bounded robust controller to stabilize the closed-loop system (solid lines in Figures 5.2, 5.4). Note that the stability region information is completely contained in the value of the level set obtained at the time of the computation of the stability region

(c^{max}) and the supervisor reaches this inference by simply evaluating the Lyapunov function, and comparing it to c^{max} . In the next scenario, a stabilizing formulation of MPC is used (requiring the states to go to a small invariant set at the end of the horizon), with a horizon length of $T = 0.02$ minutes and a $T_{design} = 20$ minutes. For the initial condition of the trajectory shown by the dashed lines in Figures 5.2, 5.4, the MPC yields a feasible solution and drives the states close to the origin. For the initial condition depicted by the dotted lines in Figures 5.2, 5.4, however, the MPC does not yield a feasible solution, and therefore the supervisor initially implements the bounded robust controller, switching to the MPC at $t = 0.465$ minutes, when the MPC becomes feasible, and leads to closed-loop stability.

Finally, we demonstrate the implementation of the relaxed switching scheme of Theorem 5.2 using two Lyapunov functions. We use a nominal nonlinear MPC formulation, without stability constraints (setting $\Omega_{MPC} = \mathbb{R}^n$, $F(x(t+T)) = 0$, $T = 0.02$ minutes in Eqs.5.4–5.5) and with the design parameter $T_{design} = 3$ minutes. Starting from an initial condition within Ω_2 under MPC, the switching logic allows MPC to be implemented in the closed-loop even though the states escape out of Ω_2 , since they are still within Ω_1 (see solid line in Figure 5.3 and dashed lines in Figure 5.4). Note also that in this case the nominal MPC does not enforce the desired degree of uncertainty attenuation (note the oscillations). After the time T_{design} has elapsed, the supervisor implements the bounded controller (associated with Ω_2) in the closed-loop system to achieve practical stabilization.

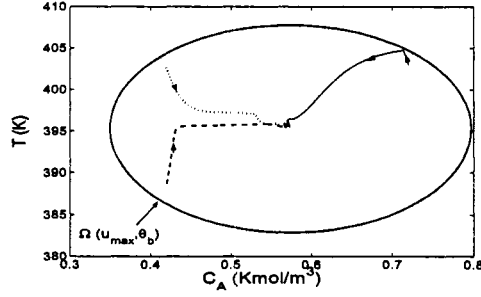


Figure 5.2: Closed-loop state trajectory: implementation of the robust hybrid predictive controller of Theorem 5.1 using a nominal MPC formulation without stability constraints (solid line) and an MPC formulation with stability constraints, from two different initial conditions (dashed and dotted lines).

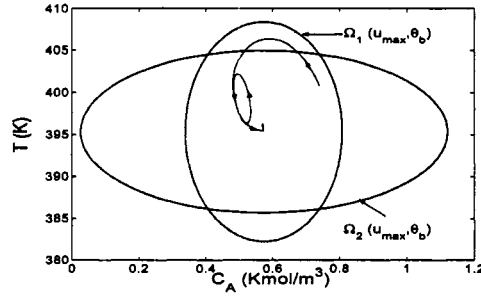


Figure 5.3: Closed-loop state trajectory: implementation of the robust hybrid predictive controller of Theorem 5.2, with $l = 2$, $N = 1$, using a nominal MPC formulation without stability constraints (solid line).

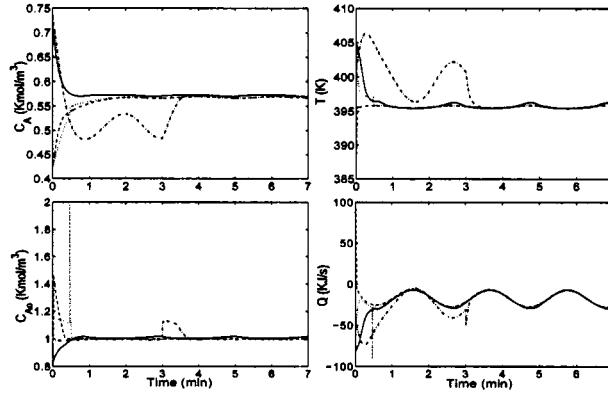


Figure 5.4: Closed-loop state (top) and input (bottom) profiles: implementation of the robust hybrid predictive controller of Theorem 5.1 using a nominal MPC formulation without stability constraints (solid lines), an MPC formulation with stability constraints, from two different initial conditions (dashed and dotted lines) and the robust hybrid predictive controller of Theorem 5.2, with $l = 2$, $N = 1$, using a nominal MPC formulation without stability constraints (dash-dotted lines).

Chapter 6

Stabilization of Nonlinear Systems with State and Control Constraints Using Lyapunov-Based Predictive Control

6.1 Introduction

Control systems are often subject to constraints on their manipulated inputs and state variables. Input constraints arise as a manifestation of the physical limitations inherent in the capacity of control actuators (e.g., bounds on the magnitude of valve opening), and are enforced at all times (hard constraints). State constraints, on the other hand, arise either due to the necessity to keep the state variables within acceptable ranges to avoid, for example, runaway reactions (in which case they need to be enforced at all times, and treated as hard constraints) or due to the desire to maintain them within desirable bounds dictated by performance considerations (in which case they may be relaxed, and treated as soft constraints). Constraints automatically impose limitations on our ability to steer the dynamics of the closed-loop system at will, and can cause

severe deterioration in the nominal closed-loop performance and may even lead to closed-loop instability if not explicitly taken into account at the stage of controller design.

Currently, model predictive control (MPC), also known as receding horizon control (RHC), is one of the few control methods for handling state and input constraints within an optimal control setting and has been the subject of numerous research studies that have investigated the stability properties of MPC (e.g., see [3, 141] for extensive surveys of various MPC formulations). In these MPC formulations the stability guarantees are typically based on an assumption of initial feasibility of the optimization problem; and the set of initial conditions starting from where a given MPC formulation is guaranteed to be feasible is not explicitly characterized. Attention has also been focused on the problem of state constraints satisfaction [83, 241, 182, 161, 244, 19, 202, 204], and has typically been analyzed within the soft constraints framework, i.e., with the understanding that state constraints may be relaxed. In the minimum time approach, the state constraints are relaxed up-to some time, and set as hard constraints thereafter. In other approaches, they are typically relaxed for all times, and only incorporated in the objective function as appropriate penalties on state constraint violation ('softening' of state constraints). In either approach, the problem of providing explicitly the set of initial conditions starting from where stabilization can be achieved and state and input constraints are guaranteed to be feasible has not been addressed.

The desire to implement control approaches that allow for an explicit characterization of their stability properties has motivated significant work on the design of stabilizing control laws, using Lyapunov techniques, that provide explicitly-defined regions of attraction for the closed-loop system; the reader may refer to [114] for a survey of results in this area. In [66, 68], a class of Lyapunov-based bounded robust nonlinear controllers (inspired by the results on bounded control originally presented in [127]) that enforce robust stability from an explicitly characterized set of initial conditions, was de-

veloped. Despite their well-characterized stability and constraint-handling properties, those Lyapunov-based controllers are not guaranteed to be optimal with respect to an arbitrary performance criterion, and do not allow for incorporation of performance considerations in the design.

In this chapter, we propose a Lyapunov-based model predictive control design for the stabilization of nonlinear systems with both state and input constraints. The design of the Lyapunov-based MPC uses a bounded controller, with its associated region of stability, as an auxiliary controller that is used to characterize the stability properties of the Lyapunov-based MPC. The proposed Lyapunov-based MPC is shown to possess an explicitly characterized set of initial conditions, starting from where it is guaranteed to be feasible, and hence stabilizing, while enforcing the state and input constraints at all times. For the case when the state constraints are soft, we propose a switched predictive control strategy that reduces the time for which state constraints are violated, driving the states into the state and input constraints feasibility region of the Lyapunov-based predictive controller. We demonstrate the application of the Lyapunov-based predictive controller designs through a chemical process example.

The rest of the chapter is organized as follows: in Section 6.2, we describe the class of systems considered, and briefly review a bounded controller design. In Section 6.3 we present a Lyapunov-based predictive controller that guarantees stabilization from an explicitly characterized set of initial conditions, while enforcing the state and input constraints at all times. In Section 6.4, we present another predictive control formulation, and a switching strategy that reduces the time for which the state constraints are violated when state constraints can be treated as soft constraints. Finally, in Section 6.5, we present simulation results to demonstrate the application of our results.

6.2 Preliminaries

In this chapter, we consider the problem of stabilization of continuous-time nonlinear systems with state and input constraints, with the following state-space description:

$$\dot{x}(t) = f(x(t)) + G(x(t))u(t) \quad (6.1)$$

$$u \in U \quad (6.2)$$

$$x \in X \quad (6.3)$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables, $u = [u^1 \cdots u^m]' \in \mathbb{R}^m$ denotes the vector of manipulated inputs, $U \subseteq \mathbb{R}^m$, $X \subseteq \mathbb{R}^n$ denote the constraints on the manipulated inputs and the state variables, respectively, $f(\cdot)$ is a sufficiently smooth $n \times 1$ nonlinear vector function, and $G(\cdot)$ is a sufficiently smooth $n \times m$ nonlinear matrix function. Without loss of generality, it is assumed that the origin is the equilibrium point of the unforced system (i.e., $f(0) = 0$). Throughout the chapter, the notation $\|\cdot\|$ will be used to denote the standard Euclidean norm of a vector, while the notation $\|\cdot\|_Q$ refers to the weighted norm, defined by $\|x\|_Q^2 = x'Qx$ for all $x \in \mathbb{R}^n$, where Q is a positive-definite symmetric matrix and x' denotes the transpose of x . The notation $L_f V$ denotes the standard Lie derivative of a scalar function $V(\cdot)$ with respect to the vector function $f(\cdot)$. In order to provide the necessary background for our results in Sections 6.3 and 6.4, we will briefly review in the remainder of this section the design procedure for, and the stability properties of, a bounded control design, which will be used to characterize the feasibility region of the Lyapunov-based MPC formulation in Section 6.3. Throughout the manuscript, we assume that for any $u \in U$ the solution of the system of Eq.6.1 exists and is continuous for all t , and we focus on the state feedback problem where measurements of $x(t)$ are assumed to be available for all t .

6.2.1 Bounded Lyapunov-based control

Consider the system of Eq.6.1 for which a control Lyapunov function, V , exists. Using the results in [127] (see also [66]), the following bounded control law can be constructed:

$$u(x) = \begin{cases} -k(x)(L_G V)'(x) & , \quad \|(L_G V)'(x)\| \neq 0 \\ 0 & , \quad \|(L_G V)'(x)\| = 0 \end{cases} := b(x) \quad (6.4)$$

where

$$k(x) = \frac{L_f^* V(x) + \sqrt{\left(L_f^* V(x)\right)^2 + (u^{max}\|(L_G V)'(x)\|)^4}}{\|(L_G V)'(x)\|^2 \left[1 + \sqrt{1 + (u^{max}\|(L_G V)'(x)\|)^2}\right]} \quad (6.5)$$

$L_G V(x) = [L_{g^1} V \cdots L_{g^m} V]$ is a row vector, where g^i is the i -th column of G , $L_f^* V = L_f V + \rho V$ and $\rho > 0$, and u^{max} is a real positive number such that $\|u\| \leq u^{max}$ implies $u \in U$. For the above controller, one can show, using a standard Lyapunov argument, that whenever the closed-loop state, x , evolves within the region described by the set:

$$\Phi_{x,u} = \{x \in X : L_f^* V(x) \leq u^{max}\|(L_G V)'(x)\|\} \quad (6.6)$$

then the controller satisfies the state and input constraints, and the time-derivative of the Lyapunov function is negative-definite. Therefore, starting from any initial state in the set $\Phi_{x,u}$, asymptotic stability of the constrained closed-loop system can be guaranteed, provided that the state trajectory of the closed-loop system remains within the state-space region described by the set $\Phi_{x,u}$. To ensure this, we need to construct an invariant subset of $\Phi_{x,u}$ (preferably the largest). One way to construct such a subset is using the level sets of V , i.e.,

$$\Omega_{x,u} = \{x \in \mathbb{R}^n : V(x) \leq c_{x,u}^{max}\} \quad (6.7)$$

where $c_{x,u}^{max} > 0$ is the largest number for which $\Omega_{x,u} \subseteq \Phi_{x,u}$. $\Omega_{x,u}$ then provides an estimate of the stability region starting from where the origin of the constrained closed-loop system under the control law of Eqs.6.4-6.5 is guaranteed to be asymptotically stable and the state and input constraints are satisfied for all times.

The bounded controller of Eqs.6.4-6.5 possesses a robustness property (with respect to measurement errors) that preserves closed-loop stability when the control action is

implemented in a discrete (sample and hold) fashion with a sufficiently small hold time (Δ). Specifically, the control law ensures that, for all initial conditions in $\Omega_{x,u}$, the closed-loop state remains in $\Omega_{x,u}$ and eventually converges to some neighborhood of the origin (we will refer to this neighborhood as Ω^b) whose size depends on Δ . This robustness property was characterized in [146] for the problem of stabilization under input constraints, and carries over to the case of input and state constraints. This property will be exploited in the Lyapunov-based predictive controller design of Section 6.3 and is stated in Proposition 6.1 below (the proof of the proposition is similar to that of Proposition 6.1 in [146], and is omitted for brevity). For further results on the analysis and control of sampled-data nonlinear systems, the reader may refer to [89, 158, 107, 240].

Proposition 6.1: *Consider the constrained system of Eq.6.1, under the bounded control law of Eqs.6.4–6.5 with $\rho > 0$ and let $\Omega_{x,u}$ be the stability region estimate under continuous implementation of the bounded controller. Let $u(t) = u(j\Delta)$ for all $j\Delta \leq t < (j+1)\Delta$ and $u(j\Delta) = b(x(j\Delta))$, $j = 0, \dots, \infty$. Then, given any positive real number d , there exist positive real numbers Δ^* , δ' and ϵ^* such that if $\Delta \in (0, \Delta^*]$ and $x(0) := x_0 \in \Omega_{x,u}$, then $x(t) \in \Omega_{x,u} \subseteq X$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$. Also, if $V(x_0) \leq \delta'$ then $V(x(\tau)) \leq \delta' \forall \tau \in [0, \Delta)$ and if $\delta' < V(x_0) \leq c_{x,u}^{max}$, then $\dot{V}(x(\tau)) \leq -\epsilon^* \forall \tau \in [0, \Delta)$.*

Remark 6.1: Control Lyapunov function (CLF)-based stabilization of nonlinear systems has been studied extensively in the nonlinear control literature (e.g., see [9, 127, 207]). The construction of constrained CLFs (i.e., CLFs that take the constraints into account) remains a difficult problem (especially for nonlinear systems) that is the subject of ongoing research. For several classes of nonlinear systems that arise commonly in the modeling of engineering systems, systematic and computationally feasible methods are available for constructing unconstrained CLFs (CLFs for the unconstrained system) by exploiting the system structure. Examples include the use of quadratic

functions for feedback linearizable systems and the use of back-stepping techniques to construct CLFs for systems in strict feedback form. In this chapter, the bounded controllers in Eqs.6.4-6.5 are designed using unconstrained CLFs, which are also used to explicitly characterize the associated regions of stability via Eqs.6.7-6.6. While the resulting estimates do not necessarily capture the entire domain of attraction (this remains an open problem even for linear systems), we will use them throughout the chapter for a concrete illustration of the basic ideas of the results. It is possible to obtain substantially improved estimates by using, for example, a combination of several CLFs (see, for example, [74, 147]).

6.3 Lyapunov-based Model predictive control

Consider model predictive control of the system of Eq.6.1 with hard state and input constraints. We present here a Lyapunov-based MPC formulation (see Remark 6.3 for a discussion on this formulation and its relationship to other Lyapunov-based formulations) that guarantees feasibility of the optimization problem subject to hard constraints on the state and input, and hence constrained stabilization of the closed-loop system from an explicitly characterized set of initial conditions. For this MPC design, the control action at state x and time t is obtained by solving, on-line, a finite horizon optimal control problem of the form:

$$P(x, t) : \min\{J(x, t, u(\cdot)) | u(\cdot) \in S, x \in X\} \quad (6.8)$$

$$s.t. \quad \dot{x} = f(x) + G(x)u \quad (6.9)$$

$$\dot{V}(x(\tau)) \leq -\epsilon^* \quad \forall \tau \in [t, t + \Delta) \text{ if } V(x(t)) > \delta' \quad (6.10)$$

$$V(x(\tau)) \leq \delta' \quad \forall \tau \in [t, t + \Delta) \text{ if } V(x(t)) \leq \delta' \quad (6.11)$$

where $S = S(t, T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ , mapping $[t, t + T]$ into U and T is the specified horizon. Eq.6.9 is the nonlinear model describing the time evolution of the state x , V is the Lyapunov function used in the bounded controller design and δ' , ϵ^* are defined in

Proposition 6.1. A control $u(\cdot)$ in S is characterized by the sequence $\{u[j]\}$ where $u[j] := u(j\Delta)$ and satisfies $u(t) = u[j]$ for all $t \in [j\Delta, (j+1)\Delta)$. The performance index is given by

$$J(x, t, u(\cdot)) = \int_t^{t+T} [\|x^u(s; x, t)\|_Q^2 + \|u(s)\|_R^2] ds \quad (6.12)$$

where Q is a positive semi-definite symmetric matrix and R is a strictly positive definite symmetric matrix. $x^u(s; x, t)$ denotes the solution of Eq.6.1, due to control u , with initial state x at time t . The minimizing control $u^0(\cdot) \in S$ is then applied to the plant over the interval $[j\Delta, (j+1)\Delta)$ and the procedure is repeated indefinitely. This defines an implicit model predictive control law:

$$M(x) := \operatorname{argmin}(J(x, t, u(\cdot))) := u_1 \quad (6.13)$$

Closed-loop stability and state and input constraint feasibility properties of the closed-loop system under the Lyapunov-based predictive controller are inherited from the bounded controller under discrete implementation and are formalized in Proposition 6.2 below.

Proposition 6.2: *Consider the constrained system of Eq.6.1 under the MPC law of Eqs.6.8–6.13 with $\Delta \leq \Delta^*$ where Δ^* was defined in Proposition 6.1. Then, given any $x_0 \in \Omega_{x,u}$, where $\Omega_{x,u}$ was defined in Eq.6.7, the optimization problem of Eq.6.8–6.13 is feasible for all times, $x(t) \in \Omega_{x,u} \subseteq X$ for all $t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$.*

Proof of Proposition 6.2: The proof of this proposition is divided in three parts. In the first part we show that for all $x_0 \in \Omega_{x,u}$, the predictive control design of Eqs.6.8–6.13 is feasible. We then show that $\Omega_{x,u}$ is invariant under the predictive control algorithm of Eqs.6.8–6.13, and that the state and input constraints are satisfied for all times. Finally, we prove practical stability for the closed-loop system.

Part 1: Consider some $x_0 \in \Omega_{x,u}$ under the predictive controller of Eqs.6.8–6.13, with a prediction horizon $T = N\Delta$, where Δ is the hold time and $1 \leq N < \infty$ is the number of the prediction steps. Note that since $V(\cdot)$ is a continuous function of the state, one

can find a finite, positive real number, δ' , such that $V(x) \leq \delta'$ implies $\|x\| \leq d$. The initial condition can either be such that $V(x_0) \leq \delta'$ or $\delta' < V(x_0) \leq c_{x,u}^{max}$.

Case 1: If $\delta' < V(x_0) \leq c_{x,u}^{max}$, the control input trajectory under the bounded controller of Eqs.6.4–6.5 provides a feasible solution to the constraint of Eq.6.10 (see Proposition 6.1). A feasible initial guess for the optimization problem of Eqs.6.8-6.13 therefore exists, and is given by $u(j\Delta) = b(j\Delta)$, $j = 1, \dots, N$. Note that if $u(x(t)) = b(x(t))$ for $t \in [0, \Delta]$, and $\Delta \in (0, \Delta^*]$, then $\dot{V} \leq -\epsilon^*$ and $b(x(t)) \in U$ (since $b(\cdot)$ is computed using the bounded controller of Eqs.6.4–6.5). Also, under discrete implementation of the bounded controller of Eqs.6.4–6.5, for any $x_0 \in \Omega_{x,u}$, $x(t) \in \Omega_{x,u} \forall t \geq 0$, therefore the constraint $x(t) \in X$ is also satisfied by this control trajectory (note that $\Omega_{x,u} \subseteq \Phi_{x,u} \subseteq X$).

Case 2: If $V(x_0) \leq \delta'$, once again we infer from Proposition 6.1 that the control input trajectory provided by the bounded controller of Eqs.6.4–6.5 provides a feasible initial guess, given by $u(j\Delta) = b(x(j\Delta))$, $j = 1, \dots, N$ (recall from Proposition 6.1 that, under the bounded controller of Eqs.6.4–6.5, if $V(x_0) \leq \delta'$ then $V(x(t)) \leq \delta' \forall t \geq 0$). This shows that for all $x_0 \in \Omega_{x,u}$, the Lyapunov based predictive controller of Eqs.6.8-6.13 is feasible for all times.

Part 2: As shown in Part 1, for any $\delta' < V(x_0) \leq c_{x,u}^{max}$, the constraint of Eq.6.10 in the optimization problem is feasible. Upon implementation, therefore, the value of the Lyapunov function decreases, and since $\Omega_{x,u}$ is a level set of V , the closed-loop state trajectory cannot escape out of $\Omega_{x,u}$. On the other hand, if $V(x_0) \leq \delta'$, feasibility of the constraint of Eq.6.11 guarantees that the closed-loop state trajectory evolves such that $V(x(t)) \leq \delta' \forall t \geq 0$. In both cases, $\Omega_{x,u}$ continues to be an invariant region under the Lyapunov based predictive controller of Eqs.6.8-6.13. Also, since $\Omega_{x,u} \subseteq \Phi_{x,u} \subseteq X$, we have that $x(t) \in X$ for all $t \geq 0$.

Part 3: Finally, consider an initial condition, x_0 , such that $\delta' < V(x_0) \leq c_{x,u}^{max}$. Since the optimization problem continues to be feasible for all $t \geq 0$, we have that $\dot{V}(x(t)) < 0$

for all $\delta' < V(x(t)) \leq c_{x,u}^{max}$. All trajectories originating in $\Omega_{x,u}$, therefore converge to the set defined by $\Omega^t := \{x \in \mathbb{R}^n : V(x) \leq \delta'\}$. For $V(x_0) \leq \delta'$, the feasibility of the optimization problem of Eqs.6.8-6.13 implies $V(x(t)) \leq \delta' \forall t \geq 0$. Therefore, for all $x_0 \in \Omega_{x,u}$, $\limsup_{t \rightarrow \infty} V(x(t)) \leq \delta'$. For such a choice of δ' , we therefore have that $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$. This completes the proof of Proposition 6.2.

Remark 6.2: The set $\Phi_{x,u}$ defined by Eq.6.6 can be understood as the intersection of the sets $\Phi_u = \{x \in \mathbb{R}^n : L_f^* V(x) \leq u^{max} \|(L_G V)'(x)\|\}$, which defines the set where $\dot{V} < 0$ and $\|u\| \leq u^{max}$, and the set X describing the state constraints. The set defined by

$$\Omega_u = \{x \in \mathbb{R}^n : V(x) \leq c_u^{max}\} \quad (6.14)$$

where $c_u^{max} > 0$ is the largest number for which $\Omega_u \subseteq \Phi_u$, therefore, is the region of guaranteed feasibility for the Lyapunov-based predictive controller in the absence of state constraints, see [146]. It can be argued that the set $\Phi_{x,u}$ is a sufficiently non-conservative estimate of the set of initial conditions within which the level set, $\Omega_{x,u}$, can be constructed (see Fig.6.1). To this end, consider the following cases: if $X \subset \Phi_u$ (as in Fig.6.1), the closed-loop system cannot be initialized outside of X , because in that case the state constraints are violated at the outset (see Section 6.4 for how this can be handled in the case that the state constraints can be treated as soft constraints). On the other hand, if $\Phi_u \subset X$, the closed-loop system state cannot be initialized from outside of Φ_u , since outside of this set, negative definiteness of $\dot{V}$ is not guaranteed under input constraints (this points to the fact that there may be initial conditions that satisfy the state constraints, but starting from where it may not be possible to stabilize the system due to the presence of input constraints). Also, it is possible to generate a larger estimate of Φ_u , by using, for example, a family of Lyapunov functions [74], to cover larger portions of the set X . Note also that, in general, merely initializing the closed-loop system inside X is not sufficient to guarantee subsequent satisfaction of the state constraints because the closed-loop state trajectory, even if it eventually

stabilizes, may do so in a way that it goes out of X , and violates the state constraints before stabilizing (this motivates the construction of the invariant set $\Omega_{x,u}$).

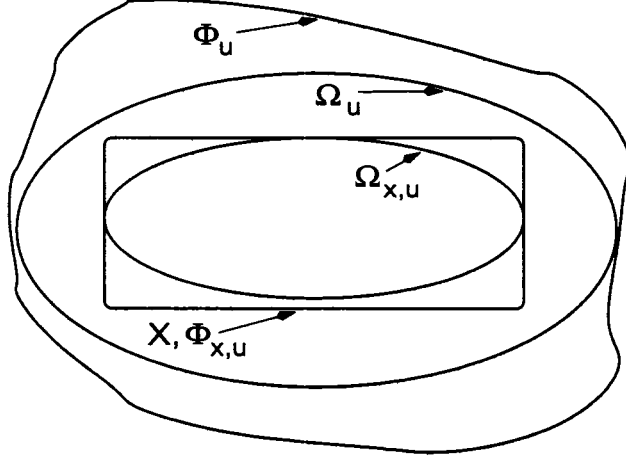


Figure 6.1: A schematic representing the input (Φ_u, Ω_u) and state and input $(\Phi_{x,u}, \Omega_{x,u})$ constrained stability regions, together with the set (X) describing the state constraints.

Remark 6.3: Note that the predictive controller formulation of Eqs.6.8–6.13 requires that the value of the Lyapunov function decrease during the first step only. Practical stability of the closed-loop system is achieved since, due to the receding nature of controller implementation, only the first move of the set of calculated moves is implemented and the problem is re-solved at the next time step. If the optimization problem is initially feasible, and continues to be feasible, then every control move that is implemented, enforces a decay in the value of the Lyapunov function, leading to stability (see Remark 6.4 for how initial and subsequent feasibility is guaranteed for an explicitly characterized set of initial conditions). Lyapunov-based predictive control approaches (see, for example, [115, 170]) typically incorporate a similar Lyapunov function decay constraint, albeit requiring the constraint of Eq.6.10 to hold at the *end* of the prediction horizon as opposed to only the first time step. An input trajectory that only requires the value of the Lyapunov function value to decrease at the end of the horizon may involve the state trajectory going out of the level set (and therefore, possibly out of

the state constraint satisfaction region, violating the state constraints), and motivates using a constraint that requires the Lyapunov function to decrease during the first time step (this also facilitates the explicit characterization of the feasibility region).

Remark 6.4: For $0 < \Delta \leq \Delta^*$, the constraint of Eq.6.10, is guaranteed to be satisfied (the control action computed by the bounded controller design provides a feasible initial guess to the optimization problem). Note that this is so because the constraint requires the Lyapunov function value to decay, not at the *end* of the prediction horizon (as is customarily done in Lyapunov-based MPC approaches), but only at the end of the first time step. Furthermore, since the state is initialized in $\Omega_{x,u}$, which is a level set of V , the closed-loop system evolves so as to stay within $\Omega_{x,u}$, thereby guaranteeing feasibility at future times. Since the level set $\Omega_{x,u}$ is completely contained in the set defining the state constraints, and the state trajectory under the predictive controller continues to evolve within this set, the state constraints are satisfied at all times.

Remark 6.5: For linear systems, the work in [182] uses the minimum time approach, where the smallest time, T_{min} , beyond which the state constraints can be satisfied on an infinite horizon is identified, and the state constraints are relaxed up-to that time. The need to satisfy the constraints after T_{min} may, however, result in large violations of the state constraints during the initial period. Note also that even if the state constraints are relaxed up-to T_{min} , the initial conditions starting from where the closed-loop system can be stabilized are not explicitly characterized. Furthermore, while possible for linear systems, the computation of the time beyond which state constraints are satisfied, is more difficult for nonlinear systems. Even if computable, the state constraints are not enforced at all times, and hence not strictly treated as hard constraints. In contrast, the Lyapunov-based MPC of Proposition 6.2 guarantees feasibility and stabilization from an explicitly characterized set of initial conditions while enforcing the state and input constraints at all times.

Remark 6.6: Note that the constraints of Eqs.6.10-6.11 incorporate stability consid-

erations in the control design, which override the performance considerations specified by the objective function of Eq.6.12 (which means that the implemented control action may not be the one dictated by the performance objective of Eq.6.12 alone). In contrast to analytic bounded control methods, however, the predictive control design allows for specification of performance objective in the control design that ultimately determines the choice of control action out of the allowable (feasible) set of control moves determined by the constraints on the manipulated input and the states and the stability requirements.

Remark 6.7: Together with the computational difficulties of solving a nonlinear optimization problem at each time step, one of the key challenges that impact on the practical implementation of nonlinear MPC (NMPC) is the inherent difficulty of characterizing, *a priori*, the set of initial conditions starting from where a given NMPC controller is guaranteed to stabilize the closed-loop system, or for a given set of initial conditions, to identify the value of the prediction horizon for which the optimization problem will be feasible subject to state and input constraints. The Lyapunov-based predictive controller formulation guarantees initial and subsequent feasibility of the optimization problem irrespective of the choice of the prediction horizon and also provides, at the same time, an explicit characterization of a set of initial conditions starting from where stability is guaranteed. In addition, the optimization problem in the predictive controller is initialized with a feasible initial guess, which substantially reduces the computation burden. Note also that any other Lyapunov-based analytic control design can be used as the auxiliary controller, and the choice is not limited to the bounded controller used in this chapter. The Lyapunov-based analytic control design should, however, provide an explicit characterization of the state and input constrained stability region, and be robust with respect to discrete implementation.

6.4 Handling soft state constraints via controller switching

Consider now the nonlinear system of Eq.6.1 where the state constraints represent desired bounds on the values of the state variables. In this case, the state constraints can be treated as soft constraints, allowing their violation for some period of time. It is important, nevertheless, to implement control action that reduces the time for which constraints are violated. We propose in this section a control design that uses two predictive formulations and switches between them. First, we design a predictive controller that, while respecting input constraints, drives the state trajectory into the feasible region of the predictive controller formulation of Eqs.6.8-6.13, in a way that reduces the time for which the state constraints are violated, and then implement the predictive controller of Eqs.6.8-6.13 to achieve stabilization together with state and input constraint satisfaction for the rest of the time. To this end, we first cast the system of Eq.6.1 as a switched system of the form:

$$\begin{aligned} \dot{x} &= f(x) + G(x)u_{i(t)} \\ u_i &\in U \\ i &\in \{1, 2\} \end{aligned} \tag{6.15}$$

where $i : [0, \infty) \rightarrow \{1, 2\}$ is the switching signal. The index, $i(t)$, represents a discrete state that indexes the control input, u , with the understanding that $i(t) = 1$ if and only if $u_i(x(t)) = u_1$ (i.e., the Lyapunov-based MPC formulation of Eqs.6.8-6.13 is used) and $i(t) = 2$ if and only if $u_i(x(t)) = u_2$ (i.e., an MPC formulation designed to reduce the time of state constraint violation, is used). Theorem 6.1 below presents both the control law, u_2 , and the switching law.

Theorem 6.1: *Consider the switched nonlinear system of Eq.6.15, for which there exists a control Lyapunov function V , and for a given pair of positive real numbers (d, ρ) , Δ is chosen such that $\Delta \leq \Delta^*$, where Δ^* was defined in Proposition 6.1. Given any initial condition $x_0 \in \Omega_u$, where Ω_u was defined in Eq.6.14, let T_b be the time it takes for the bounded controller of Eqs.6.4–6.5, under discrete implementation with a discretization step Δ , to achieve $x(T_b) \in \Omega_{x,u}$. Consider the following optimization problem:*

$$u = \operatorname{argmin}(J) := u_2 \quad (6.16)$$

$$J = qV(x(t + \Delta)) + \int_t^{t+T} [\|u(s)\|_R^2] ds \quad (6.17)$$

where q is a positive real number, R is a positive definite matrix, T is the prediction horizon given by $T = T_b - t$, subject to the following constraints:

$$\dot{x} = f(x) + G(x)u \quad (6.18)$$

$$u \in U \quad (6.19)$$

$$V(x(t + k\Delta)) \leq V(x(t + (k - 1)\Delta)), \quad k = 1, \dots, T/\Delta \quad (6.20)$$

$$x(t + T) \in \Omega_{x,u} \quad (6.21)$$

Let T_{switch} be the earliest time such that $x(T_{switch}) \in \Omega_{x,u}$, where $\Omega_{x,u}$ was defined in Eq.6.7, under the controller of Eqs.6.16-6.21. Then, the following switching law:

$$i(t) = \begin{cases} 2 & , \quad 0 \leq t \leq T_{switch} \\ 1 & , \quad t > T_{switch} \end{cases} \quad (6.22)$$

ensures, for the closed-loop system, that $x(T_b) \in \Omega_{x,u}$, $T_{switch} \leq T_b$, $x(t) \in \Omega_{x,u} \subseteq X \forall t > T_{switch}$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$.

Proof of Theorem 6.1: The proof of the theorem proceeds as follows: we first show that the optimization problem of Eqs.6.16–6.21 is feasible for all $0 \leq t \leq T_{switch}$, $x(T_{switch}) \in \Omega_{x,u}$ and that $T_{switch} \leq T_b$. Then, we use the result of Proposition 6.2 to show that for $t > T_{switch}$, the controller of Eqs.6.8-6.13 ensures that $x(t) \in \Omega_{x,u} \subseteq X \forall t > T_{switch}$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$.

Case 1: Consider $x_0 \in \Omega_u \setminus \Omega_{x,u}$. Using the result of Proposition 6.1 in [146], we have that under discrete implementation of the bounded controller of Eqs.6.4–6.5 with $\Delta \leq \Delta^*$, the state trajectory, starting from x_0 , evolves such that $x(t) \in \Omega_u$ and $V(x(t + \Delta)) \leq V(x(t))$ for all $t \geq 0$. From the definition of T_b , we have that $x(T_b) \in \Omega_{x,u}$. The optimization problem of Eqs.6.16–6.21 is guaranteed to be initially feasible, since a feasible initial guess can always be obtained using the control input trajectory under the bounded controller and is given by:

$$u(k\Delta) = b(x(k\Delta)), k = 1, \dots, T/\Delta$$

Subsequently, the tail of the solution at the first time step:

$$u(k\Delta), k = 2, \dots, T/\Delta$$

is a feasible initial guess for the constraints in the optimization problem at the next time step (at the next time step, the horizon reduces from $T = T_b$ to $T = T_b - \Delta$). Note that under the implementation of the solution of the control trajectory at the first time step, we get

$$V(x(t + k\Delta)) \leq V(x(t + (k - 1)\Delta)), k = 1, \dots, T/\Delta$$

Under the implementation of the tail, therefore, we have that

$$V(x(t + k\Delta)) \leq V(x(t + (k - 1)\Delta)), k = 2, \dots, T/\Delta$$

and also $x(T_b) \in \Omega_{x,u}$, which is the constraint that the optimization problem needs to enforce at the next time step. The optimization problem of Eqs.6.16–6.21, therefore, is guaranteed to be initially and successively feasible, and hence $x(T_b) \in \Omega_{x,u}$.

By definition of T_{switch} , if the state trajectory enters $\Omega_{x,u}$ before T_b , then T_{switch} is set to that value, hence $x(T_{switch}) \in \Omega_{x,u}$ where $T_{switch} \leq T_b$. From Proposition 6.2, we get that for all $x(T_{switch}) \in \Omega_{x,u}$, $x(t) \in \Omega_{x,u} \subseteq X \forall t \geq T_{switch}$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$.

Case 2: For any initial condition $x_0 \in \Omega_{x,u} \subseteq \Omega_u$ we have that $T_b = T_{switch} = 0$ (since $x_0 \in \Omega_{x,u}$), and the switching law of Eq.6.22 dictates that the controller of Eqs.6.8–6.13 is implemented for all times. Since $x_0 \in \Omega_{x,u}$, from Proposition 6.2, we get that $x(t) \in \Omega_{x,u} \subseteq X \forall t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$. This completes the proof of Theorem 6.1.

Remark 6.8: The implementation of the predictive controller of Theorem 6.1 is described algorithmically below:

1. Given the system model of Eq.6.1, the constraints on the input and states, and a control Lyapunov function, V , design the bounded controller of Eqs.6.4–6.5

and compute the stability region estimate under input constraints, Ω_u , and the stability region under state and input constraints, $\Omega_{x,u}$.

2. Given the size of the ball, d , that the state is required to converge to, compute Δ^* for the predictive controller of Proposition 6.2 and use a $\Delta \in (0, \Delta^*]$ for the purpose of MPC implementation.
3. If $x_0 \in \Omega_{x,u}$, proceed to step 6, else continue.
4. For $x_0 \in \Omega_u$, compute the time T_b taken by the bounded controller under discrete implementation, with a discretization step Δ , to drive the state trajectory inside $\Omega_{x,u}$.
5. Implement the controller of Eqs.6.16–6.21 until the time T_{switch} that the state trajectory enters $\Omega_{x,u}$ (this is guaranteed to happen by T_b at the latest), at which time, switch to the predictive controller of Eqs.6.8-6.13.
6. Implement the predictive controller of Eqs.6.8-6.13 to achieve stabilization under input and state constraints.

Remark 6.9: For linear systems, the problem of state constraints satisfaction is typically handled by relaxing the constraints, while appropriately penalizing the state constraint violation within the objective function [244], or solving a multi-objective problem [204] that minimizes both the duration and size of state constraint violation. While these approaches do away with any potential infeasibility due to the state constraints, they do not limit the time for which the state constraints are violated. The use of the bounded controller to obtain an estimate of the time within which the state trajectory can be driven inside $\Omega_{x,u}$ allows the use of this value in the constraint of Eq.6.21 and guarantees state constraint satisfaction by that time. Furthermore, since the objective function minimizes the Lyapunov function value itself at the next time instance, and the target set, $\Omega_{x,u}$, is a level set of V , it is likely that the resulting control action will drive the trajectory inside the feasible region faster, and result in a

smaller time (compared to implementing the bounded controller itself) for which the state constraints are violated (see the simulation example for a demonstration).

Remark 6.10: The problem of implementing MPC with guaranteed stability regions was recently addressed for state [75] and output [145] feedback control of linear systems, for nonlinear systems subject to input constraints [74], and for nonlinear systems subject to input constraints and uncertainty [147], by means of a hybrid predictive control structure that embeds the implementation of MPC within the stability region of a Lyapunov-based bounded controller and uses the bounded controller as a fall-back component that can be switched to in the event of infeasibility or instability of the predictive controller. In this chapter, the switching takes place between two different predictive control formulations (the controller of Eqs.6.16–6.21 and the controller of Eqs.6.8–6.13). Unlike the hybrid predictive control structure, switching between the two controllers here is not to provide a fall back mechanism in the event of infeasibility (the Lyapunov-based predictive controller of Eqs.6.8–6.13 is guaranteed to be feasible from an explicitly characterized set of initial conditions), but rather to use the controller of Eqs.6.16–6.21 to guide the system trajectory into the state and input constrained stability region of the Lyapunov-based predictive control design of Eqs.6.8–6.13. The bounded controller design is used in this chapter, not as a fall-back, but for the purpose of providing an estimate of the feasibility region (and, therefore, the stability region) for the Lyapunov-based predictive controller, and feasible initial guesses for the control moves (the decision variables in the optimization problem).

Remark 6.11: Note that while the closed-loop system is formulated as a switched system (see Eq.6.15), the switching law restricts the number of switches between the two controllers to at most one (the switch takes place only if $x_0 \in \Omega_u \setminus \Omega_{x,u}$). The closed-loop system, therefore, does not need to satisfy the multiple Lyapunov function stability criteria [25, 94], which would have to be satisfied if there were back and forth switchings between the controllers over the infinite time interval. The switching law

of Eq.6.22 also avoids any chattering by allowing only a finite number of switches (in this case, one) over any finite time interval.

6.5 Application to a chemical process example

Consider a continuous stirred tank reactor where an irreversible, first-order exothermic reaction of the form $A \xrightarrow{k} B$ takes place. The mathematical model for the process takes the form:

$$\begin{aligned}\dot{C}_A &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A \\ \dot{T}_R &= \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{Q_\sigma}{\rho c_p V}\end{aligned}\tag{6.23}$$

where C_A denotes the concentration of the species A , T_R denotes the temperature of the reactor, Q is the heat removed from the reactor, V is the volume of the reactor, k_0 , E , ΔH are the pre-exponential constant, the activation energy, and the enthalpy of the reaction and c_p and ρ , are the heat capacity and fluid density in the reactor. The values of all process parameters can be found in Table 6.1. The control objective is to stabilize the reactor at the unstable equilibrium point $(C_A^s, T_R^s) = (0.57 \text{ Kmol/m}^3, 395.3 \text{ K})$, while keeping the state variables between $C_A^{min} = 0.41 \text{ Kmol/m}^3 \leq C_A \leq 0.73 \text{ Kmol/m}^3 = C_A^{max}$ and $T_R^{min} = 392.3 \text{ K} \leq T_R \leq 398.3 \text{ K} = T_R^{max}$ using the rate of heat input, Q , and change in inlet concentration of species A , $\Delta C_A = C_{A0} - C_{A0s}$, as manipulated inputs with constraints: $|Q| \leq 0.167 \text{ KJ/min}$ and $|\Delta C_{A0}| \leq 1 \text{ Kmol/m}^3$. We construct a bounded controller of the form of Eq.6.4 using $V(x) = x'Px$ where $x = (C_A - C_A^s, T_R - T_R^s)$, $P = \begin{bmatrix} 60.2 & 3.82 \\ 3.82 & 0.34 \end{bmatrix}$ where the matrix P was computed using the linearized system. The computation of the stability region estimates, under input constraints Ω_u and under state and input constraints $\Omega_{x,u}$, however, was done using the nonlinear system dynamics, and are shown in Fig.6.2. The parameters in the objective function of Eq.6.12 are chosen as $Q = qI$, with $q = 1.0$, and $R = rI$, with $r = 1.0$ and those in the objective function of Eq.6.17 are chosen as $q = 10.0$ and $R = rI$, with $r = 0.01$. The constrained nonlinear optimization problem is solved using the

MATLAB subroutine fmincon, and the set of ODEs is integrated using the MATLAB solver ODE45.

Table 6.1: Process parameters and steady-state values for the reactor of Eq.6.23.

V	$=$	0.1	m^3
R	$=$	8.314	$KJ/Kmol \cdot K$
C_{A0s}	$=$	0.79	$Kmol/m^3$
T_{A0s}	$=$	352.6	K
Q_s	$=$	0.0	KJ/min
ΔH	$=$	-4.78×10^4	$KJ/Kmol$
k_0	$=$	72×10^9	min^{-1}
E	$=$	8.314×10^4	$KJ/Kmol$
c_p	$=$	0.239	$KJ/Kg \cdot K$
ρ	$=$	1000.0	Kg/m^3
F	$=$	100×10^{-3}	m^3/min
T_{Rs}	$=$	395.33	K
C_{As}	$=$	0.57	$Kmol/m^3$

We first demonstrate the implementation of the Lyapunov-based predictive controller of Proposition 6.2 (Eqs.6.8-6.13) for the case when the state constraints are hard constraints and need to be satisfied at all times. To this end, we consider an initial condition that belongs to the state and input constrained stability region of the predictive controller, $\Omega_{x,u}$. As shown by the solid line in Fig.6.2, starting from the initial condition $(C_A, T_R) = (0.445 \text{ } Kmole/m^3, 398.0 \text{ } K)$, successful stabilization of the closed-loop system is achieved, together with state and input constraint satisfaction for all times. The corresponding state and input profiles are shown in Fig.6.3.

Consider now the case where the state constraints reflect desirable bounds on the state variables, and can be treated as soft constraints. In this case, the closed-loop state could be initialized in Ω_u , from initial conditions where state constraints are initially violated. Starting from an initial condition that violates the state constraint on the temperature, $(C_A, T_R) = (0.678 \text{ } Kmole/m^3, 391.6 \text{ } K)$, it takes 0.42 minutes for the state trajectory to enter $\Omega_{x,u}$ under the implementation of the bounded controller (see dashed lines in Figs.6.2-6.3). Setting $T_b = 0.42$ minutes, therefore and implementing

the predictive controller of Theorem 6.1, we find that the controller is able to drive the state trajectory inside of $\Omega_{x,u}$ at $t = 0.26$ minutes, substantially reducing the time for which the soft constraints are violated (see dotted lines in Figs.6.2-6.3). After $T_{switch} = 0.26$ minutes the predictive controller of Proposition 6.2 is employed to successfully achieve stabilization in the presence of state and input constraints.

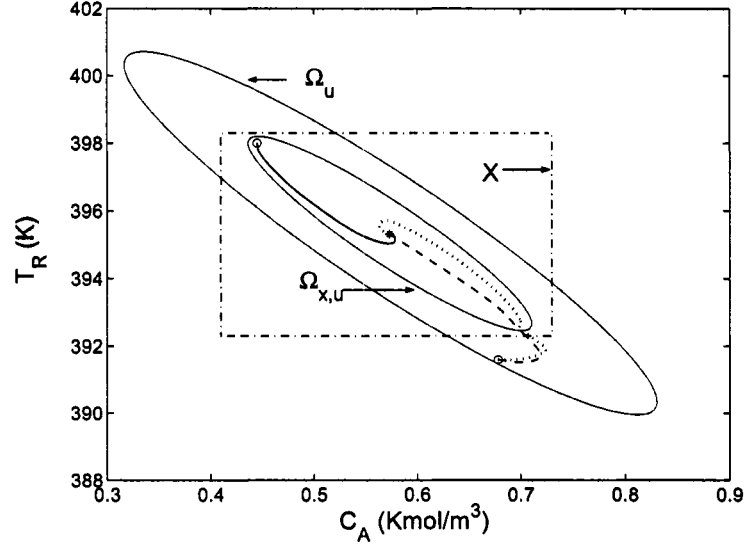


Figure 6.2: Closed-loop state trajectory under the predictive controller of Proposition 6.2 (solid line), under the bounded controller (dashed line) and under the predictive controller of Theorem 6.1 (dotted line).

6.6 Conclusions

In this chapter, we considered the problem of stabilization of nonlinear systems subject to state and control constraints. We proposed a Lyapunov-based MPC design that guarantees stabilization and state and input constraint satisfaction from an explicitly characterized set of initial conditions. An auxiliary Lyapunov-based analytical bounded control design was used to characterize the stability region of the predictive controller and also provide a feasible initial guess to the optimization problem in the predictive controller formulation. For the case when the state constraints are soft, we proposed a switched predictive control strategy that reduces the time for which state constraints

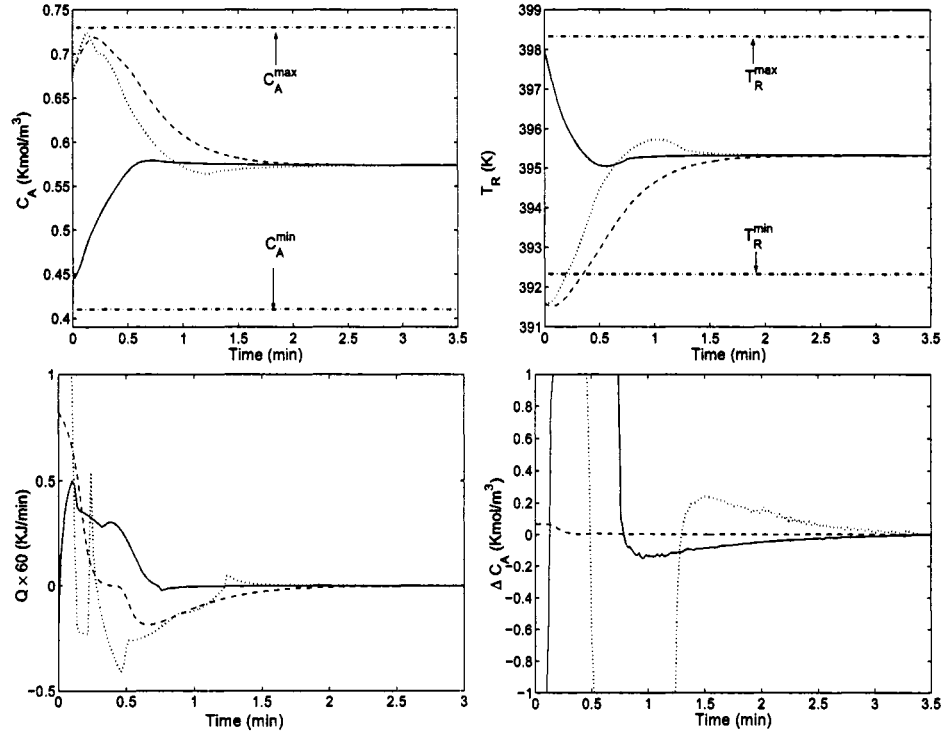


Figure 6.3: Closed-loop state (top) and input (bottom) profiles under the predictive controller of Proposition 6.2 (solid lines), under the bounded controller (dashed lines) and under the predictive controller of Theorem 6.1 (dotted lines).

are violated, driving the states into the state and input constraints feasibility region of the Lyapunov-based predictive controller. We demonstrate the application of the Lyapunov-based predictive controller designs through a chemical process example.

Chapter 7

Predictive Control of Switched Nonlinear Systems With Scheduled Mode Transitions

7.1 Introduction

The operation of processes often involves controlled, discrete transitions between multiple, continuous modes of operation in order to handle, for example, changes in raw materials, energy sources, product specifications and market demands. The superposition of discrete events on the continuous process dynamics gives rise to a hybrid system behavior, i.e., intervals of piecewise continuous behavior interspersed by discrete transitions. Compared with purely continuous systems, the hybrid nature of these systems makes them more difficult to describe, analyze, and control. Motivated by these challenges as well as the abundance of situations where hybrid systems arise in practice, significant research efforts have focused on the study of hybrid systems over the last decade, covering a wide range of problems including, for example, modeling [236], optimization [90], stability analysis [25, 124, 94, 49], and control [128, 26, 93, 235].

A class of hybrid systems that has attracted significant attention recently, because

it can model several practical control problems that involve integration of supervisory logic-based control schemes and feedback control algorithms, is the class of switched (or multi-modal) systems. For this class, results have been developed for stability analysis using the tools of multiple Lyapunov functions (MLFs), for linear [164] and nonlinear [25] systems, and the concept of dwell time [94]; the reader may refer to [124, 49] for a survey of results in this area. These results have motivated the development of methods for control of various classes of switched systems (see, e.g., [230, 96, 93]).

In a recent work [70], a framework for coordinating feedback and switching for control of hybrid nonlinear systems with input constraints was developed. The key feature of the control methodology in [70] is the integrated synthesis of: (1) a family of lower-level bounded nonlinear controllers that stabilize the continuous dynamical modes, and provide an explicit characterization of the stability region associated with each mode, and (2) upper-level switching laws that determine, on the basis of the stability regions, whether or not a transition will preserve stability of the overall switched closed-loop system. While the approach allows one to *determine* whether or not a switch can be made at any given time without loss of stability, it does not address the problem of *ensuring* that such a switch be made safely at some predetermined time.

Guiding the system through a prescribed switching sequence requires a control algorithm that can incorporate both state and input constraints in the control design. One such control method is model predictive control (MPC) and has been studied extensively (see, for example, [149, 115, 170] and the survey paper, [141]). One of the important issues that arise in the possible use of predictive control policies for the purpose of stabilization of the individual modes, however, is the difficulty they typically encounter in identifying, a priori (i.e., before controller implementation), the set of initial conditions starting from where feasibility and closed-loop stability are guaranteed. This typically results in the need for extensive closed-loop simulations to search over the whole set of possible initial conditions, thus adding to the overall

computational load. This difficulty is more pronounced when considering MPC of hybrid systems that involve switching between multiple modes.

For linear systems, the switched system can be transformed into a mixed logical dynamical system, and a mixed-integer linear program can be solved to come up with an optimal switching sequence and switching times [1, 55]. For nonlinear systems, one can, in principle, set up the mixed integer nonlinear programming problem, where the decision variables include the control action together with the switching schedule. The presence of nonlinear dynamics, that typically make the optimization problem non-convex, together with the discrete decision variables results in high computational complexity, and the computation time requirements render it unsuitable for the purpose of real-time control.

In many systems of practical interest, the switched system is required to follow a prescribed switching schedule, where the switching times are no longer decision variables, but are prescribed via an operating schedule. Motivated by this practical problem, we propose a predictive control framework for the constrained stabilization of switched nonlinear systems that transit between their constituent modes at prescribed switching times. The main idea is to design a Lyapunov-based predictive controller for each constituent mode in which the switched system operates, and incorporate constraints in the predictive controller design which upon satisfaction ensure that the prescribed transitions between the modes occur in a way that guarantees stability of the switched closed-loop system. This is achieved as follows: for each constituent mode, a Lyapunov-based model predictive controller (MPC) is designed, and an analytic bounded controller, using the same Lyapunov function, is used to explicitly characterize a set of initial conditions for which the MPC, irrespective of the controller parameters, is guaranteed to be feasible, and hence stabilizing. Then, constraints are incorporated in the MPC design which, upon satisfaction, ensure that: (1) the state of the closed-loop system, at the time of the transition, resides in the stability region

of the mode that the system is switched into, and (2) the Lyapunov function for each mode is non-increasing wherever the mode is re-activated, thereby guaranteeing stability. The proposed control method is demonstrated through application to a chemical process example.

7.2 Preliminaries

We consider the class of switched nonlinear systems represented by the following state-space description

$$\begin{aligned}\dot{x}(t) &= f_{\sigma(t)}(x(t)) + G_{\sigma(t)}(x(t))u_{\sigma(t)}(t) \\ u_{\sigma(t)} &\in \mathcal{U}_{\sigma} \\ \sigma(t) &\in \mathcal{K} := \{1, \dots, p\}\end{aligned}\tag{7.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the vector of continuous-time state variables, $u_{\sigma}(t) = [u_{\sigma}^1(t) \cdots u_{\sigma}^m(t)]^T \in \mathcal{U}_{\sigma} \subset \mathbb{R}^m$ denotes the vector of constrained manipulated inputs taking values in a nonempty compact convex set $\mathcal{U}_{\sigma} := \{u_{\sigma} \in \mathbb{R}^m : \|u_{\sigma}\| \leq u_{\sigma}^{max}\}$, where $\|\cdot\|$ is the Euclidian norm, $u_{\sigma}^{max} > 0$ is the magnitude of the constraints, $\sigma : [0, \infty) \rightarrow \mathcal{K}$ is the switching signal which is assumed to be a piecewise continuous (from the right) function of time, i.e., $\sigma(t_k) = \lim_{t \rightarrow t_k^+} \sigma(t)$ for all k , implying that only a finite number of switches is allowed on any finite interval of time. p is the number of modes of the switched system, $\sigma(t)$, which takes different values in the finite index set $\mathcal{K}$, represents a discrete state that indexes the vector field $f(\cdot)$, the matrix $G(\cdot)$, and the control input $u(\cdot)$, which altogether determine $\dot{x}$. Throughout the chapter, we use the notations $t_{k_r^{in}}$ and $t_{k_r^{out}}$ to denote the time at which, for the r -th time, the k -th subsystem is switched in and out, respectively, i.e., $\sigma(t_{k_r^{in}}^+) = \sigma(t_{k_r^{out}}^-) = k$. With this notation, it is understood that the continuous state evolves according to $\dot{x} = f_k(x) + G_k(x)u_k$ for $t_{k_r^{in}} \leq t < t_{k_r^{out}}$. $\mathcal{T}_{k,in} = \{t_{k_1^{in}}, t_{k_2^{in}}, \dots\}$ and $\mathcal{T}_{k,out} = \{t_{k_1^{out}}, t_{k_2^{out}}, \dots\}$ denote the set of switching times at which the k -th subsystem is switched in and out, respectively. It is assumed that all entries of the vector functions $f_k(x)$, and the $n \times m$ matrices $G_k(x)$, are sufficiently smooth and that $f_k(0) = 0$ for all $k \in \mathcal{K}$.

Throughout the chapter, the notation $L_f \bar{h}$ denotes the standard Lie derivative of a scalar function $\bar{h}(x)$ with respect to the vector function $f(x)$, $L_f \bar{h}(x) = \frac{\partial \bar{h}}{\partial x} f(x)$, and $\limsup_{t \rightarrow \infty} f(x(t)) = \lim_{t \rightarrow \infty} \{\sup_{\tau \geq t} f(x(\tau))\}$.

In order to provide the necessary background for our main results in section 7.3, we will briefly review in the remainder of this section a bounded controller design the stability properties of which are exploited in the design of a Lyapunov-based model predictive controller and in explicitly characterizing its stability region. For simplicity, we focus on the state feedback control problem where measurements of $x(t)$ are assumed to be available for all t .

7.2.1 Bounded Lyapunov-based control

Consider the system of Eq.7.1, for a fixed $\sigma(t) = k$ for some $k \in \mathcal{K}$, for which a control Lyapunov function, V_k , exists. Using the results in [127] (see also [66]), the following continuous bounded control law can be constructed

$$u_k(x) = -k_k(x)(L_{G_k} V_k)^T(x) := b_k(x) \quad (7.2)$$

$$k_k(x) = \frac{L_{f_k}^* V_k(x) + \sqrt{\left(L_{f_k}^* V_k(x)\right)^2 + (u_k^{max} \|(L_{G_k} V_k)^T(x)\|)^4}}{\|(L_{G_k} V_k)^T(x)\|^2 \left[1 + \sqrt{1 + (u_k^{max} \|(L_{G_k} V_k)^T(x)\|)^2}\right]} \quad (7.3)$$

$L_{G_k} V_k(x) = [L_{g_k^1} V_k \cdots L_{g_k^m} V_k]$ is a row vector, where g_k^i is the i -th column of G_k , $L_{f_k}^* V_k = L_{f_k} V_k + \rho_k V_k$ and $\rho_k > 0$. For the above controller, one can compute an estimate of the stability region that is described by

$$\Omega_k(u_k^{max}) = \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\} \quad (7.4)$$

where $c_k^{max} > 0$ is the largest number for which $\Phi_k(u_k^{max}) \supseteq \Omega_k(u_k^{max}) \setminus \{0\}$, and

$$\Phi_k(u_k^{max}) = \{x \in \mathbb{R}^n : L_{f_k}^* V_k(x) < u_k^{max} \|(L_{G_k} V_k)^T(x)\|\} \quad (7.5)$$

The bounded controller of Eqs.7.2-7.3 possesses a robustness property with respect to measurement errors, that preserves closed-loop stability when the control action

is implemented in a discrete (sample and hold) fashion with a sufficiently small hold time (Δ). Specifically, the control law ensures that, for all initial conditions in Ω_k , the closed-loop state remains in Ω_k and eventually converges to some neighborhood of the origin whose size depends on Δ . This robustness property, formalized below in Proposition 7.1, will be exploited in the Lyapunov-based predictive controller design of section 7.2.2. For further results on the analysis and control of sampled-data nonlinear systems, the reader may refer to [89, 158, 107, 240].

Proposition 7.1: *Consider the constrained system of Eq.7.1 for a fixed value of $\sigma(t) = k$, under the bounded control law of Eqs.7.2–7.3 designed using the Lyapunov function V_k and $\rho_k > 0$, and the stability region estimate Ω_k under continuous implementation. Let $u_k(t) = u_k(j\Delta)$ for all $j\Delta \leq t < (j+1)\Delta$ and $u_k(j\Delta) = b_k(x(j\Delta))$, $j = 0, \dots, \infty$. Then, given any positive real number d_k , there exists a positive real number Δ_k^* such that if $x(0) := x_0 \in \Omega_k$ and $\Delta \in (0, \Delta_k^*]$, then $x(t) \in \Omega_k \forall t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$.*

Proof of Proposition 7.1: The proof consists of two parts. In the first part, we establish that the bounded state feedback control law of Eqs.7.2–7.3 enforces asymptotic stability for all initial conditions in Ω_k with a certain robustness margin. In the second part, given the size, d_k , of a ball around the origin that the system is required to converge to, we show the existence of a positive real number Δ_k^* , such that if the discretization time Δ is chosen to be in $(0, \Delta_k^*]$, then Ω_k remains invariant under discrete implementation of the bounded control law, and also that the state of the closed-loop system converges to the ball $\|x\| \leq d_k$.

Part 1: Substituting the control law of Eqs.7.2–7.3 into the system of Eq.7.1 for a fixed $\sigma(t) = k$ it can be shown that:

$$\dot{V}_k(x) = L_{f_k} V_k(x) + L_{G_k} V_k(x) u(x) \leq \frac{-\rho_k V_k}{\left[1 + \sqrt{1 + (u_k^{max} \|(L_{G_k} V_k)^T(x)\|)^2}\right]} \quad (7.6)$$

for all $x \in \Omega_k$, where Ω_k was defined in Eqs.7.4. Since the denominator term in Eq.7.6

is bounded in Ω_k , there exists a positive real number, ρ_k^* , such that $\dot{V}_k \leq -\rho_k^* V_k$ for all $x \in \Omega_k$, which implies that the origin of the closed-loop system, under the control law of Eqs.7.2–7.3, is asymptotically stable, with Ω_k as an estimate of the domain of attraction.

Part 2: Note that since $V_k(\cdot)$ is a continuous function of the state, one can find a finite, positive real number, δ'_k , such that $V_k(x) \leq \delta'_k$ implies $\|x\| \leq d_k$. In the rest of the proof, we show the existence of a positive real number Δ_k^* such that all state trajectories originating in Ω_k converge to the level set of V_k ($V_k(x) \leq \delta'_k$) for any value of $\Delta \in (0, \Delta_k^*]$ and hence we have that $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$.

To this end, consider a “ring” close to the boundary of the stability region, described by $\mathcal{M}_k := \{x \in \mathbb{R}^n : (c_k^{max} - \delta_k) \leq V_k(x) \leq c_k^{max}\}$ (see also Fig.7.1), for a $0 \leq \delta_k < c_k^{max}$. Let the control action be computed for some $x(0) := x_0 \in \mathcal{M}_k$ and held constant until a time Δ_k^{**} , where Δ_k^{**} is a positive real number ($u_k(t) = u_k(x_0) := u_0 \forall t \in [0, \Delta_k^{**}]$). Then, $\forall t \in [0, \Delta_k^{**}]$,

$$\begin{aligned} \dot{V}_k(x(t)) &= L_{f_k} V_k(x(t)) + L_{G_k} V_k(x(t)) u_0 \\ &= L_{f_k} V_k(x_0) + L_{G_k} V_k(x_0) u_0 + (L_{f_k} V_k(x(t)) - L_{f_k} V_k(x_0)) \\ &\quad + (L_{G_k} V_k(x(t)) u_0 - L_{G_k} V_k(x_0) u_0) \end{aligned} \quad (7.7)$$

Since the control action is computed based on the states in $\mathcal{M}_k \subseteq \Omega_k$, $L_{f_k} V_k(x_0) + L_{G_k} V_k(x_0) u_0 \leq -\rho_k^* V_k(x_0)$. By definition, for all $x_0 \in \mathcal{M}_k$, $V_k(x_0) \geq c_k^{max} - \delta_k$, therefore $L_{f_k} V_k(x_0) + L_{G_k} V_k(x_0) u_0 \leq -\rho_k^* (c_k^{max} - \delta_k)$.

Since the function $f_k(\cdot)$ and the elements of the matrix $G_k(\cdot)$ are continuous, $\|u_k\| \leq u_k^{max}$, and $\mathcal{M}_k$ is bounded, then one can find, for all $x_0 \in \mathcal{M}_k$ and a fixed Δ_k^{**} , a positive real number K_k^1 , such that $\|x(t) - x_0\| \leq K_k^1 \Delta_k^{**}$ for all $t \leq \Delta_k^{**}$.

Since the functions $L_{f_k} V_k(\cdot)$, $L_{G_k} V_k(\cdot)$ are continuous, then given that $\|x(t) - x_0\| \leq K_k^1 \Delta_k^{**}$, one can find positive real numbers K_k^2 and K_k^3 such that $\|L_{f_k} V_k(x(t)) - L_{f_k} V_k(x_0)\| \leq K_k^3 K_k^1 \Delta_k^{**}$ and $\|L_{G_k} V_k(x(t)) u_0 - L_{G_k} V_k(x_0) u_0\| \leq K_k^2 K_k^1 \Delta_k^{**}$. Using these inequalities in Eq.7.7, we get

$$\dot{V}_k(x(t)) \leq -\rho_k^* (c_k^{max} - \delta_k) + (K_k^1 K_k^2 + K_k^1 K_k^3) \Delta_k^{**} \quad (7.8)$$

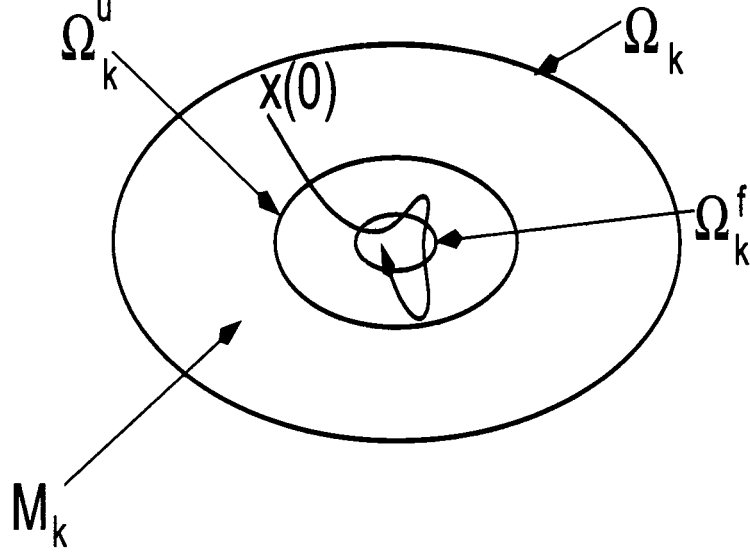


Figure 7.1: A schematic representation of the stability region and the evolution of the closed-loop state trajectory under discrete implementation of the bounded controller.

For a choice of $\Delta_k^{**} < \frac{\rho_k^*(c_k^{max} - \delta_k) - \epsilon_k}{(K_k^1 K_k^2 + K_k^1 K_k^3)}$ where ϵ_k is a positive real number such that

$$\epsilon_k < \rho_k^*(c_k^{max} - \delta_k) \quad (7.9)$$

we get that $\dot{V}_k(x(t)) \leq -\epsilon_k < 0$ for all $t \leq \Delta_k^{**}$. This implies that, given δ'_k , if we pick δ_k such that $c_k^{max} - \delta_k < \delta'_k$ and find a corresponding value of Δ_k^{**} then if the control action is computed for any $x \in \mathcal{M}_k$, and the 'hold' time is less than Δ_k^{**} , we get that $\dot{V}_k$ remains negative during this time, and therefore the state of the closed-loop system cannot escape Ω_k (since Ω_k is a level set of V_k). We now show the existence of Δ'_k such that for all $x_0 \in \Omega_k^f := \{x \in \mathbb{R}^n : V_k(x_0) \leq c_k^{max} - \delta_k\}$, we have that $x(\Delta) \in \Omega_k^u := \{x_0 \in \mathbb{R}^n : V_k(x_0) \leq \delta'_k\}$, where $\delta'_k < c_k^{max}$, for any $\Delta \in (0, \Delta'_k]$.

Consider Δ'_k such that

$$\delta'_k = \max_{V_k(x_0) \leq c_k^{max} - \delta_k, u_k \in \mathcal{U}_k, t \in [0, \Delta'_k]} V_k(x(t)) \quad (7.10)$$

Since V_k is a continuous function of x , and x evolves continuously in time, then for any value of $\delta_k < c_k^{max}$, one can choose a sufficiently small Δ'_k such that Eq.7.10 holds. Let

$\Delta_k^* = \min\{\Delta_k^{**}, \Delta_k'\}$. We now show that for all $x_0 \in \Omega_k^u$ and $\Delta_k \in (0, \Delta_k^*]$, $x(t) \in \Omega_k^u$ for all $t \geq 0$.

For all $x_0 \in \Omega_k^u \cap \Omega_k^f$, by definition $x(t) \in \Omega_k^u$ for $0 \leq t \leq \Delta_k$ (since $\Delta_k \leq \Delta_k'$). For all $x_0 \in \Omega_k^u \setminus \Omega_k^f$ (and therefore $x_0 \in \mathcal{M}_k$), $\dot{V}_k < 0$ for $0 \leq t \leq \Delta$ (since $\Delta_k \leq \Delta_k^{**}$). Since Ω_k^u is a level set of V_k , then $x(t) \in \Omega_k^u$ for $0 \leq t \leq \Delta_k$. Either way, for all initial conditions in Ω_k^u , $x(t) \in \Omega_k^u$ for all future times.

We note that for x such that $x \in \Omega_k \setminus \Omega_k^u$, negative definiteness of $\dot{V}_k$ is guaranteed for $\Delta \leq \Delta_k^* \leq \Delta_k^{**}$. Hence, all trajectories originating in Ω_k converge to Ω_k^u , which has been shown to be invariant under the bounded control law with a hold time Δ less than Δ_k^* , and therefore, for all $x(0) \in \Omega_k$, $\limsup_{t \rightarrow \infty} V_k(x(t)) \leq \delta_k'$. Finally, since $V_k(x) \leq \delta_k'$ implies $\|x\| \leq d_k$, therefore we have that $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$. This completes the proof of Proposition 7.1.

Remark 7.1: Control Lyapunov function (CLF)-based stabilization of nonlinear systems has been studied extensively in the nonlinear control literature (e.g., see [9, 127, 207]). The construction of constrained CLFs (i.e., CLFs that take the constraints into account) remains a difficult problem (especially for nonlinear systems) that is the subject of ongoing research. For several classes of nonlinear systems that arise commonly in the modeling of engineering systems, systematic and computationally feasible methods are available for constructing unconstrained CLFs (CLFs for the unconstrained system) by exploiting the system structure. Examples include the use of quadratic functions for feedback linearizable systems and the use of back-stepping techniques to construct CLFs for systems in strict feedback form. In this chapter, the bounded controllers in Eqs.7.2-7.3 are designed using unconstrained CLFs, which are also used to explicitly characterize the associated regions of stability via Eqs.7.4-7.5. While the resulting estimates do not necessarily capture the entire domain of attraction (this remains an open problem even for linear systems), we will use them throughout the chapter for a concrete illustration of the results. It is possible to obtain substantially improved

estimates by using a combination of several CLFs (see, for example, [74, 147]).

7.2.2 Lyapunov-based predictive control

In this section, we consider predictive control of the system of Eq.7.1, for a fixed $\sigma(t) = k$ for some $k \in \mathcal{K}$. We present here a Lyapunov-based design of MPC that guarantees feasibility of the optimization problem and hence constrained stabilization of the closed-loop system from an explicitly characterized set of initial conditions. For this predictive control design, the control action at state x and time t is obtained by solving, on-line, a finite horizon optimal control problem of the form

$$P(x, t) : \min\{J(x, t, u_k(\cdot)) | u_k(\cdot) \in S_k\} \quad (7.11)$$

$$s.t. \quad \dot{x} = f_k(x) + G_k(x)u_k \quad (7.12)$$

$$\dot{V}_k(x(\tau)) \leq -\epsilon_k \text{ if } V_k(x(t)) > \delta'_k, \tau \in [t, t + \Delta) \quad (7.13)$$

$$V_k(x(\tau)) \leq \delta'_k \text{ if } V_k(x(t)) \leq \delta'_k, \tau \in [t, t + \Delta) \quad (7.14)$$

where ϵ_k, δ'_k are defined in Eq.7.9 and Proposition 7.1, respectively, $S_k = S_k(t, T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ , mapping $[t, t + T]$ into $\mathcal{U}_k$, T is the specified horizon and V_k is the Lyapunov function used in the bounded controller design. A control $u_k(\cdot)$ in S_k is characterized by the sequence $\{u_k[j]\}$ where $u_k[j] := u_k(j\Delta)$ and satisfies $u_k(t) = u_k[j]$ for all $t \in [j\Delta, (j+1)\Delta)$. The performance index is given by

$$J(x, t, u_k(\cdot)) = \int_t^{t+T} [\|x^u(s; x, t)\|_Q^2 + \|u_k(s)\|_R^2] ds \quad (7.15)$$

where Q is positive semi-definite symmetric matrix, and R is strictly positive definite symmetric matrix. $x^u(s; x, t)$ denotes the solution of Eq.7.1, due to control u , with initial state x at time t . The minimizing control $u_k^0(\cdot) \in S_k$ is then applied to the plant over the interval $[t, t + \Delta)$ and the procedure is repeated indefinitely. Stability

properties of the closed-loop system under the Lyapunov-based predictive controller are inherited from the bounded controller under discrete implementation and are formalized in Proposition 7.2 below.

Proposition 7.2: *Consider the constrained system of Eq.7.1 for a fixed value of $\sigma(t) = k$ under the MPC control law of Eqs.7.11–7.15, designed using a control Lyapunov function V_k that yields a stability region Ω_k under continuous implementation of the bounded controller of Eqs.7.2–7.3 with a fixed $\rho_k > 0$. Then, given any positive real number d_k , there exist positive real numbers Δ_k^* and δ'_k , such that if $x(0) \in \Omega_k$ and $\Delta \in (0, \Delta_k^*]$, then $x(t) \in \Omega_k \forall t \geq 0$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$.*

Proof of Proposition 7.2: From the proof of Proposition 7.1, we infer that given a positive real number, d_k , there exists an admissible manipulated input trajectory (provided by the bounded controller), and values of Δ_k^* and δ'_k , such that for any $\Delta \in (0, \Delta_k^*]$ and $x(0) \in \Omega_k$, $\limsup_{t \rightarrow \infty} V_k(x(t)) \leq \delta'_k$ and $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$. The rest of the proof is divided in three parts. In the first part we show that for all $x_0 \in \Omega_k$, the predictive controller of Eqs.7.11–7.15 is feasible. We then show that Ω_k is invariant under the predictive control algorithm of Eqs.7.11–7.15. Finally, we prove practical stability for the closed-loop system.

Part 1: Consider some $x_0 \in \Omega_k$ under the predictive controller of Eqs.7.11–7.15, with a prediction horizon $T = N\Delta$, where Δ is the hold time and $1 \leq N < \infty$ is the number of prediction steps. The initial condition can either be such that $V_k(x_0) \leq \delta'$ or $\delta'_k < V_k(x_0) \leq c_k^{max}$.

Case 1: If $\delta'_k < V_k(x_0) \leq c_k^{max}$, the control input trajectory under the bounded controller of Eqs.7.2–7.3 provides a feasible solution to the constraint of Eq.7.13 (see Proposition 7.1), given by $u(j\Delta) = b_k(x(j\Delta))$, $j = 1, \dots, N$. Note that if $u = b_k$ for $t \in [0, \Delta]$, and $\Delta \in (0, \Delta_k^*]$, then $\dot{V}_k(x(t)) \leq -\epsilon_k < 0 \forall t \in [0, \Delta]$ and $b_k(x(t)) \in U_k$ (since b_k is computed using the bounded controller of Eqs.7.2–7.3).

Case 2: If $V_k(x_0) \leq \delta'_k$, once again we infer from Proposition 7.1 that the control input

trajectory provided by the bounded controller of Eqs.7.2–7.3 provides a feasible initial guess, given by $u(j\Delta) = b_k(x(j\Delta))$, $j = 1, \dots, N$ (recall from Proposition 7.1, that under the bounded controller of Eqs.7.2–7.3, if $V_k(x_0) \leq \delta'_k$ then $V_k(x(t)) \leq \delta'_k \forall t \geq 0$). This shows that for all $x_0 \in \Omega_k$, the Lyapunov based predictive controller of Eqs.7.11–7.15 is feasible.

Part 2: As shown in Part 1, for any $x_0 \in \Omega_k \setminus \Omega_k^u$, the constraint of Eq.7.13 in the optimization problem is feasible. Upon implementation, therefore, the value of the Lyapunov function decreases. Since Ω_k is a level set of V_k , the state trajectories cannot escape Ω_k . On the other hand, if $x_0 \in \Omega_k^u$, feasibility of the constraint of Eq.7.14 guarantees that the closed-loop state trajectory stays in $\Omega_k^u \subset \Omega_k$. In both cases, Ω_k continues to be an invariant region under the Lyapunov based predictive controller of Eqs.7.11–7.13.

Part 3: Finally, consider an initial condition $x_0 \in \Omega_k \setminus \Omega_k^u$. Since the optimization problem continues to be feasible, we have that $\dot{V}_k < 0$ for all $x(t) \notin \Omega_k^u$ i.e., $V_k(x(t)) > \delta'_k$. All trajectories originating in Ω_k , therefore converge to Ω_k^u . For $x_0 \in \Omega_k^u$, the feasibility of the optimization problem implies $x(t) \in \Omega_k^u$, i.e., $V_k(x(t)) \leq \delta'_k$. Therefore, for all $x_0 \in \Omega_k$, $\limsup_{t \rightarrow \infty} V_k(x(t)) \leq \delta'_k$. Also, since $V_k(x) \leq \delta'_k$ implies $\|x\| \leq d_k$, we have that $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$. This completes the proof of Proposition 7.2.

Remark 7.2: Note that Lyapunov-based predictive control approaches (see, for example, [115, 170]) typically incorporate a similar Lyapunov function decay constraint, albeit requiring the constraint of Eq.7.13 to hold at the *end* of the prediction horizon as opposed to during the first time step, and assume the initial feasibility of this constraint. In contrast, the predictive controller formulation of Eqs.7.11–7.15 requires that the value of the Lyapunov function decrease during the first step only, allowing for the use of the bounded controller as an auxiliary controller to explicitly characterize the set of initial conditions stating from where the predictive controller is guaranteed to be feasible.

Remark 7.3: The fact that only practical stability is achieved is not a limitation of the MPC formulation, but is due to discrete implementation. Even if the bounded controller is used instead, under the same implement-and-hold time of Δ , the bounded controller can also only guarantee that the state of the closed-loop system converges to the set Ω_k^u , the size of which is limited by the value of the hold time, Δ (in the limit as Δ goes to zero – continuous implementation – the bounded controller and the predictive controller enforce asymptotic stability). Note also that any other Lyapunov-based analytic control design that provides an explicit characterization of the stability region and is robust with respect to discrete implementation can be used as an auxiliary controller.

Remark 7.4: One of the key challenges that impact on the practical implementation of NMPC is the inherent difficulty of characterizing, *a priori*, the set of initial conditions starting from where a given NMPC controller is guaranteed to stabilize the closed-loop system, or for a given set of initial conditions, to identify the value of the prediction horizon for which the optimization problem will be feasible. Use of conservatively large horizon lengths to address stability only increases the size and complexity of the nonlinear optimization problem and could make it intractable. Owing to the fact the closed-loop stability is guaranteed by the Lyapunov-based predictive controller from an explicitly characterized set of initial conditions, irrespective of the prediction horizon, the time required for the computation of the control action, if so desired, can be made smaller by reducing the size of the optimization problem by decreasing the prediction horizon.

Remark 7.5: In the event that measurements are not continuously available, but are available only at sampling times $\Delta_s > \Delta_k^*$, i.e., greater than what a given bounded control design can tolerate (and, therefore, greater than the maximum allowable discretization for the Lyapunov-based predictive controller), it is necessary to redesign the bounded controller to increase the robustness margin, and generate a revised estimate

of the feasibility (and stability) region under the predictive controller. A larger value of Δ_k^* may be achieved by increasing the value of the parameter ρ_k in the design of the bounded controller (see proof of Proposition 7.1). If the value of the sampling time is reasonable, an increase in the value of the parameter ρ_k , while leading to a shrinkage in the stability region estimate, can increase Δ_k^* to a value greater than Δ_s and preserve the desired feasibility and stability guarantees of the Lyapunov-based predictive controller.

7.3 Predictive control of switched nonlinear systems

Consider now the nonlinear switched system of Eq.7.1, with a prescribed switching sequence (including the switching times) defined by $\mathcal{T}_{k,in} = \{t_{k_1^{in}}, t_{k_2^{in}}, \dots\}$ and $\mathcal{T}_{k,out} = \{t_{k_1^{out}}, t_{k_2^{out}}, \dots\}$. Also, assume that for each mode of the switched system, a Lyapunov-based predictive controller of the form of Eqs.7.11-7.15 has been designed and an estimate of the stability region generated. The control problem is formulated as the one of designing a Lyapunov-based predictive controller that guides the closed-loop system trajectory in a way that the schedule described by the switching times is followed and stability of the closed-loop system is achieved. A predictive control algorithm that addresses this problem is presented in Theorem 7.1 below.

Theorem 7.1: *Consider the constrained nonlinear system of Eq.7.1, the control Lyapunov functions V_k , $k = 1, \dots, p$, and the stability region estimates Ω_k , $k = 1, \dots, p$ under continuous implementation of the bounded controller of Eqs.7.2-7.3 with fixed $\rho_k > 0$, $k = 1, \dots, p$. Let $0 < T_{design} < \infty$ be a design parameter. Let t be such that $t_{k_1^{in}} \leq t < t_{k_2^{out}}$ and $t_{m_j^{in}} = t_{k_2^{out}}$ for some m, k . Consider the following optimization problem*

$$P(x, t) : \min\{J(x, t, u_k(\cdot)) | u_k(\cdot) \in S_k\} \quad (7.16)$$

$$J(x, t, u_k(\cdot)) = \int_t^{t+T} [\|x^u(s; x, t)\|_Q^2 + \|u_k(s)\|_R^2] ds \quad (7.17)$$

where T is the prediction horizon given by $T = t_{k_r^{out}} - t$, if $t_{k_r^{out}} < \infty$ and $T = T_{design}$ if $t_{k_r^{out}} = \infty$, subject to the following constraints

$$\dot{x} = f_k(x) + G_k(x)u_k \quad (7.18)$$

$$\dot{V}_k(x(\tau)) \leq -\epsilon_k \text{ if } V_k(x(t)) > \delta'_k, \tau \in [t, t + \Delta) \quad (7.19)$$

$$V_k(x(\tau)) \leq \delta'_k \text{ if } V_k(x(t)) \leq \delta'_k, \tau \in [t, t + \Delta_{k_r}) \quad (7.20)$$

$$V_m(x(t_{m_j^{in}})) \leq \begin{cases} V_m(x(t_{m_{j-1}^{in}})) - \epsilon^* & , \quad j > 1, V_m(x(t_{m_{j-1}^{in}})) > \delta'_m \\ \delta'_m & , \quad j > 1, V_m(x(t_{m_{j-1}^{in}})) \leq \delta'_m \\ c_m^{max} & , \quad j = 1 \end{cases} \quad (7.21)$$

if $t_{k_r^{out}} = t_{m_j^{in}} < \infty$ where ϵ^* is a positive real number. Then, given a positive real number d^{max} , there exist positive real numbers Δ^* and δ'_k , $k = 1, \dots, m$ such that if the optimization problem of Eqs.7.16–7.21 is feasible at all times, the minimizing control is applied to the system over the interval $[t, t + \Delta_{k_r}]$, where $\Delta_{k_r} \in (0, \Delta^*]$ and $t_{k_r^{out}} - t_{k_r^{in}} = l_{k_r} \Delta_{k_r}$ for some integer $l_{k_r} > 0$ and the procedure is repeated, then, $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d^{max}$.

Proof of Theorem 7.1: The proof of this theorem follows from the assumption of feasibility of the constraints of Eqs.7.19–7.21 at all times. Given the radius of the ball around the origin, d^{max} , the value of δ'_k and Δ_k^* for all $k \in \mathcal{K}$ is computed the same way as in the proof of Proposition 7.1. Then, for the purpose of MPC implementation, a value of $\Delta_{k_r} \in (0, \Delta^*]$ is chosen where $\Delta^* = \min_{k=1, \dots, p} \Delta_k^*$ and $t_{k_r^{out}} - t_{k_r^{in}} = l_{k_r} \Delta_{k_r}$ for some integer $l_{k_r} > 0$ (note that given any two positive real numbers $t_{k_r^{out}} - t_{k_r^{in}}$ and Δ^* , one can always find a positive real number $\Delta_{k_r} \leq \Delta^*$ such that $t_{k_r^{out}} - t_{k_r^{in}} = l_{k_r} \Delta_{k_r}$ for some integer $l_{k_r} > 0$).

Part 1: First consider the case when the switching is infinite. Let t be such that $t_{k_r^{in}} \leq t < t_{k_r^{out}}$ and $t_{m_j^{in}} = t_{k_r^{out}} < \infty$. Consider the active mode k . If $V_k(x) > \delta'_k$, the continued feasibility of the constraint of Eq.7.19 implies that $V_k(x(t_{k_r^{out}})) < V_k(x(t_{k_r^{in}}))$. The transition constraint of Eq.7.21 ensures that if this mode is switched

out and then switched back in, then $V_k(x(t_{k_{r+1}}^{in})) < V_k(x(t_{k_r}^{in}))$. In general $V_k(x(t_{k_i}^{in})) < V_k(x(t_{k_{i-1}}^{in})) < \dots < c_k^{max}$. Under the assumption of feasibility of the constraints of Eqs.7.19-7.21 for all future times, therefore, the value of $V_k(x)$ continues to decrease. If the mode of this Lyapunov function is not active, there exists at least some $j \in 1, \dots, p$ such that mode j is active and Lyapunov function V_j continues to decrease until the time that $V_j \leq \delta'_j$ (this happens because there are finite number of modes, even if the number of switches may be infinite). From this point onwards, the constraint of Eq.7.20 ensures that V_j continues to be less than δ'_j . Hence $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d^{max}$.

Part 2: For the case of a finite switching sequence, consider a t such that $t_{k_r}^{in} \leq t < t_{k_r}^{out} = \infty$. Under the assumption of continued feasibility of Eqs.7.19-7.21, $V_k(x(t_{k_r}^{in})) < V_k(x(t_{k_{r-1}}^{in})) < \dots < c_k^{max}$. At the time of the switch to mode k , therefore, $x(t_{k_r}^{in}) \in \Omega_k$. From this point onwards, the Lyapunov based controller is implemented using the Lyapunov function V_k , and the constraint of Eq.7.21 is removed, in which case the predictive controller of Theorem 7.1 reduces to the predictive controller of Eqs.7.11-7.15. Since the value of Δ_{k_r} is chosen to be in $(0, \Delta^*]$, where $\Delta^* = \min_{k=1, \dots, p} \Delta_k^*$, therefore $\Delta_{k_r} < \Delta_k^*$, which guarantees feasibility and convergence to the ball $\|x\| \leq d^{max}$ for any value of the prediction horizon (hence, for a choice of $T = T_{design}$), and leads to $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d^{max}$. This completes the proof of Theorem 7.1.

Remark 7.6: Note that the constraint of Eq.7.19 is guaranteed to be feasible between mode transitions, provided that the system is initialized within the stability region, and does not require the assumption of feasibility. This stability constraint ensures that the value of the Lyapunov function of the currently active mode keeps decreasing (recall that one of the criteria in the multiple Lyapunov-function stability analysis is that the individual modes of the switched system be stable). The constraint of Eq.7.19 expresses two transition requirements simultaneously: (1) the MLF constraints that requires that the value of the Lyapunov function be less than what it was the last time the system switched into that mode (required when the switching sequence is infinite, see [25] for

details), and (2) the stability region constraint that requires that the state of the process reside within the stability region of the target mode at the time of the switch; since the stability regions of the modes are expressed as level sets of the Lyapunov functions, the MLF-constraint also expresses the stability region constraint. The understanding that it is a reasonably chosen switching schedule (that is, one that does not result in closed-loop instability), motivates assuming the feasibility of the transition constraints for all times. Note that the feasibility of the transition constraints can also be used to validate the switching schedule, and can be used to abort the switching schedule (i.e., to decide that the remaining switches should not be carried out) in the interest of preserving closed-loop stability.

Remark 7.7: Fig.7.2 is a schematic that depicts the main features of the predictive controller for mode transitions, for representative switching between two modes, where the switching schedule dictates the transitions, and is accounted for in the predictive control design. A representative implementation of the predictive control algorithm of Theorem 7.1 is also depicted schematically by the solid lines in Fig. 7.3. The algorithm below explains the implementation of the predictive controller of Theorem 7.1, including the course of action that needs to be taken if the predictive controller does not yield a feasible solution

1. Given the system model of Eq.7.1 and the constraints on the input, design the bounded controller of Eqs.7.2–7.3 for each mode and compute the stability region estimate, $\Omega_k(u_k^{max})$, for the bounded controller using Eqs.7.4–7.5.
2. Given the size of the ball, d^{max} , that the state is required to converge to, compute Δ_k^* , $k = 1, \dots, p$ for the predictive controller of Eqs.7.11–7.15, such that under single mode operation for each mode, $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d^{max}$. Compute $\Delta^* = \min_{k=1}^p \{\Delta_k^*\}$, and use a $\Delta_{k_r} \in (0, \Delta^*]$ for which $t_{k_r^{out}} - t_{k_r^{in}} = l_{k_r} \Delta_{k_r}$ for some integer $l_{k_r} > 0$, for the purpose of MPC implementation.
3. Consider the time $t_{k_r^{in}}$, which designates the time that the closed-loop system

switches to the k -th mode for the r -th time, and the state belongs to the stability region of the k -th mode (for the purpose of initialization, i.e., at $t = 0$, this would correspond to t_1^{in} , and the state belonging to the stability region $\Omega_1(u_1^{max})$).

4. From the set of prescribed switching times, pick $t_{m_j^{in}} = t_{k^{out}}$ ($t_{m_j^{in}}$, therefore, is the time that the next switch takes place, and that the system, upon exiting from the current mode enters mode m for the j -th time).
5. Consider the predictive controller of Theorem 7.1. The constraint $V_m(t_{m_j^{in}}) \leq V_m(t_{m_{j-1}^{in}}) - \epsilon^*$ requires that when the closed-loop system enters the mode m , the value of V_m is less than what it was at the time that the system last entered mode m (this is a version of the multiple Lyapunov function stability condition, see [25]). If the system has never entered mode m before, i.e., for $j = 1$, set $V_m(t_{m_j^{in}}) = c_m^{max}$ (this requires that the state belongs to the stability region corresponding to mode m). If the closed-loop state has already entered the desired ball around the origin, implement $V_m(t_{m_j^{in}}) \leq \delta'_m$, that ensures that the state stays within the ball, $\|x\| \leq d^{max}$.
6. If at any time, the predictive controller of Theorem 7.1 does not yield a feasible solution, or the switching schedule does not prescribe another switch, go to step 7; else implement the predictive controller up-to time $t_{k^{out}}$, i.e., until the time that the system switches into mode m and go back to step 4 to proceed with the rest of the switching sequence.
7. Implement the Lyapunov-based predictive controller of Eqs.7.11-7.15 for the current mode to stabilize the closed-loop system.

Remark 7.8: Note that since the switching times are fixed, the prediction of the states in the controller needs to be carried out from the current time up-to the time of the next switch only. The predictive controller is therefore implemented with a shrinking horizon between successive switching times. Note, however, that the value of the horizon is *not*

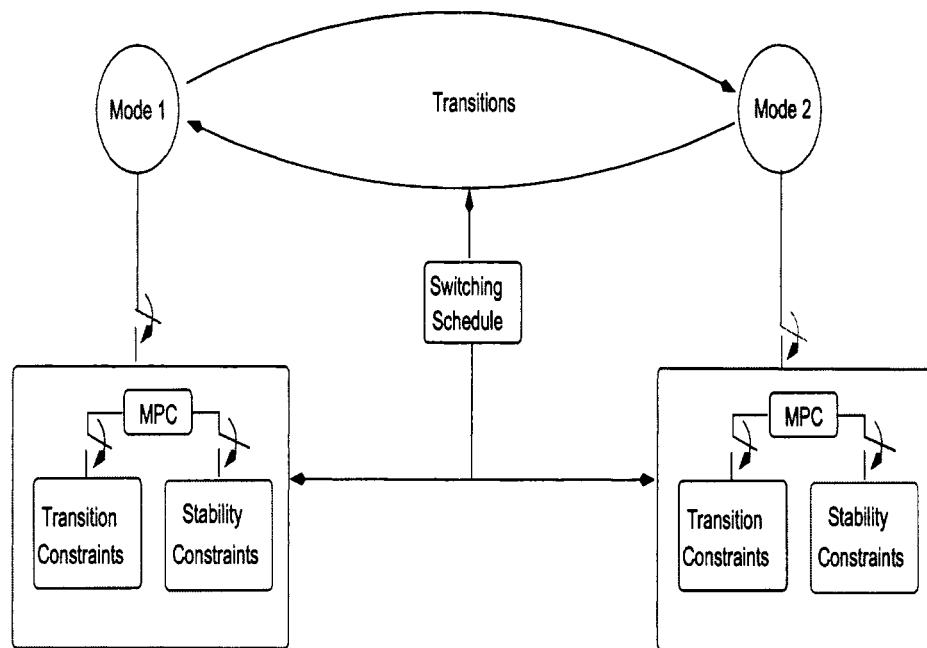


Figure 7.2: A schematic representation of the predictive control structure comprised of the predictive and bounded controllers for the constituent modes, together with transition constraints.

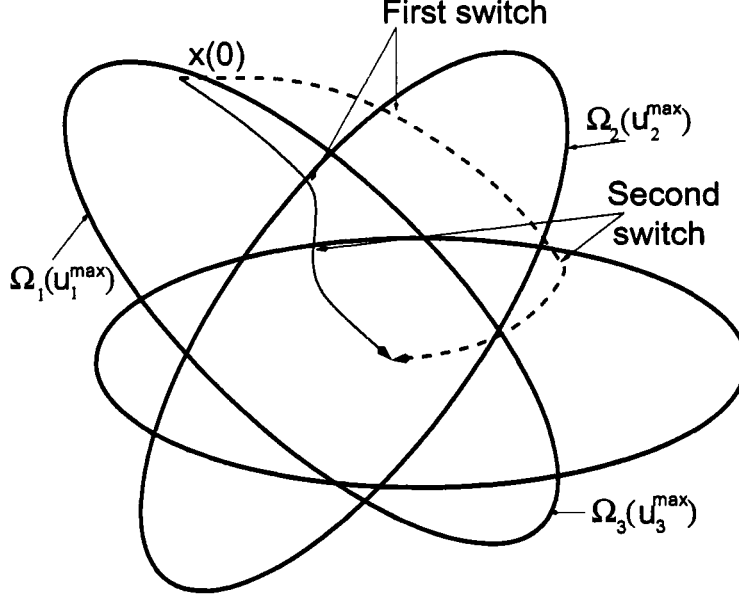


Figure 7.3: Schematic representation of the closed-loop trajectory of a switched system under the implementation of the predictive control algorithm of Theorem 7.1 (solid line) and using the relaxed constraints of Remark 7.9 (dashed line).

a decision variable (and therefore does not incur any computational burden); its value is obtained simply by evaluating the difference between the next switching time and the current time. However, in the case that the switching schedule terminates, then after the last switch has been made and the system is evolving in the terminal mode, the horizon is fixed and set equal to a preset design parameter T_{design} . Note that from this point onward, the controller design of Eqs.7.16-7.20 reduces to the Lyapunov based predictive controller of Eqs.7.11-7.15 for the terminal mode, and guarantees practical stability of the closed-loop system for any value of prediction horizon, including for $T = T_{design}$. Note also that picking the hold time (Δ_{k_r}) during the operation between $t_{k_r^{in}}$ and $t_{k_r^{out}}$ such that the time interval between switches be an integer multiple of the sampling time is made to ensure that the system does not go through a transition while the final control move in a given mode is being implemented. To see the reasoning behind this, note first that, due to the discrete nature of controller implementation,

the final control in the pre-switching mode is implemented for Δ_{k_r} time. Therefore, if the system undergoes a transition during that time (i.e., if $t + \Delta_{k_r}$ is greater than $t_{k_r^{out}}$), then the control could cause the closed-loop state to leave the stability region of the target mode after switching has been executed. Picking Δ_{k_r} as above precludes such a possibility.

Remark 7.9: The constraints of Eqs.7.19-7.20 require that the closed-loop state evolve so that the value of the Lyapunov function for the current mode continues to decay, and therefore the closed-loop state trajectory evolves in a way that it enters the stability region of the target mode (due to Eq.7.21), while, up-to the time of the switch, continuing to evolve in the stability region of the current mode. Owing to this, if at any time the transition constraints are not feasible, and the switching sequence is aborted, the Lyapunov-based predictive controller for the then-current mode is guaranteed to stabilize the closed-loop system. While this condition safeguards against instability, it may be restrictive, and hamper the feasibility of the optimization problem, especially if the boundaries of the stability regions of the constituent modes do not have a significant overlap. When dealing with a finite switching sequence, the constraints of Eqs.7.19-7.20 can be relaxed for all times before the final switch takes place and also a relaxed version of the constraint of Eq.7.21 may be used up to the time of the terminal switch, requiring only that $V_m(t_{m_j^{in}}) \leq c_m^{max}$ (i.e., the closed-loop state resides in the stability region of the target mode, see dashed lines in Fig.7.3).

Remark 7.10: For purely continuous systems, the problem of implementing various predictive control algorithms with guaranteed stability regions was recently addressed for linear systems under state [75] and output [145] feedback, nonlinear systems [74] and for nonlinear systems in the presence of uncertainty [147], by means of a hybrid control structure that uses bounded control as a fall-back controller in the event of infeasibility or instability of the predictive controller. In this chapter, the bounded controller design is not used as a fall back controller, but rather for the purpose of providing an estimate

of the stability region for the Lyapunov-based predictive controller, and feasible initial guesses for the control moves (the decision variables in the optimization problem). Note that the Lyapunov-based predictive controller of Eqs.7.11-7.15 is guaranteed to be feasible and stabilizing from an explicitly characterized set of initial conditions.

Remark 7.11: Note that the use of the predictive controller is both beneficial and essential to the problem of implementing a prescribed switching schedule using the proposed approach. In particular, while the bounded controller can achieve stabilization under single mode operation, the bounded controller framework simply does not address performance considerations, and more importantly does not allow for incorporating transition constraints, and there is no guarantee that the bounded controller (or even the Lyapunov-based predictive controller, if it does not incorporate the transition constraints) can stabilize the closed-loop system when following a prescribed switching schedule (see the simulation example for a demonstration). In contrast, the predictive controller approach provides a natural framework for specifying appropriate transitions constraints, which upon being feasible, ensure closed-loop stability.

Remark 7.12: For a nonlinear switched system of the form of Eq.7.1, while one can in principle set up the mixed integer nonlinear programming problem, where the decision variables (and hence the solution to the optimization problem) include the control action together with the switching schedule, it is not possible to use it for the purpose of online implementation. In many systems of practical interest, however, the switched system may be only required to follow a prescribed switching schedule. The predictive controller algorithm provides an implementable controller for switched nonlinear systems with a prescribed switching sequence by incorporating appropriate constraints on the implemented control action in each of the individual modes. Note also that if it is possible to solve the mixed integer optimization problem off-line, it can be used to provide a set of optimal switching times, and an associated switching sequence, which can be then implemented online using the proposed approach.

7.4 Application to a chemical process example

We consider a continuous stirred tank reactor where an irreversible, first-order exothermic reaction of the form $A \xrightarrow{k} B$ takes place. The operation schedule requires switching between two available inlet streams consisting of pure A at flow rates F_1, F_2 , concentrations C_{A1}, C_{A2} , and temperatures T_{A1}, T_{A2} , respectively. For each mode of operation, the mathematical model for the process takes the form

$$\begin{aligned}\dot{C}_A &= \frac{F_\sigma}{V}(C_{A\sigma} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A \\ \dot{T}_R &= \frac{F_\sigma}{V}(T_{A\sigma} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{Q_\sigma}{\rho c_p V}\end{aligned}\quad (7.22)$$

where C_A denotes the concentration of the species A , T_R denotes the temperature of the reactor, Q_σ is the heat removed from the reactor, V is the volume of the reactor, k_0 , E , ΔH are the pre-exponential constant, the activation energy, and the enthalpy of the reaction, c_p and ρ , are the heat capacity and fluid density in the reactor and $\sigma(t) \in \{1, 2\}$ is the discrete variable. The values of all process parameters can be found in Table 7.1. The control objective is to stabilize the reactor at the unstable equilibrium point $(C_A^s, T_R^s) = (0.57, 395.3)$ using the rate of heat input, Q_σ , and change in inlet concentration of species A , $\Delta C_{A\sigma} = C_{A\sigma} - C_{A\sigma_s}$ as manipulated inputs with constraints: $|Q_\sigma| \leq 1 \text{ KJ/hr}$ and $|\Delta C_{A\sigma}| \leq 1 \text{ mol/l}$, $\sigma = 1, 2$. For both the modes, we considered quadratic Lyapunov functions of the form $V(x) = x' P_k x$ where $x = (C_A - C_A^s, T_R - T_R^s)$, with $P_1 = \begin{bmatrix} 33.24 & 1.32 \\ 1.32 & 0.06 \end{bmatrix}$ and $P_2 = \begin{bmatrix} 7.39 & 0.32 \\ 0.32 & 0.016 \end{bmatrix}$ and used these in the Lyapunov-based controller design to compute the stability regions for the two modes, Ω_1 and Ω_2 , shown in Fig.7.4. The matrices P_k were computed by solving a Riccati inequality using the linearized system matrices. The computation of the stability region (using Eqs.7.4-7.5), however, was done using the nonlinear system dynamics. The parameters in the objective function of Eq.7.15 are chosen as $Q = qI$, with $q = 1$, and $R = rI$, with $r = 1.0$. The constrained nonlinear optimization problem is solved using the MATLAB subroutine `fmincon`, and the set of ODEs is integrated using the MATLAB solver `ODE45`.

Table 7.1: Process parameters and steady-state values for the reactor of Eq.7.22.

V	$=$	0.1	m^3
R	$=$	8.314	$kJ/kmol \cdot K$
$C_{A1,s}$	$=$	0.79	$kmol/m^3$
$C_{A2,s}$	$=$	1.0	$kmol/m^3$
T_{A1}	$=$	352.6	K
T_{A2}	$=$	310.0	K
$Q_{1,s}$	$=$	0.0	KJ/hr
$Q_{2,s}$	$=$	0.0	KJ/hr
ΔH	$=$	-4.78×10^4	$kJ/kmol$
k_0	$=$	1.2×10^9	s^{-1}
E	$=$	8.314×10^4	$kJ/kmol$
c_p	$=$	0.239	$kJ/kg \cdot K$
ρ	$=$	1000.0	kg/m^3
F_1	$=$	3.34×10^{-3}	m^3/s
F_2	$=$	1.67×10^{-3}	m^3/s
T_{Rs}	$=$	395.33	K
C_{As}	$=$	0.57	$kmol/m^3$

We first demonstrate the implementation of the Lyapunov-based predictive controller to a single mode operation of the chemical reactor, i.e., one in which the process is operated for all times in mode 1. To this end, we consider an initial condition that belongs to the stability region of the predictive controller for mode 1. As shown by the solid line in Fig.7.4 (see Fig.7.5 for the state and input profiles), starting from the initial condition $(C_A, T_R) = (0.14, 404.9)$, which belongs to the stability region of the predictive controller for mode 1, successful stabilization of the closed-loop system is achieved.

To demonstrate the need to account for the switched nature of the system, we choose a schedule involving a switch from inlet stream 1 (mode 1) to inlet stream 2 (mode 2) at time $t = 0.1 \text{ hr}$. Once again the system is initialized within the stability region of mode 1, and the predictive controller for mode 1 is implemented. Up until $t = 0.1 \text{ hr}$, the state of the closed-loop system moves towards the desired steady state (as seen from the dashed-lines in Fig.7.4); however, when the system switches to mode

2, the MPC controller, designed for the stabilization of the process in mode 2, does not yield a feasible solution. If the bounded controller for mode 2 is implemented, the resulting control action is not able to stabilize the closed-loop system (dashed lines in Figs.7.4–7.5). This happens because at the time of the transition, the state of the closed-loop system (marked by the o in Fig.7.4) does not belong to the stability region of mode 2. Note also that while the predictive formulation of Eqs.7.11-7.15 guarantees stabilization for all initial conditions belonging to the stability region of mode 1, it does not incorporate constraints which enable or ensure a safe transition to mode 2.

Finally, the predictive control algorithm of Theorem 7.1 (which incorporates constraints that account for switching) is implemented (dash-dotted lines in Figs.7.4–7.5). The MPC controller of mode 1 is designed to drive the state of the closed-loop system such that the state belongs to the stability region of mode 2 at the switching time. Consequently, when the system switches to mode 2 at $t = 0.1 \text{ hr}$, the closed-loop system state at the switching time (marked by the $\diamond$ in Fig.7.4) belongs to the stability region of the MPC designed for mode 2. At this time, when the process switches to mode 2 and the corresponding predictive controller is implemented, closed-loop stability is achieved.

7.5 Conclusions

In this chapter, a predictive control framework was proposed for the constrained stabilization of switched nonlinear systems that transit between their constituent modes at prescribed switching times. The main idea is to design a Lyapunov-based predictive controller for each of the constituent modes in which the switched system operates, and incorporate constraints in the predictive controller design which upon satisfaction ensure stability of the switched closed-loop system. This was achieved as follows: for each constituent mode, a Lyapunov-based model predictive controller (MPC) is designed, and an analytic bounded controller, using the same Lyapunov function is used to explicitly characterize the set of initial conditions for which the MPC, irrespective

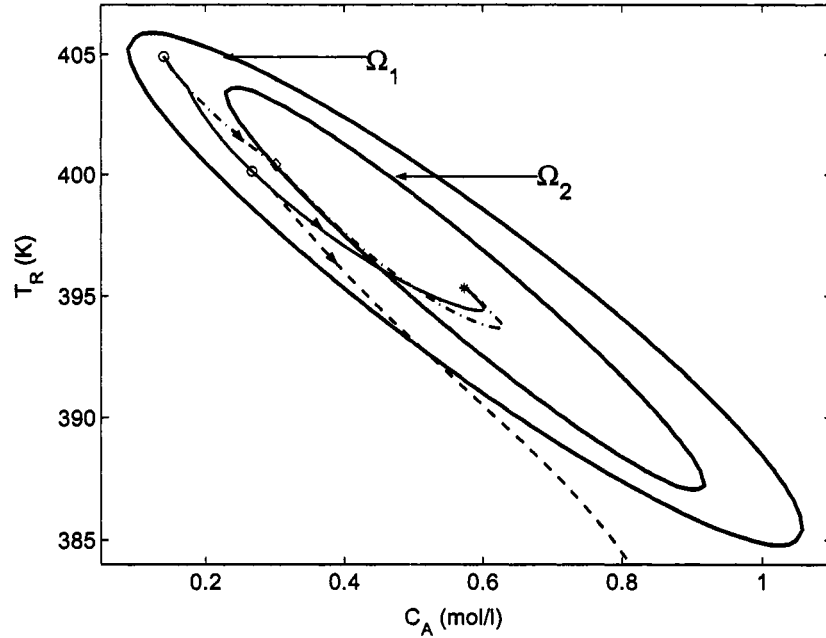


Figure 7.4: Closed-loop state trajectory when the reactor is operated in mode 1 for all times under the stabilizing MPC formulation of Eqs.7.11-7.15 (solid line), when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller design of Eqs.7.11-7.15 (dashed line), and when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller of Theorem 7.1 (dashed-dotted line).

of the controller parameters, is guaranteed to be feasible, and hence stabilizing. Then, constraints were incorporated in the MPC design which, upon satisfaction, ensure that: (1) the state of the closed-loop system, at the time of the transition, resides in the stability region of the mode that the system is switched into, and (2) the value of the Lyapunov function for each of the modes eventually decays to zero, thereby guaranteeing stability. The proposed control method was demonstrated through application to a chemical process example.

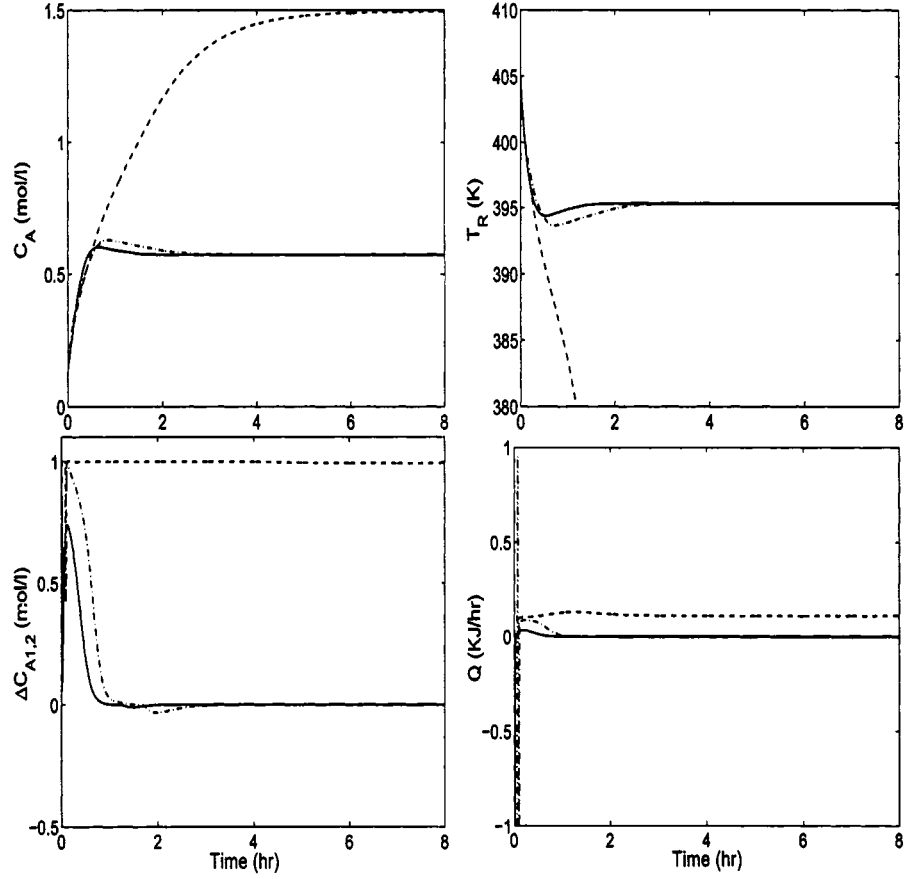


Figure 7.5: Closed-loop state (top plots) and manipulated input (bottom plots) profiles: when the reactor is operated in mode 1 for all times under the stabilizing MPC formulation of Eqs.7.11-7.15 (solid line), when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller design of Eqs.7.11-7.15 (dashed line), and when the reactor operation involves switching from mode 1 to mode 2 at $t = 0.1$ hr, under the predictive controller of Theorem 7.1 (dashed-dotted line).

Chapter 8

Integrated Fault-Detection and Fault-Tolerant Control of Process Systems

8.1 Introduction

Modern-day chemical plants involve a complex arrangement of processing units connected, in series and/or in parallel, and highly integrated with respect to material and energy flows through recycle streams and to information flow through tightly interacting control approaches. Increasingly faced with the requirements of safety, reliability and profitability, chemical plant operation is relying extensively on highly automated process control systems. Automation, however, tends to also increase vulnerability of the plant to faults (e.g., defects/malfunctions in process equipment, sensors and actuators, failures in the controllers or in the control loops) potentially causing a host of economic, environmental, and safety problems that can seriously degrade the operating efficiency of the plant if not addressed within a time appropriate to the context of the process dynamics. Examples include physical damage to the plant equipment, increase in the wasteful use of raw material and energy resources, increase in the downtime for

process operation resulting in significant production losses, and jeopardizing personnel and environmental safety. Management of abnormal situations is a challenge in the chemical industry since abnormal situations account annually for \$10 billion in lost revenue in the US alone [159]. These considerations provide a strong motivation for the development of methods and strategies for the design of advanced fault-tolerant control structures that ensure an efficient and timely response to enhance fault recovery, prevent faults from propagating or developing into total failures, and reduce the risk of safety hazards. Given the geographically-distributed, interconnected nature of the plant units and the large number of distributed sensors and actuators typically involved [239], the success of a fault-tolerant control method requires efficient fault detection, control designs that account for the complex nonlinear dynamics and constraints, and a high-level supervisor that coordinates the overall plant response to achieve fault-tolerant control.

Fault-tolerant control has been an active area of research for the past ten years, and has motivated many research studies in the context of aerospace engineering applications (see, e.g., [163, 245]) and is based on the underlying assumption of the availability of more control configurations than is required. Under this assumption, the reliable control approach dictates use of all the control loops at the same time so that failure of one control loop does not lead to the failure of the entire control structure (e.g., [237]). The use of only as many control loops as is required at a time, is often motivated by economic considerations (to save on unnecessary control action), and in this case, fault-tolerant control can be achieved through control-loop reconfiguration. Recently, fault-tolerant control has gained increasing attention in the context of chemical process control; however, the available results are mostly based on the assumption of a linear process description (e.g., [17, 233]), and do not account for complexities such as control constraints or the unavailability of state measurements.

In process control, given the complex dynamics of chemical processes (e.g., nonlin-

earities, uncertainties and constraints) the success of any fault-tolerant control method requires an integrated approach that brings together several essential elements, including: (1) the design of advanced feedback control algorithms that handle complex dynamics effectively, (2) the quick detection of process faults, and (3) the design of supervisory switching schemes that orchestrate the transition from the failed control configuration to available well-functioning fall-back configurations to ensure fault-tolerance. The realization of such an approach is increasingly aided by a confluence of recent, and ongoing, advances in several areas of process control research, including advances in nonlinear controller designs, advances in the analysis and control of hybrid process systems and advances in fault detection. In the remainder of this section, we will briefly review the state-of-the-art in these areas, as pertinent to the focus of this manuscript.

The highly nonlinear behavior of many chemical processes has motivated extensive research on nonlinear process control. Excellent reviews of results in the area of nonlinear process control can be found, for example, in [22, 238, 91]; for a more recent review, see [38]. The problems caused by input constraints have motivated numerous studies on the dynamics and control of systems subject to input constraints. Important contributions in this area include results on optimization-based control methods such as model predictive control (e.g., [84, 141, 76]), Lyapunov-based control (e.g., [127, 223, 106, 114, 66, 69]) and hybrid predictive control (e.g., [74, 147]).

The occurrence of faults in chemical processes and subsequent switching to fall-back control configurations naturally leads to the superposition of discrete events on the underlying continuous process dynamics thereby making a hybrid systems framework a natural setting for the analysis and design of fault-tolerant control structures. Proper coordination of the switching between multiple (or redundant) actuator/sensor configurations provides a means for fault-tolerant control. However, at this stage, despite the large and growing body of research work on a diverse array of hybrid system

problems (e.g., [90, 85, 94, 50, 19, 70]), the use of a hybrid system framework for the study of fault-tolerant control problems for nonlinear systems subject to constraints has received limited attention. In a previous work [71], a hybrid systems approach to fault-tolerant control was employed where, under the assumption of full state measurements and knowledge of the fault, stability region-based reconfiguration is implemented to achieve fault-tolerant control.

Existing results on the design of fault-detection filters include those that use past plant-data and those that use fundamental process models for the purpose of fault-detection filter design. Statistical and pattern recognition techniques for data analysis and interpretation (e.g., [116, 191, 160, 59, 157, 58, 47, 220, 5, 243]) use past plant-data to construct indicators that identify deviations from normal operation to detect faults. The problem of using fundamental process models for the purpose of detecting faults has been studied extensively in the context of linear systems [139, 78, 79, 144]; and more recently, some existential results in the context of nonlinear systems have been derived [198, 53].

In summary, a close examination of the existing literature indicates the lack of general and practical methods for the design of integrated fault-detection and fault-tolerant control structures for chemical plants accounting explicitly for actuator/controller failures, process nonlinearities and input constraints. Motivated by these considerations, we consider in this chapter the problem of implementing fault-tolerant control to nonlinear processes with input constraints subject to control actuator failures, and present and demonstrate an approach predicated upon the idea of integrating fault-detection, feedback and supervisory control. To illustrate the main idea behind the proposed approach, we first assume availability of measurements of all the process state variables. For the processes under consideration, a family of candidate control configurations, characterized by different manipulated inputs, is first identified. For each control configuration, a Lyapunov-based controller that enforces asymptotic closed-loop stability

in the presence of constraints, is designed, and the constrained stability region, associated with it, is explicitly characterized. A fault-detection filter is used to compute the expected closed-loop behavior in the absence of faults. Deviations of the process states from the expected closed-loop behavior are used to detect faults. A switching policy is then derived, on the basis of the stability regions, to orchestrate the activation/deactivation of the constituent control configurations in a way that guarantees closed-loop stability in the event that a failure is detected. Often, in chemical process applications, all state variables are not available for measurement. To deal with the problem of lack of process state measurements, a nonlinear observer is designed to generate estimates of the states, which are then used to implement the state feedback controller and the fault-detection filter. A switching policy is then derived to orchestrate the activation/deactivation of the constituent control configurations in a way that accounts for the estimation error. Finally, simulation studies are presented to demonstrate the implementation and evaluate the effectiveness of the proposed fault-tolerant control scheme.

8.2 Preliminaries

8.2.1 Process description

We consider a class of continuous-time, single-input nonlinear processes with constraints on the manipulated input, represented by the following state-space description:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g_{k(t)}(x(t))(u_{k(t)} + m_{k(t)}), & y_m &= h_m(x) \\ k(t) &\in \mathcal{K} = \{1, \dots, N\}, \quad N < \infty, & |u_{k(t)}| &\leq u_{max}^k \end{aligned} \quad (8.1)$$

where $x(t) \in \mathbb{R}^n$ denotes the vector of process state variables, $y_m \in \mathbb{R}$ denotes the measured variable, $u_k(t) \in [-u_{max}^k, u_{max}^k] \subset \mathbb{R}$ denotes the constrained manipulated input associated with the k -th control configuration and $m_{k(t)} \in \mathbb{R}$ denotes the fault in the k -th control configuration. For each value that k assumes in $\mathcal{K}$, the process is controlled via a different manipulated input which defines a given control configuration.

It is assumed that the origin is the equilibrium point of the nominal process (i.e.,

$f(0) = 0$, $g_k(x) \neq 0 \forall x \in \mathbb{R}^n$, and that the vector functions $f(\cdot)$ and $g_k(\cdot)$ are sufficiently smooth, for all k , on $\mathbb{R}^n$. Throughout the manuscript, a function $\beta(r, s)$ is said to belong to class $\mathcal{KL}$ if, for each fixed s , the mapping $\beta(\cdot, s)$ belongs to class $\mathcal{K}$ (a continuous function $\alpha(\cdot)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$) and for each fixed r , the mapping $\beta(r, \cdot)$ is decreasing, and $\beta(r, s) \rightarrow 0$ as $s \rightarrow \infty$; see also [112]. The notation $\|\cdot\|$ is used to denote the standard Euclidean norm of a vector, the notation $|\cdot|$ is used to denote the absolute value of a scalar and x' denotes the transpose of x and the notation $R = [r_1 \ r_2]$ is used to denote the augmented vector $R \in \mathbb{R}^{m+n}$ comprising of the vectors $r_1 \in \mathbb{R}^m$ and $r_2 \in \mathbb{R}^n$. The notation $L_f h$ denotes the standard Lie derivative of a scalar function $h(\cdot)$ with respect to the vector function $f(\cdot)$ and the notation $x(T^+)$ denotes the limit of the trajectory $x(t)$ as T is approached from the right, i.e., $x(T^+) = \lim_{t \rightarrow T^+} x(t)$. Throughout the manuscript, we assume that for any $|u_k| \leq u_{max}^k$ the solution of the system of Eq.8.1 exists and is continuous for all t .

8.2.2 Motivating example

To motivate our fault-tolerant control design methodology, we introduce in this subsection a benchmark chemical reactor example that will be used to illustrate the design and implementation of the fault-tolerant control structure. To this end, consider a well-mixed, non-isothermal continuous stirred tank reactor where three parallel irreversible elementary exothermic reactions of the form $A \xrightarrow{k_1} B$, $A \xrightarrow{k_2} U$ and $A \xrightarrow{k_3} R$ take place, where A is the reactant species, B is the desired product and U , R are undesired byproducts. Under standard modeling assumptions, a mathematical model of the process can be derived from material and energy balances and takes the following

form:

$$\begin{aligned}
\frac{dT}{dt} &= \frac{F}{V}(T_{A0} - T) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} k_{i0} \exp\left(\frac{-E_i}{RT}\right) C_A + \frac{Q}{\rho c_p V} \\
\frac{dC_A}{dt} &= \frac{F}{V}(C_{A0} - C_A) - \sum_{i=1}^3 k_{i0} \exp\left(\frac{-E_i}{RT}\right) C_A \\
\frac{dC_B}{dt} &= -\frac{F}{V}C_B + k_{10} \exp\left(\frac{-E_1}{RT}\right) C_A
\end{aligned} \tag{8.2}$$

where C_A and C_B denote the concentrations of the species A and B , T denotes the temperature of the reactor, Q denotes the rate of heat input/removal from the reactor, V denotes the volume of the reactor, ΔH_i , k_i , E_i , $i = 1, 2, 3$, denote the enthalpies, pre-exponential constants and activation energies of the three reactions, respectively, c_p and ρ denote the heat capacity and density of the reactor, respectively. The values of the process parameters and the corresponding steady-state values can be found in [71]. It was verified that under these conditions, the process of Eq.8.2 has three steady-states (two locally asymptotically stable and one unstable at $(T_s, C_{As}, C_{Bs}) = (388.57 \text{ K}, 3.59 \text{ kmol/m}^3, 0.41 \text{ kmol/m}^3)$). The control objective considered here is the one of stabilizing the reactor at the (open-loop) unstable steady-state. Operation at this point is typically sought to avoid high temperature, while simultaneously achieving reasonable conversion. To accomplish this objective in the presence of control system failures, we consider as manipulated inputs the rate of heat input, $u_1 = Q$, subject to the constraints $|Q| \leq u_{max}^1 = 748 \text{ KJ/s}$, the inlet stream temperature, $u_2 = T_{A0} - T_{A0s}$, subject to the constraints $|u_2| \leq u_{max}^2 = 100 \text{ K}$, with $T_{A0s} = 300 \text{ K}$ and the inlet reactant concentration, $u_3 = C_{A0} - C_{A0s}$, subject to the constraints $|u_3| \leq u_{max}^3 = 4 \text{ kmol/m}^3$, with $C_{A0s} = 4 \text{ kmol/m}^3$.

Each of the above manipulated inputs, together with measurements of reactor temperature and/or concentration, represents a unique control configuration (or control-loop) that, by itself, can stabilize the reactor. In the event of some failure in the primary configuration (involving the heat input, Q), the important questions that arise include how can the supervisor detect this fault (note that measurements of the manipulated

input variable are not available), and which control loop to activate once failure is detected in the active configuration. The answer to the first question involves the design of an appropriate fault-detection filter and is presented in section 8.3. The approach that we will utilize to answer the second question, i.e., that of deciding which backup controller should be activated in the event of a fault, will be based on the stability regions under the individual control configuration. To this end, we next review a state feedback control design that allows for characterizing the constrained stability region under each control configuration. Note that this particular choice of the controller is presented only as an example to illustrate our results, and that any other controller design that allows for an explicit characterization of the constrained stability region can be used instead.

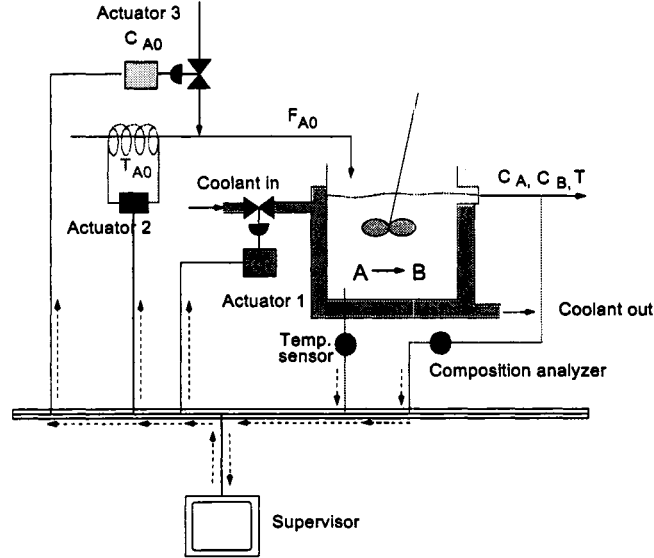


Figure 8.1: A schematic of the CSTR showing the three candidate control configurations.

8.2.3 Bounded Lyapunov-based control

Consider the system of Eq.8.1, for which a family of control Lyapunov functions (CLFs), $V_k(x)$, $k \in \mathcal{K} \equiv \{1, \dots, N\}$ has been found (see Remark 8.1 below for a discussion on the construction of CLFs). Using each control Lyapunov function, we construct, using

the results in [127] (see also [66]), the following continuous bounded control law:

$$u_k(x) = -\frac{L_f^* V_k(x) + \sqrt{\left(L_f^* V_k(x)\right)^2 + (u_{max}^k \|(L_{g_k} V_k)(x)\|)^4}}{\|(L_{g_k} V_k)(x)\|^2 \left[1 + \sqrt{1 + (u_{max}^k \|(L_{g_k} V_k)(x)\|)^2}\right]} (L_{g_k} V_k)(x) \quad (8.3)$$

when $(L_{g_k} V_k)(x) \neq 0$ and $u_k(x) = 0$ when $(L_{g_k} V_k)(x) = 0$, $L_f^* V_k(x) = \frac{\partial V_k(x)}{\partial x} f(x) + \rho_k V_k(x)$, $\rho_k > 0$ and $L_{g_k} V_k(x) = \frac{\partial V_k(x)}{\partial x} g_k(x)$. Let Π_k be the set defined by

$$\Pi_k(u_{max}^k) = \{x \in \mathbb{R}^n : L_f^* V_k(x) \leq u_{max}^k \|(L_{g_k} V_k)(x)\|\} \quad (8.4)$$

and assume that

$$\Omega_k := \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\} \subseteq \Pi_k(u_{max}^k) \quad (8.5)$$

for some $c_k^{max} > 0$. It can be shown, using standard Lyapunov arguments, that in the absence of faults ($m_{k(t)} = 0$), Ω_k provides an estimate of the stability region, starting from where the control law of Eq.8.3 guarantees asymptotic (and local exponential) stability of the origin of the closed-loop system under each control configuration. This implies that there exist class $\mathcal{KL}$ functions β_i , $i = 1, \dots, N$, such that $\|x(t)\| \leq \beta_i(\|x(0)\|, t)$. We will use this property later in the design of the output feedback controllers.

Remark 8.1: Referring to the above controller design, it is important to make the following remarks. First, a general procedure for the construction of CLFs for nonlinear systems of the form of Eq.8.1 is currently not available. Yet, for several classes of nonlinear systems that arise commonly in the modeling of engineering applications, it is possible to exploit system structure to construct CLFs (see, for example, [117, 80]). Second, given that a CLF, V_k , has been obtained for the system of Eq.8.1, it is important to clarify the essence and scope of the additional assumption that there exists a level set, Ω_k , of V_k that is contained in Π_k . Specifically, the assumption that the set, Π_k , contains an invariant subset around the origin, is necessary to guarantee the existence of a set of initial conditions for which closed-loop stability is guaranteed

(note that even though $\dot{V}_k < 0 \forall x \in \Pi_k \setminus \{0\}$, there is no guarantee that trajectories starting within Π_k remain within Π_k for all times). Moreover, the assumption that Ω_k is a level set of V_k is made only to simplify the construction of Ω_k . This assumption restricts the applicability of the proposed control method because a direct method for the construction of a CLF with level sets contained in Π_k is not available. However, the proposed control method remains applicable if the invariant set Ω_k is not a level set of V_k but can be constructed in some other way (which, in general, is a difficult task). Note also that possibly larger estimates of the stability region can be computed using constructive procedures such as Zubov's method [56] or by using a combination of several Lyapunov functions.

8.3 Integrated fault-detection and fault-tolerant control: state feedback case

8.3.1 State feedback fault-tolerant control

Consider the system of Eq.8.1, where all process states are available as measurements, i.e., $h_m(x) = x$, and without loss of generality, assume that it starts operating using control configuration i , under the controller of Eq.8.3. At some unknown time, T_i^f , a fault occurs in the first control configuration such that for all $t \geq T_i^f$, $m_i = -u_i$, i.e., control configuration i fails. The problems at hand are those of detecting that a fault has occurred and, upon detection, to decide which of the available backup configurations should be implemented in the closed-loop to achieve fault-tolerant control. To this end, we consider a fault-detection filter and a switching logic of the form:

$$\dot{w}(t) = f_f(w, x), \quad r(t) = h_f(w, x), \quad k(t) = \varphi(r, w, x) \quad (8.6)$$

where $w \in \mathbb{R}^n$ is the state of the filter, $r(t) \in \mathbb{R}$ is a residual that indicates the occurrence of a fault, and is the output of the filter, $f_f \in \mathbb{R}^n$ is the vector field describing the evolution of the filter state w , and $\varphi(r, w, x)$ is the switching logic that

dictates which of the available control configurations should be activated.

The main idea behind the fault-tolerant control design is as follows: (1) use the available state measurements, the process model, and the computed control action to simulate the evolution of the closed-loop process in the absence of actuator faults, compare it with the actual evolution of the states, and use the difference between the two behaviors, if any, to detect faults, and (2) having detected the fault, activate a backup control configuration for which the closed-loop state is within its stability region estimate. To formalize this idea, consider the constrained system of Eq.8.1 for which a bounded controller of the form of Eq.8.3 has been designed for each control configuration, and the stability region, Ω_j , $j = 1, \dots, N$ has been explicitly characterized. The fault-detection filter and the fault-tolerant control design are described in Theorem 8.1 below.

Theorem 8.1: *Let $k(0) = i$ for some $i \in \mathcal{K}$ and $x(0) := x_0 \in \Omega_i$. Set $w(0) = x(0)$, and consider the system*

$$\dot{w} = f(w) + g_i(w)u_i(w); \quad r = \|w - x\| \quad (8.7)$$

where $w \in \mathbb{R}^n$ is the filter state and $u_i(\cdot)$ is the feedback control law defined in Eq.8.3. Let T_i^f be such that $m_i(t) = 0 \forall 0 \leq t \leq T_i^f$, then $r(T_i^{f+}) > 0$ if and only if $m_i(T_i^f) \neq 0$. Furthermore, let T_i^s be the earliest time such that $r(t) > 0$, then the following switching rule:

$$k(t) = \left\{ \begin{array}{ll} i, & 0 \leq t < T_i^s \\ j \neq i, & t \geq T_i^s, x(T_i^s) \in \Omega_j \end{array} \right\} \quad (8.8)$$

guarantees asymptotic stability of the origin of the closed-loop system.

Proof of Theorem 8.1: We split the proof of the theorem in two parts. In the first part we show that the filter detects a fault if and only if one occurs, and in the second part we establish closed-loop stability under the switching rule of Eq.8.8.

Part 1: Let $x(T_i^f) := x_{T_i^f}$ and $w(T_i^f) := w_{T_i^f}$ and consider

$$\dot{w}(T_i^f) - \dot{x}(T_i^f) = f(x_{T_i^f}) + g(x_{T_i^f})(u_i(x_{T_i^f}) + m_i(T_i^f)) - (f(w_{T_i^f}) + g(w_{T_i^f})u_i(w_{T_i^f})) \quad (8.9)$$

with $m_i(T_i^f) \neq 0$. Since $w_{T_i^f} = x_{T_i^f}$, we have that

$$f(x_{T_i^f}) + g(x_{T_i^f})(u_i(x_{T_i^f}) + m_i(T_i^f)) - (f(w_{T_i^f}) + g(w_{T_i^f})u_i(w_{T_i^f})) = g(x_{T_i^f})m_i(T_i^f) \quad (8.10)$$

Furthermore, since $g(x_{T_i^f}) \neq 0$, we have that

$$\dot{w}(T_i^f) - \dot{x}(T_i^f) = g(x_{T_i^f})m_i(T_i^f) \neq 0 \quad (8.11)$$

if and only if $m_i(T_i^f) \neq 0$. Since $w_{T_i^f} - x_{T_i^f} = 0$ and $\dot{w}(T_i^f) - \dot{x}(T_i^f) \neq 0$ if and only if $m_i(T_i^f) \neq 0$, we have that

$$w(T_i^{f+}) - x(T_i^{f+}) \neq 0 \quad (8.12)$$

or

$$r(T_i^{f+}) = \|w(T_i^{f+}) - x(T_i^{f+})\| > 0 \quad (8.13)$$

if and only if $m_i(T_i^f) \neq 0$.

Part 2: We prove closed-loop stability for the two possible cases; first if no switching occurs, and second if a switch occurs at a time T_i^s .

Case 1: The absence of a switch implies $r_i(t) = 0$. Furthermore, $r_i(t) = 0 \implies x(t) = w(t)$. Since $x(0) = w(0) \in \Omega_i$, and control configuration i is implemented for all times in this case, we have that asymptotic closed-loop stability is achieved.

Case 2: At time T_i^s , the supervisor switches to a control configuration j for which $x(T_i^s) \in \Omega_j$. From this time onwards, since configuration j is implemented in the closed-loop system for all times, and since $x(T_i^s) \in \Omega_j$, closed-loop stability follows. This completes the proof of Theorem 8.1.

Remark 8.2: The fault-detection filter and fault-tolerant controller are designed and implemented as follows (see also Fig.8.2):

- Given any $x_0 \in \Omega_i$, initialize the filter states as $w(0) = x_0$ and integrate the filter dynamics using Eq.8.7.
- Compute the norm of the difference between the filter states and the process

states, $r(t) = \|w(t) - x(t)\|$ and if $r(t) = 0$, continue to implement control configuration i .

- At any time T_i^s that $r(T_i^s) > 0$, switch to a control configuration $j \neq i$, for which $x(T_i^s) \in \Omega_j$ to achieve asymptotic stability of the origin of the closed-loop system.

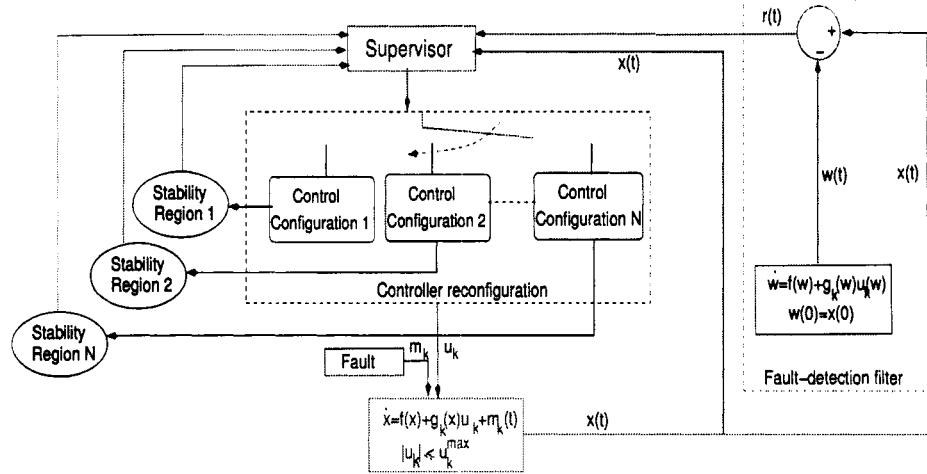


Figure 8.2: Integrated fault-detection and fault-tolerant control design: state feedback case.

Remark 8.3: Note that the fault-detection filter uses a replica of the process dynamics, and that the state of the filter w is initialized at the same value as the process states $x(0)$. In the absence of faults, the evolution of $w(t)$ is identical to $x(t)$, and hence $r(t) = 0$. In the presence of faults, however, the effect of the fault is registered by a change in the evolution of the process, but not in that of the filter state (since the filter state dynamics include the computed control action, $u_i(w)$, and not the implemented control action, $u_i(w) + m_i$). This change is detected by a change in the value of $r(t)$ and declared as a fault. Note also, that the fact that the faults m_i appear as additive terms to the manipulated input variable is a natural consequence of focussing on the problem of detecting (through the design of appropriate fault-detection filters) and dealing (via reconfiguration) with faults in control actuators. The approach employed in the design of the fault-detection filter can also be used to detect faults that do not necessarily appear in the control actuators, as long as they influence the evolution of

the state variables. Note that the nonlinear model-based filter used in this work is just one example of a filter design that can be used within the proposed approach. Other data-based fault-detection filters that detect a fault on the basis of monitoring the evolution of the process states in comparison with past data (see, for example, [191, 192, 6, 5]) can also be used within the proposed approach.

Remark 8.4: Once a fault is detected, the switching logic ensures that the backup control configuration that is implemented in the closed-loop is one that can guarantee closed-loop stability in the presence of constraints, and this is achieved by verifying that the state of the process, at the time that a fault is detected, is present in the constrained stability region of the candidate control configuration. Early detection of a fault enhances the chances that corrective action can be taken in time to achieve fault-tolerant control (Theorem 8.1 guarantees that a fault is detected as soon as it occurs). Specifically, it may happen that a fault occurs when the closed-loop state resides in the stability region of one of the backup configurations, but if the fault is not immediately detected, the destabilizing effect of the fault may drive the state outside the stability region of the backup configuration by the time a fault is detected (for a demonstration, see subsection 8.4.3).

Remark 8.5: In the event that the process state, at the time of the failure of the primary control configuration, lies in the stability region of more than one backup control configuration, additional performance considerations such as ease and/or cost of implementing one control configuration over another, can be used in choosing which control configuration should be implemented in the closed-loop system. If the state at the time of a failure lies outside the stability region of all the backup controllers, then this indicates that the back up controllers do not have enough control action available and calls for increasing the allowable control action in the fall-back configurations. Note that the set of initial conditions starting from where a given control configuration can stabilize a steady state – the so-called null-controllable region – is fundamentally

limited by the constraints on the available control action, and that different control laws typically provide estimates of the stability region which are subsets of the null-controllable region.

Remark 8.6: In the presence of plant model mismatch or unknown disturbances, the value of $r(t)$ will be nonzero even in the absence of faults. The FDFTC problem in the presence of time varying disturbances with known bounds on the disturbances can be handled by (1) redesigning the filter to account for the disturbances; specifically, requiring that a fault be declared only if the value of $r(t)$ increases beyond some threshold, δ , where δ accounts for the deviation of the plant dynamics from the nominal dynamics in the absence of faults and (2) by redesigning the controllers for the individual control configurations to mitigate the effect of disturbances on the process, and characterizing the robust stability regions and using them as criteria for deciding which backup controller should be implemented in the closed-loop. Note that while Theorem 8.1 presents the fault-detection filter and fault-tolerant control (FDFTC) design for a fault in the primary control configuration, extensions to faults in successive backup configurations are straightforward and involve similar filter designs for the active control configuration and a switching logic that orchestrates switching to the remaining control configurations.

Remark 8.7: While we illustrate our idea using a single input, extensions to multi-input systems are possible, and fault-detection filters can be designed in the same way, using a replica of the process dynamics. The case of multi-input systems, however, introduces an additional layer of complexity due to the need of identifying which particular manipulated input has failed, i.e., the additional problem of fault-isolation. For the purpose of presenting the integrated fault-detection and fault-tolerant control structure, we focus here on multiple control configurations, where each control configuration comprises of a single input that does not require the filter to perform the additional task of fault-isolation.

Remark 8.8: Note that the fault-detection filter presented in Theorem 8.1 detects the presence of both complete and partial failures. Once a fault is detected, the control reconfiguration strategy is the same for both cases, and that is to shut down the faulty configuration and switch to some well-functioning fall-back configuration. Note that in the case of a partial failure, unless the faulty configuration is shut down, the backup control configurations will have to be redesigned to be robust with respect to the bounded disturbance generated by the faulty configuration (for the backup control configuration, the unmeasured actuator action of the faulty control configuration will act as a disturbance and will be bounded because of the fact that the actuator itself has a limited capacity and, therefore, even if the implemented control action is not the same as that prescribed by the controller, it cannot exceed the physical limitations and will remain bounded). By shutting down the faulty configuration, however, the source of the disturbance is eliminated and no controller redesign is needed for the backup control configurations.

8.3.2 Simulation results

In this subsection, we illustrate the implementation of the proposed fault-detection/fault-tolerant control methodology to the chemical reactor example introduced in subsection 8.2.2. We first describe the controller design for the individual control configurations. Note that our objective is full state stabilization; however, to facilitate the controller design and subsequent stability analysis, we use a state transformation to transform the system of Eq.8.2.2 into the following one describing the input/output dynamics:

$$\dot{e} = Ae + l_k(e) + b\alpha_k u_k \quad := \quad \bar{f}_k(e) + \bar{g}_k(e)u_k \quad (8.14)$$

where $e \in \mathbb{R}^n$ is the variable in transformed co-ordinate (for the specific transformations used for each control configuration, please see below) $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $l_k(\cdot) = L_{f_k}^2 h_k(x)$, $\alpha_k(\cdot) = L_{g_k} L_{f_k} h_k(x)$, $h_k(x) = y_k$ is the output associated with the k -th configuration, $x = [x_1 \ x_2]^T$ with $x_1 = T - T_s$, $x_2 = C_A - C_{As}$, and the functions

$f_k(\cdot)$ and $g_k(\cdot)$ can be obtained by re-writing the (T, C_A) model equations in Eq.8.2 in the form of Eq.8.1. The explicit forms of these functions are omitted for brevity. A quadratic Lyapunov function of the form $V_k = e^T P_k e$, where P_k is a positive-definite symmetric matrix that satisfies the Riccati inequality $A^T P_k + P_k A - P_k b b^T P_k < 0$, is used for controller design. In particular:

1. For the first configuration with $u_1 = Q$, we consider the controlled output $y_1 = C_A - C_{As}$. The coordinate transformation (in error variables form) takes the form: $e_1 = C_A - C_{As}$, $e_2 = \frac{F}{V}(C_{A0} - C_A) - \sum_{i=1}^3 k_{i0} e^{\frac{-E_i}{RT}} C_A$ and yields a relative degree of two with respect to the manipulated input.
2. For the second configuration with $u_2 = T_{A0} - T_{A0s}$, we choose the output $y_2 = C_A - C_{As}$ which yields the same relative degree as in the first configuration, $r_2 = 2$, and the same coordinate transformation.
3. For the third configuration with $u_3 = C_{A0} - C_{A0s}$, a coordinate transformation of the form used for configurations 1 and 2 above does not yield a sufficiently large estimate of the stability region, we therefore choose a candidate Lyapunov function of the form $V_3(x) = x' P x$, where $P > 0$ and $x = [T - T_s \quad C_A - C_{As}]'$ with $P = \begin{bmatrix} 0.011 & 0.019 \\ 0.019 & 0.101 \end{bmatrix}$.

Fig.8.3 depicts the stability region, in the (T, C_A) space, for each configuration. The desired steady-state is depicted with an asterisk that lies in the intersection of the three stability regions. The reactor as well as the fault-detection filter for the first control configuration is initialized at $T(0) = 330 \text{ K}$, $C_A(0) = 3.6 \text{ kmol/m}^3$, $C_B(0) = 0.0 \text{ kmol/m}^3$, using the Q -control configuration, and the supervisor proceeds to monitor the evolution of the closed-loop trajectory.

As shown by the solid lines in Figs.8.3-8.4, the controller proceeds to drive the closed-loop trajectory towards the desired steady-state, up until the Q -configuration fails after 3 minutes of reactor startup (see Fig.8.6a). As can be seen in Fig.8.5a,

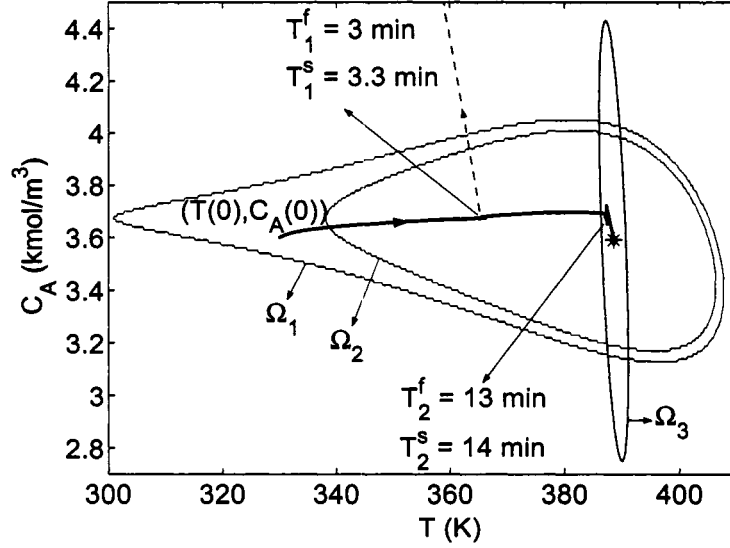


Figure 8.3: Evolution of the closed-loop state profiles under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid line) and under arbitrary switching (dashed line).

at this time the value of $r_1(t)$ becomes non-zero and the fault-detection filter detects this fault. If the supervisor switches arbitrarily, and in particular, switches to backup configuration 3, closed-loop stability is not achieved (dashed lines in Figs.8.3-8.4). Note that this happens because the closed-loop state is outside the stability region of the third control configuration, and even though the third control configuration does not encounter a fault ($r_3(t) = 0$; see dashed line in Fig.8.5b), the limited control action available in this configuration is unable to achieve closed-loop stability. On the basis of the switching logic of Eq.8.8, the supervisor activates the second configuration (with T_{A0} as the manipulated input, see Fig.8.6b), which continues to drive the state trajectory closer to the desired steady-state.

To demonstrate the implementation of the proposed FDFTC strategy when faults occur in successive control configurations, we consider the case when a second failure occurs (this time in the T_{A0} -configuration) at $t = 13$ minutes. Once again, the filter detects this failure via an increase in the value of $r_2(t)$ (solid line in Fig.8.5b) using

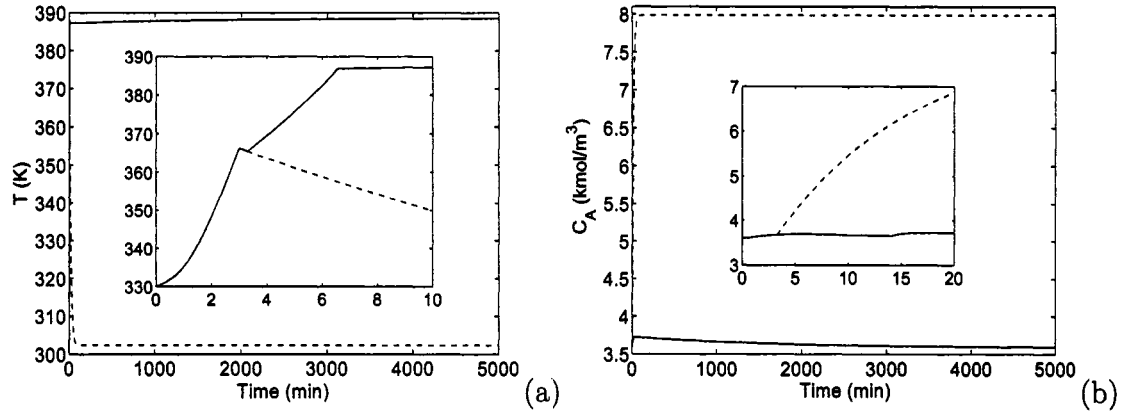


Figure 8.4: Evolution of the closed-loop (a) temperature and (b) concentration under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).

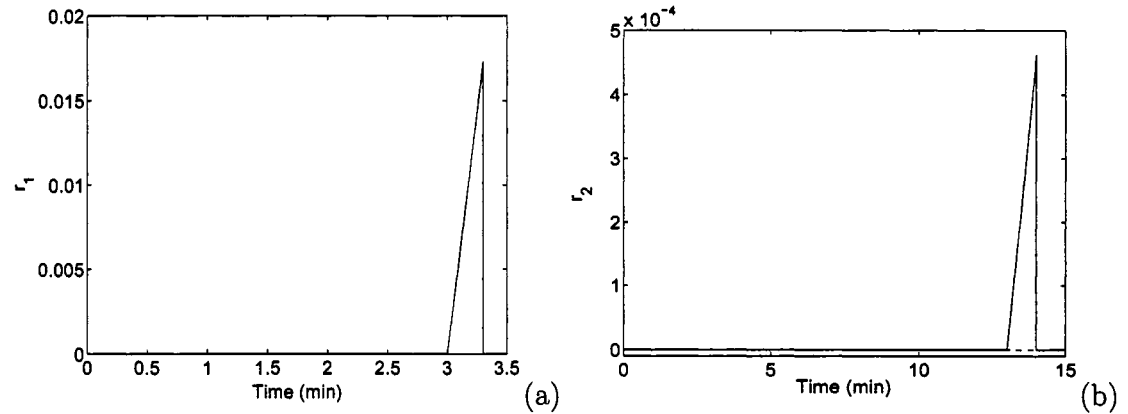


Figure 8.5: Evolution of the closed-loop residual under the fault-detection filter for (a) control configuration 1 and (b) control configurations 2 and 3 under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).

the fault-detection filter for control configuration 2. From Fig.8.3, it is clear that the failure of the second control configuration occurs when the closed-loop trajectory is within the stability region of the third configuration. Therefore, the supervisor immediately activates the third control configuration (with C_{A0} as the manipulated input, see Fig.8.6c) which finally stabilizes the reactor at the desired steady-state.

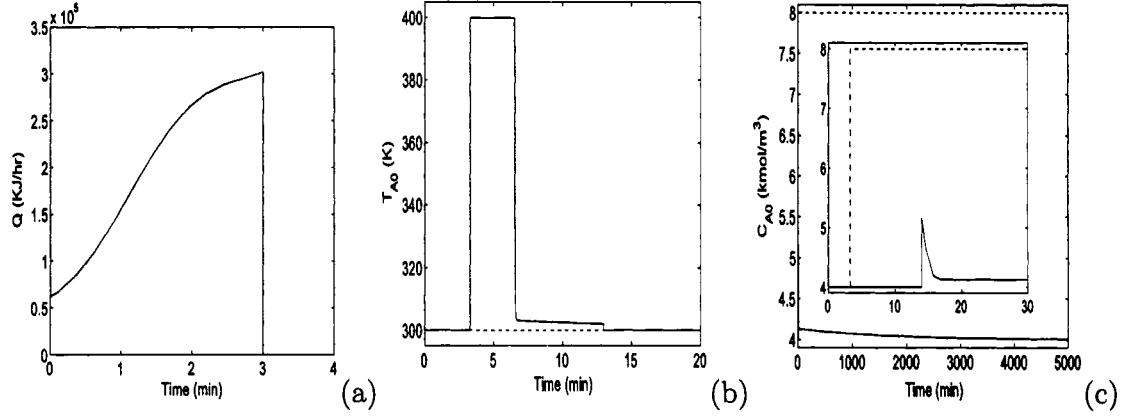


Figure 8.6: Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Eq.8.8 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).

8.4 Integrated fault-detection and fault-tolerant control: output feedback case

The feedback controllers, the fault-detection filters and the switching rules in the previous section were designed under the assumption of availability of measurements of all the process states. The unavailability of full state measurements has several implications. First, it necessitates generating estimates of the states to be used in conjunction with both the state feedback controller and the fault-detection filter. The state estimates, however, contain errors, and this results in a difference between the expected closed-loop behavior of the measured variables (computed using the state estimates) and the evolution of the measured variables, even in the absence of actuator faults. The fault-detection filter has to be redesigned to account for this fact so that it does

not treat this difference to be an indicator of an actuator fault (i.e., to prevent a false alarm). Also, the switching logic has to account for the fact that the supervisor can monitor only the state estimates and needs to make inferences about the true values of the states using the state estimates.

In the remainder of this section, we first review an output feedback controller design, proposed in [69], based on a combination of a high-gain observer and a state feedback controller (see also [133, 108, 110, 213, 39] for results on observer designs and output feedback control for unconstrained nonlinear systems) and characterize the stability properties of the closed-loop system under output feedback control. Then, we present the fault-detection filter and fault-tolerant controller and demonstrate its application via a simulation example.

8.4.1 Output feedback control

To facilitate the design of a state estimator with the required convergence properties, we make the following assumption:

Assumption 1: *For each $i \in \mathcal{K}$, there exists a set of coordinates*

$$\begin{bmatrix} \xi_i \end{bmatrix} = \begin{bmatrix} \xi_i^1 \\ \xi_i^2 \\ \vdots \\ \xi_i^n \end{bmatrix} = \chi_i(x) = \begin{bmatrix} h_m(x) \\ L_f h_m(x) \\ \vdots \\ L_f^{n-1} h_m(x) \end{bmatrix} \quad (8.15)$$

such that the system of Eq. 8.1 takes the form

$$\begin{aligned} \dot{\xi}_i^1 &= \xi_i^2 \\ &\vdots \\ \dot{\xi}_i^{n-1} &= \xi_i^n \\ \dot{\xi}_i^n &= L_f^n h_m(\chi_i^{-1}(\xi)) + L_{g_i} L_f^{n-1} h_m(\chi_i^{-1}(\xi))(u_{i(t)} + m_{i(t)}) \end{aligned} \quad (8.16)$$

where $L_{g_i} L_f^{n-1} h_m(x) \neq 0$ for all $x \in \mathbb{R}^n$. Also, $\xi_i \rightarrow 0$ if and only if $x \rightarrow 0$.

We note that the change of variables is invertible, since for every x , the variable ξ_i is uniquely determined by the transformation $\xi_i = \chi_i(x)$. This implies that if one

can estimate the values of ξ_i for all times, using an appropriate state observer, then we automatically obtain estimates of x for all times, which can be used to implement the state feedback controller. The existence of such a transformation will facilitate the design of high-gain observers which will be instrumental in preserving the same closed-loop stability properties achieved under full state feedback.

Proposition 8.1 below presents the output feedback controller used for each mode and characterizes its stability properties. The proof of the proposition, which invokes singular perturbation arguments, is a special case of the proof of Theorem 8.2 in [69], and is omitted for brevity. To simplify the statement of the proposition, we first introduce the following notation. We define $\alpha_i(\cdot)$ as a class $\mathcal{K}$ function that satisfies $\alpha_i(\|x\|) \leq V_i(x)$. We also define the set $\Omega_{b,i} := \{x \in \mathbb{R}^n : V_i(x) \leq \delta_{b,i}\}$, where $\delta_{b,i}$ is chosen such that $\beta_i(\alpha_i^{-1}(\delta_{b,i}), 0) < \alpha_i^{-1}(c_i^{max})$, where $\beta_i(\cdot, \cdot)$ is a class $\mathcal{KL}$ function and c_i^{max} is a positive real number defined in Eq.8.5.

Proposition 8.1: *Consider the nonlinear system of Eq.8.1, for a fixed mode, $k(t) = i$, and with $m_i(t) \equiv 0$, under the output feedback controller:*

$$\begin{aligned} \dot{\tilde{y}} &= \begin{bmatrix} -L_i a_1^{(i)} & 1 & 0 & \cdots & 0 \\ -L_i^2 a_2^{(i)} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -L_i^n a_n^{(i)} & 0 & 0 & \cdots & 0 \end{bmatrix} \tilde{y} + \begin{bmatrix} L_i a_1^{(i)} \\ L_i^2 a_2^{(i)} \\ \vdots \\ L_i^n a_n^{(i)} \end{bmatrix} y_m \\ u_i &= u_i^c(\hat{x}, u_i^{max}) \end{aligned} \quad (8.17)$$

where u_i^c is defined in Eq.8.3, the parameters, $a_1^{(i)}, \dots, a_n^{(i)}$ are chosen such that the polynomial $s^n + a_1^{(i)}s^{n-1} + a_2^{(i)}s^{n-2} + \cdots + a_n^{(i)} = 0$ is Hurwitz, $\hat{x} = \chi_i^{-1}(\text{sat}(\tilde{y}))$, $\text{sat}(\cdot) = \min\{1, \zeta_{max,i}/|\cdot|\}(\cdot)$, with $\zeta_{max,i} = \beta_\zeta(\delta_{\zeta,i}, 0)$ where β_ζ is a class $\mathcal{KL}$ function and $\delta_{\zeta,i}$ is the maximum value of the norm of the vector $[h_m(x) \cdots L_{f_i}^{n-1} h_m(x)]$ for $V_i(x) \leq c_i^{max}$ and let $\epsilon_i = 1/L_i$. Then, given $\Omega_{b,i}$, there exists $\epsilon_i^* > 0$ such that if $\epsilon_i \in (0, \epsilon_i^*]$, $x(0) \in \Omega_{b,i}$, and $\|\tilde{y}(0)\| \leq \delta_{\zeta,i}$, the origin of the closed-loop system is asymptotically (and locally exponentially) stable. Furthermore, given any positive real

numbers, $e_{m,i}$ and T_i^b , there exists a real positive number ϵ_i^{**} such that if $\epsilon_i \in (0, \epsilon_i^{**}]$ then $\|x(t) - \hat{x}(t)\| \leq e_{m,i}$ for all $t \geq T_i^b$.

Remark 8.9: The state observer in Eq.8.17 ensures sufficiently fast convergence that is necessary for the implementation of both the state feedback controller (and preserving its stability properties under output feedback control), and the fault-detection filter. The most important feature of this estimator (and one that will be used in the fault-detection filter design) is that the estimation error is guaranteed to fall below a certain value in a small period of time, T_i^b , which can be chosen arbitrarily small by sufficiently increasing the observer gain. This requirement or constraint on the error dynamics is needed even when other estimation schemes, such as moving horizon observers, are used (for example, see [150, 178]). For such observers, however, it is difficult in general to obtain a transparent relationship between the tunable observer parameters and the error decay rate.

Remark 8.10: Due to the lack of full state measurements, the supervisor can rely only on the available state estimates to decide whether switching at any given time is permissible, and, therefore, needs to make reliable inferences regarding the position of the states based upon the available state estimates. Proposition 8.2 below establishes the existence of a set, $\Omega_{s,i} := \{x \in \mathbb{R}^n : V_i(x) \leq \delta_{s,i}\}$, such that once the state estimation error has fallen below a certain value (note that the decay rate can be controlled by adjusting L_i), the presence of the state within the output feedback stability region, $\Omega_{b,i}$, can be guaranteed by verifying the presence of the state estimates in the set $\Omega_{s,i}$. A similar approach was employed in the construction of the output feedback stability regions $\Omega_{b,i}$ and the regions for the state estimates $\Omega_{s,i}$ in the context of output feedback control of linear systems in [145].

Proposition 8.2: *Given any positive real number $\delta_{b,i}$, there exist positive real numbers $e_{m,i}^*$ and $\delta_{s,i}$ such that if $\|x - \hat{x}\| \leq e_{m,i}$, where $e_{m,i} \in (0, e_{m,i}^*]$, and $V_i(\hat{x}) \leq \delta_{s,i}$, then $V_i(x) \leq \delta_{b,i}$.* **Proof of Proposition 8.2:** From the continuity of the function $V_i(\cdot)$,

we have that for any positive real number $e_{m,i}$, there exists a positive real number γ_i such that $\|x - \hat{x}\| \leq e_{m,i} \implies |V_i(x) - V_i(\hat{x})| \leq \gamma_i \implies V_i(x) \leq V_i(\hat{x}) + \gamma_i$. Since γ_i can be made small by choosing $e_{m,i}$ small, it follows that given any positive real number $\delta_{b,i}$, there exists a positive real number, $e_{m,i}^*$, such that for all $e_{m,i} \in (0, e_{m,i}^*]$, $\gamma_i < \delta_{b,i}$. Now, let $\delta_{s,i}$ be any positive real number that satisfies $\delta_{s,i} + \gamma_i \leq \delta_{b,i}$. Then if $\|x - \hat{x}\| \leq e_{m,i} \leq e_{m,i}^*$ and $V_i(\hat{x}) \leq \delta_{s,i}$, we have $V_i(x) \leq V_i(\hat{x}) + \gamma_i \leq \delta_{s,i} + \gamma_i \leq \delta_{b,i}$. This completes the proof of the proposition.

Remark 8.11: Note that for the inference that $\hat{x} \in \Omega_{s,i} \implies x \in \Omega_{b,i}$ to be useful in executing the switching, the set $\Omega_{s,i}$ needs to be contained within $\Omega_{b,i}$. From Proposition 8.2, this can be ensured if $e_{m,i}$ is sufficiently small, which in turn is ensured for all times greater than T_i^b provided that the observer gain is sufficiently large. In practice, use of a sufficiently high observer gain leads to an $\Omega_{b,i}$ that is almost identical to Ω_i , and furthermore, once the error has sufficiently decreased, $\Omega_{s,i}$ can be taken to be almost equal to $\Omega_{b,i}$.

8.4.2 Integrating fault-detection and fault-tolerant output feedback control

In this subsection we will present a fault-tolerant controller that uses the estimates generated by the high-gain observer for the implementation of the fault-detection filter, the state feedback controllers and the switching logic (see Fig.8.7). We proceed by first showing how the implementation of the design and implementation of the fault-detection filter should be modified to handle the absence of full state measurements. To this end, we consider the following system:

$$\begin{aligned} \dot{w}(t) &= f(w) + g_i(w)u_i(w) \\ r(t) &= \|\hat{x}(t) - w(t)\| \end{aligned} \tag{8.18}$$

Note that, as in the full state feedback case, the state equation for the filter in Eq.8.18 is a replica of the closed-loop state equation under full state feedback and in the absence of faults. However, because of the absence of full state measurements, the residual can

only be defined in terms of the state estimates, not the actual states. The residual therefore provides a measure of the discrepancy between the evolution of the nominal closed-loop system (i.e., with no faults) under full state feedback and the evolution of the closed-loop state estimates under output feedback. Since the discrepancy can be solely due to estimation errors and not necessarily due to faults, it is important to establish a bound on the residual which captures the expected difference in behavior in the absence of faults. This bound, which is given in Proposition 8.3 below, will be useful as a threshold to be used by the supervisor in declaring when a fault has occurred and consequently when switching becomes necessary.

Proposition 8.3: *Consider the nonlinear system of Eq.8.1, for a fixed mode, $k(t) = i$, and with $m_i(t) \equiv 0$, under the output feedback controller of Eq.8.17. Consider also the system of Eq.8.18. Then, given the set of positive real numbers $\{\delta_{b,i}, \delta_{\zeta,i}, \delta_{m,i}, T_i^b\}$, there exists a positive real number, $\epsilon'_i > 0$, such that if $\epsilon_i \in (0, \epsilon'_i]$, $V_i(x(0)) \leq \delta_{b,i}$, $\|\tilde{y}(0)\| \leq \delta_{\zeta,i}$, $w(T_i^b) = \hat{x}(T_i^b)$, the residual satisfies a relation of the form $r(t) \leq \delta_{m,i}$ for all $t \geq T_i^b$.*

Proof of Proposition 8.3: Consider the system of Eq.8.1 with $m_i(t) \equiv 0$ under the output feedback controller of Eq.8.17. From the result of Proposition 8.1, we have that given $x(0) \in \Omega_{b,i}$ and any positive real number T_i^b , there exists a real positive number ϵ_i^{**} such that $\|x(t) - \hat{x}(t)\| \leq k_1 \epsilon_i$, for all $t \geq T_i^b$, $\epsilon_i \in (0, \epsilon_i^{**}]$, for some $k_1 > 0$, i.e., $x(t) = \hat{x}(t) + O(\epsilon_i)$, where $O(\epsilon_i)$ is the standard order of magnitude notation. Now, consider the following two systems for $t \geq T_i^b$:

$$\dot{x}(t) = f(x(t)) + g_i(x(t))u_i(\hat{x}(t)) \quad (8.19)$$

$$\dot{w}(t) = f(w(t)) + g_i(w(t))u_i(w(t)) \quad (8.20)$$

where $w(T_i^b) = \hat{x}(T_i^b)$. The system of Eq.8.20 is exactly the closed-loop system under full state feedback and has an asymptotically (and exponentially) stable equilibrium at the origin, for all initial conditions within Ω_i . The system of Eq.8.19 is the closed-loop system under output feedback and (from Proposition 8.1) has an asymptotically (and

locally exponentially) stable equilibrium at the origin, for all initial conditions within $\Omega_{b,i} \subset \Omega_i$ and for all $\epsilon_i \leq \epsilon_i^*$. Since $x(t) = \hat{x}(t) + O(\epsilon_i)$ for all $t \geq T_i^b$, we have that $x(T_i^b) = \hat{x}(T_i^b) + O(\epsilon_i)$ and, when $\epsilon_i = 0$, the two systems of Eqs.8.19-8.20 become identical. Let $F_i(\cdot) = f(\cdot) + g_i(\cdot)u_i(\cdot)$, and $x(T_i^b) = \hat{x}(T_i^b) + O(\epsilon_i) := \eta(\epsilon_i)$, where η is a continuous function that depends smoothly on ϵ_i , then we can write

$$\begin{aligned}\dot{x}(t) &= F_i(x(t), \epsilon_i), \quad x(T_i^b) = \eta(\epsilon_i) \\ \dot{w}(t) &= F_i(w(t)), \quad w(T_i^b) = \eta(0)\end{aligned}\tag{8.21}$$

It is clear from the above representation that the state equations for both the filter system and the closed-loop system, as well as their initial conditions at T_i^b , are identical when $\epsilon_i = 0$. Therefore, we can use the theory of regular perturbations (see Chapter 8 in [112]) to establish the closeness of solutions between the two systems over the infinite time interval. In particular, since $F_i(\cdot)$ is continuous and bounded on $\Omega_{b,i}$, and the w -system is exponentially stable, an application of the result of Theorem 8.2 in [112] yields that there exists $\epsilon_i'' > 0$ such that for all $\epsilon_i \in (0, \epsilon_i'']$, $x(t) = w(t) + O(\epsilon_i)$ for all $t \geq T_i^b$. We therefore have that, for $\epsilon_i \in (0, \min\{\epsilon_i^{**}, \epsilon_i''\}]$, $r(t) = \|\hat{x}(t) - w(t)\| = \|\hat{x}(t) - x(t) + x(t) - w(t)\| \leq \|\hat{x}(t) - x(t)\| + \|x(t) - w(t)\| \leq (k_1 + k_2)\epsilon_i$ for all $t \geq T_i^b$. This implies that given any positive real number $\delta_{m,i}$, there exists $\epsilon_i' > 0$ such that $\|\hat{x}(t) - w(t)\| \leq \delta_{m,i}$ for all $\epsilon_i \in (0, \epsilon_i']$, for all $t \geq T_i^b$, where $\epsilon_i' = \min\{\epsilon_i^{**}, \epsilon_i'', \delta_{m,i}/(k_1 + k_2)\}$.

To summarize, we conclude that given the set of positive real numbers $\{\delta_{b,i}, \delta_{\zeta,i}, \delta_{m,i}, T_i^b\}$, there exists a positive real number, $\epsilon_i' > 0$, such that if $\epsilon_i \in (0, \epsilon_i']$, $V_i(x(0)) \leq \delta_{b,i}$, $\|\tilde{y}(0)\| \leq \delta_{\zeta,i}$, $w(T_i^b) = \hat{x}(T_i^b)$, the residual satisfies a relation of the form $r(t) \leq \delta_{m,i}$ for all $t \geq T_i^b$. This completes the proof of the proposition.

Remark 8.12: Note that the bound $\delta_{m,i}$ can be chosen arbitrarily small by choosing the observer gain to be sufficiently large. Note also that, unlike the case of full state feedback, the fault-detection filter is initialized only after the passage of some short period of time, $[0, T_i^b]$ (which can be chosen arbitrarily small by increasing the observer gain), to ensure that the closed-loop state estimates have converged sufficiently close to the true closed-loop states and thus – by setting the filter state w at this time

equal to the value of the state estimate – ensure that the filter state is initialized sufficiently close to the true values of the state. From this point onwards, the filter simply integrates a replica of the dynamics of the process in the absence of errors. In the absence of actuator faults, the difference between the filter states and the process states is a function of the initial error, which can be bounded from above by a value that can be made as small as desired by decreasing the initial error, which in turn can be done by appropriate choice of the observer parameters.

Having established a bound on the residual in the absence of faults, we are now ready to proceed with the design of the switching logic. To this end, consider the nonlinear system of Eq.8.1 where, for each control configuration, an output feedback controller of the form of Eq.8.17 is available and, given the desired output feedback stability regions $\Omega_{b,i} \subset \Omega_i$, $i = 1, \dots, N$, as well as the desired values for $\delta_{m,i}$, T_b^i , an appropriate observer gain has been determined (e.g., $\epsilon_i \leq \min\{\epsilon_i^*, \epsilon_i', \epsilon_i^{**}\}$ to guarantee both stability and satisfaction of the desired bound on the residual) and the sets $\Omega_{s,i}$ (see Proposition 8.2) have been computed. The implementation of the fault-detection filter and fault-tolerant controller is described in Theorem 8.2 below.

Theorem 8.2: *Let $k(0) = i$ for some $i \in \mathcal{K}$, $x(0) \in \Omega_{b,i}$, $w(T_i^b) = \hat{x}(T_i^b)$, and consider a fault for which $r(T_i^s) \geq \delta_{m,i}$, where $T_i^s > T_i^b$ is the earliest time for which $r(t) \geq \delta_{m,i}$. Then under the switching rule*

$$k(t) = \left\{ \begin{array}{ll} i, & 0 \leq t < T_i^s \\ j \neq i, & t \geq T_i^s, \hat{x}(T_i^s) \in \Omega_j^s \end{array} \right\} \quad (8.22)$$

the origin of the closed-loop system is asymptotically stable.

Proof of Theorem 8.2: Consider the nonlinear system of Eq.8.1, under the output feedback controller of Eq.8.17, and the system of Eq.8.18, where $k(0) = i$ for some $i \in \mathcal{K}$, $x(0) \in \Omega_{b,i}$, $w(T_i^b) = \hat{x}(T_i^b)$, $\epsilon_i \leq \min\{\epsilon_i^*, \epsilon_i', \epsilon_i^{**}\}$, where ϵ_i^* , ϵ_i^{**} were defined in Proposition 8.1 and ϵ_i' was defined in Proposition 8.3. Since we consider only faults for which $r(T_i^s) \geq \delta_m^i$, where $T_i^s > T_i^b$ is the earliest time for which $r(t) \geq \delta_m^i$, it follows

that: (a) in the absence of such faults, no switching takes place and configuration i is implemented for all times. Since $x(0) \in \Omega_{b,i}$ and $\epsilon_i \leq \epsilon_i^*$, asymptotic closed-loop stability of the origin follows directly from Proposition 8.1.

(b) in the case that such faults take place, the earliest time a fault is detected is $T_i^s > T_i^b$ and we have, from Eq.8.22, that $k(t) = i$ for $0 \leq t < T_i^s$. From the stability of the i -th closed-loop system established in Proposition 8.1, we have that the closed-loop trajectory stays bounded within $\Omega_{b,i}$ for $0 \leq t < T_i^s$. At time T_i^s , the supervisor switches to a control configuration j for which $\hat{x}(T_i^s) \in \Omega_{s,j}$. By design, $\hat{x}(t) \in \Omega_{s,j} \implies x(t) \in \Omega_{b,j}$ for all $t \geq T_i^s > T_i^b$. From this point onwards, configuration j is implemented in the closed-loop system for all future times and, since $x(T_i^s) \in \Omega_{b,j}$, asymptotic closed-loop stability of the origin follows from the result of Proposition 8.1. This completes the proof of Theorem 8.2.

Remark 8.13: The design and implementation of the fault-detection filter and fault-tolerant controller proceed as follows:

1. Given the nonlinear process of Eq.8.1, identify the available control configurations, $k = 1, \dots, N$. For each configuration, design the output feedback controller of Eq.8.17, and for a given choice of the output feedback stability region, $\Omega_{b,i}$, determine a stabilizing observer gain, ϵ_i^* .
2. Given any positive real numbers, $\delta_{m,i}$ and T_i^b , determine the observer gain, ϵ_i' , for which the maximum possible difference between the filter states and the state estimates, in the absence of faults, is less than the threshold $\delta_{m,i}$ for all times greater than T_i^b .
3. Given the output feedback stability region, $\Omega_{b,i}$, determine the maximum error, $e_{m,i}^*$, and the set $\Omega_{s,i}$ such that if $\|x - \hat{x}\| \leq e_{m,i} \leq e_{m,i}^*$ (i.e., the error between the estimates and the true values of the states is less than $e_{m,i}$) and $\hat{x} \in \Omega_{s,i}$ (i.e., the state estimates belong to $\Omega_{s,i}$), then $x \in \Omega_{b,i}$ (i.e., the state belongs to the

output feedback stability region).

4. For a choice of $e_{m,i} \in (0, e_{m,i}^*]$ and given T_i^b , determine the observer gain, ϵ_i^{**} , for which the maximum possible difference between the states and the state estimates, in the absence of faults, is less than the threshold $e_{m,i}$ for all times greater than T_i^b . Set $\epsilon_i := \min\{\epsilon_i^*, \epsilon_i', \epsilon_i^{**}\}$. Note that this choice guarantees that by time T_i^b : (1) the residual is within the desired threshold and (2) the presence of $\hat{x}$ within $\Omega_{s,i}$ guarantees that x belongs to $\Omega_{b,i}$.
5. Initialize the closed-loop system such that $x(0) \in \Omega_{b,i}$, for some $i \in \mathcal{K}$, and start generating the state estimates $\hat{x}(t)$. At time T_i^b , initialize and start integrating the filter dynamics of Eq.8.18 with $w(T_i^b) = \hat{x}(T_i^b)$, where $\hat{x}$ is the state estimate generated by the high-gain observer.
6. At the earliest time $T_i^s > T_i^b$ that $r(t) > \delta_{m,i}$ (implying that the difference between the expected evolution of the process states and the estimates of the process states is more than what can be accounted for by the error in the initialization of the filter states, implying that a fault has occurred), activate the backup configuration for which $\hat{x}(T_i^s) \in \Omega_{s,j}$ (note that since $t = T_i^s > T_i^b$, we have that $\|x(T_i^s) - \hat{x}(T_i^s)\| \leq e_{m,i}$; this together with $\hat{x}(T_i^s) \in \Omega_{s,j}$ implies that $x(T_i^s) \in \Omega_{b,j}$, i.e., the state belongs to the stability region of configuration j). Implement the backup configuration j to achieve closed-loop stability.

Remark 8.14: Theorem 8.2 considers faults that are “observable” from the filter’s residual, in the sense that if the residual in Eq.8.18 exceeds the allowable threshold $\delta_{m,i}$ at any time, then the supervisor can conclude with certainty that a fault has occurred. On the other hand, if the residual does not exceed the allowable threshold, it might still be possible that some “unobservable” fault – the effect of which is within the filter threshold – has taken place. Note that in contrast to the case of full state feedback, the states in this case are only known up to a certain degree of accuracy.

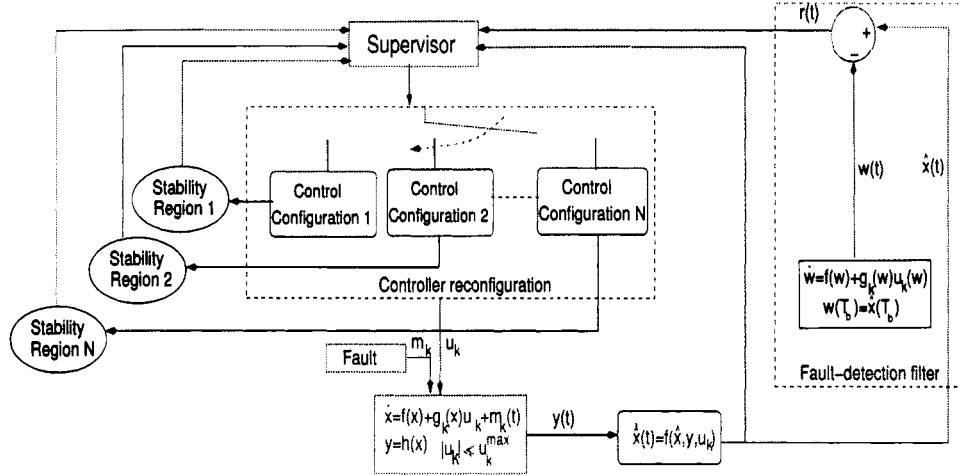


Figure 8.7: Integrated fault-detection and fault-tolerant control design under output feedback.

Therefore, any fault that causes a difference in the closed-loop behavior that is within that margin of (i.e., indistinguishable from) the effect of the estimation error will, in principle, go undetected. This class of faults is not considered in Theorem 8.2 since its effect on closed-loop stability cannot be discerned from the behavior of the residual. This, however, is not a restriction since the observability threshold $\delta_{m,i}$ is a design parameter and can be chosen arbitrarily small, thus rendering the possibility of major (i.e., destabilizing) faults that cannot be detected quite small. Ultimately, the choice of $\delta_{m,i}$ reflects a fundamental tradeoff between the need to avoid false alarms that could be caused by estimation errors (this favors a relatively large threshold) and the need to minimize the possibility of some faults going undetected (this favors a relatively small threshold).

Remark 8.15: Note that for all times prior to T_i^b , the filter is inactive. Up-until this time, the state estimates have not yet converged close enough to the true values of the states, and no inference about the state of the system can be drawn by looking at the evolution of the state estimate, and therefore no inference about any possible faults can be drawn via the fault-detection filter. If a fault occurs within this time, the filter will detect its occurrence only after the time T_i^b . By choosing a larger value of the

observer gain, however, the time T_i^b can be reduced further, if so desired. Note also that while we consider the problem of unavailability of some of the state variables as measurements, we do not consider the problem of sensor faults, i.e., we assume that the sensors do not malfunction both in the state and output feedback cases. In the event of availability of multiple measurements in a way that each of them can be used to estimate of the process states, the estimates of the states generated using the different measurements can be used to also detect sensor faults.

Remark 8.16: The central idea behind the model-based fault-detection filter design, that of comparing the evolution of the process to the expected evolution of the process in the absence of faults, can also be used to design a rule-based fault-detection filter. One example of a rule-based fault-detection filter is to declare a fault if the state estimates, after a time T_i^b , touch the boundary of $\Omega_{s,i}$, indicating that the closed-loop states themselves may be about to escape the output feedback stability region $\Omega_{b,i}$. The rule-based fault detection filter, however, would be able to detect the fault only when the state estimates hit the boundary of $\Omega_{s,i}$, as opposed to the model-based fault detection filter, which detects a fault as soon as the effect of the fault on the closed-loop evolution goes beyond a prescribed threshold. This delay in a rule-based approach could result in the state escaping the stability region of the available backup configurations (see subsection 8.4.3 for an example). Also, it may happen that the fault causes the closed-loop process states evolving within $\Omega_{s,i}$ to neither escape $\Omega_{s,i}$ nor converge to the origin. The rule based fault-detection filter would not be able to detect such a fault. In contrast, the model-based fault-detection filter of Theorem 8.2, is able to detect faults that have an effect, up-to a desirable threshold, on the evolution of the closed-loop process. Note also that the model-based fault-detection filter of Theorem 8.2 and the rule-based fault-detection filter discussed above differ only in that the model-based filter of Theorem 8.2 uses a more quantitative knowledge of the closed-loop dynamics to predict the expected closed-loop trajectory, instead of using the qualitative knowledge that the fault-free closed-loop state trajectory does

not the escape the stability region.

8.4.3 Simulation results

In this subsection, we illustrate the implementation of the proposed fault-tolerant control methodology to the chemical reactor example introduced in subsection 8.2.2. Fig.8.11 depicts the stability region, in the (T, C_A) space, for each configuration. The desired steady-state is depicted with an asterisk that lies in the intersection of the three stability regions. For the first two control configurations, a state estimator of the form of Eq.8.17 is designed. For thresholds of $\delta_m = 0.0172$ and 0.00151 in the fault detection filters, the parameters in the observer of Eq.8.17 are chosen as $L_1 = L_2 = 100$, $a_1^{(1)} = a_1^{(2)} = 10$ and $a_2^{(1)} = a_2^{(2)} = 20$. For the third configuration, the estimates, $\hat{T}$, $\hat{C}_A$ are generated as follows:

$$\begin{aligned}\frac{d\hat{T}}{dt} &= \frac{F}{V}(T_{A0} - \hat{T}) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} k_{i0} e^{\frac{-E_i}{R\hat{T}}} \hat{C}_A + \alpha_1(C_A - \hat{C}_A) \\ \frac{d\hat{C}_A}{dt} &= \frac{F}{V}(C_{A0} - \hat{C}_A) - \sum_{i=1}^3 k_{i0} e^{\frac{-E_i}{R\hat{T}}} \hat{C}_A + \alpha_2(C_A - \hat{C}_A)\end{aligned}\tag{8.23}$$

where $\alpha_1 = -10^4$ and $\alpha_2 = 10$. The reactor is initialized at $T(0) = 330 \text{ K}$, $C_A(0) = 3.6 \text{ kmol/m}^3$, $C_B(0) = 0.0 \text{ kmol/m}^3$, using the Q -control configuration, while the state estimates are initialized at $\hat{T}(0) = 390 \text{ K}$, $\hat{C}_A(0) = 3.6 \text{ kmol/m}^3$ and the supervisor proceeds to monitor the evolution of the closed-loop estimates.

We first demonstrate the need to wait for a sufficient time before initializing the filter. To this end, consider the fault-detection filter initialized at $t = 0.005$ minutes $\equiv T_1^b$ at which time the state estimates (dash-dotted lines in Fig.8.8) have not converged to the true values (solid lines in Fig.8.8). As a result, the fault-detection filter shows a false alarm (see Fig.8.9a) by crossing the threshold even when control configuration 1 is functioning properly (see Fig.8.9b) and stabilizes the closed-loop system. Note that while the initialization of the filter at a time when the state estimates have not converged leads to the residual crossing the threshold, the residual eventually goes

to zero as expected, since both the filter states and the closed-loop process states eventually stabilize and go to the same equilibrium point.

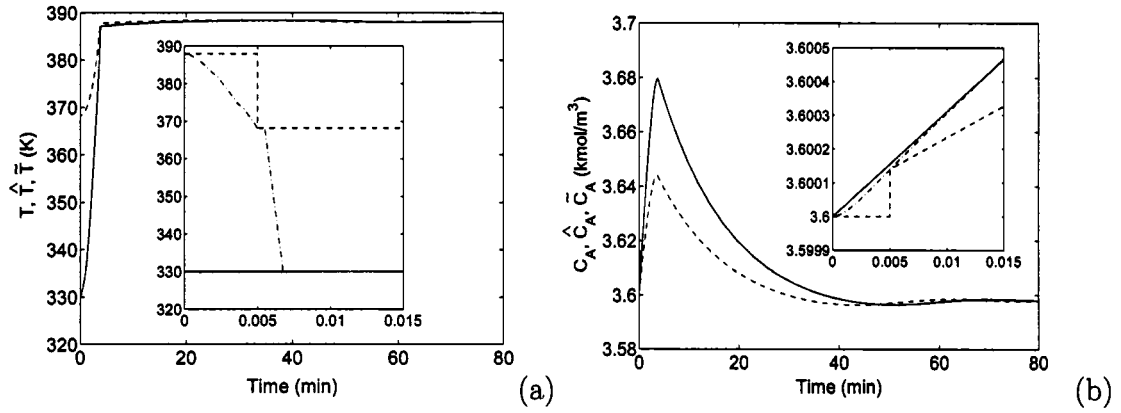


Figure 8.8: Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under control configuration 1 when the fault detection filter is initialized at $t = 0.005$ minutes.

We now demonstrate the application of the fault-detection filter and fault-tolerant controller of Theorem 8.2. Starting from the same initial conditions, the estimates of T and C_A (dash-dotted lines in Figs.8.10a,b) converge very quickly to the true values of the states (solid lines in Figs.8.10a,b). The states in the fault-detection filter are initialized and set equal to the value of the state estimates at $t = 0.01$ minutes $\equiv T_1^b$; note that by this time the estimates have converged to the true values. By initializing the fault-detection filter appropriately, a false alarm is prevented (the value of $r_1(t)$ does not hit the threshold in the absence of a fault after a time $t = 0.01$ minutes, see Fig.8.12a). As shown by the solid lines in Fig.8.11, the controller proceeds to drive the closed-loop trajectory towards the desired steady-state, up until the Q -configuration fails after 3.0 minutes $\equiv T_1^f$ of reactor startup (see solid lines in Fig.8.14a). Note that at this time, the value of $r_1(t)$ becomes non-zero and hits the threshold at $t = 3.3$ minutes $\equiv T_1^s$. From Fig.8.11, it is clear that the failure of the primary control configuration

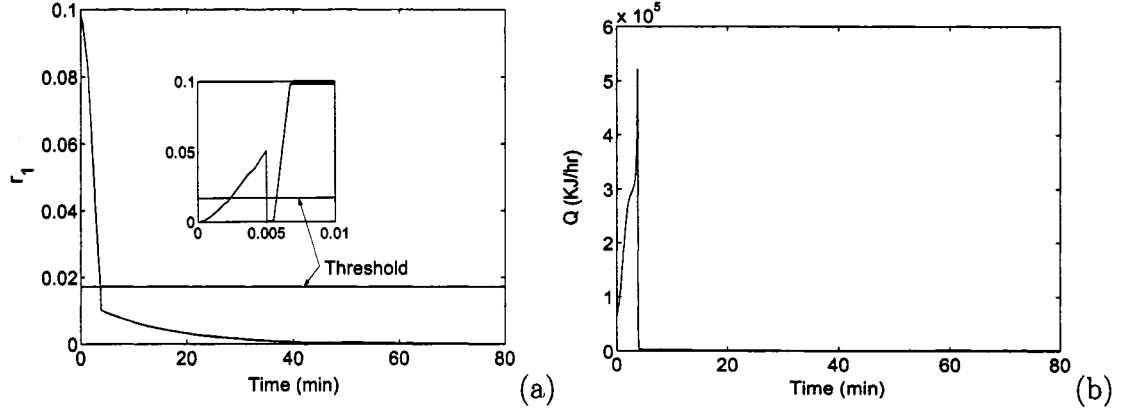


Figure 8.9: Evolution of (a) the residual and (b) the manipulated input profile for the first control configuration when the fault detection filter is initialized at $t = 0.005$ minutes.

occurs when the closed-loop trajectory is within the stability region of the second control configuration, and outside the stability region of the third control configuration. Therefore, on the basis of the switching logic of Eq.8.22, the supervisor activates the second configuration (with T_{A0} as the manipulated input). The result is shown by the solid line in Fig.8.11 where it is seen that upon switching to the T_{A0} -configuration, the corresponding controller continues to drive the state trajectory closer to the desired steady-state.

When a second failure occurs (this time in the T_{A0} -configuration) at $t = 13.0$ minutes $\equiv T_2^f$ (which is simulated by fixing T_{A0} for all $t \geq 13.0$ minutes, see solid lines in Fig.8.14b) before the process has reached the steady state, the filter detects this failure via the value of $r_2(t)$ hitting the threshold (see Fig.8.12b). From the solid line in Fig.8.11, it is clear that the failure of the second control configuration occurs when the closed-loop trajectory is within the stability region of the third configuration. However, if the fault-detection filter is not in place and the backup configuration is implemented late in the closed-loop (at $t = 30$ minutes $\equiv T_3^s$), by this time the state of the closed-loop system has moved out of the stability region of the third control configuration, and closed-loop stability is not achieved (see dashed line in Fig.8.11,

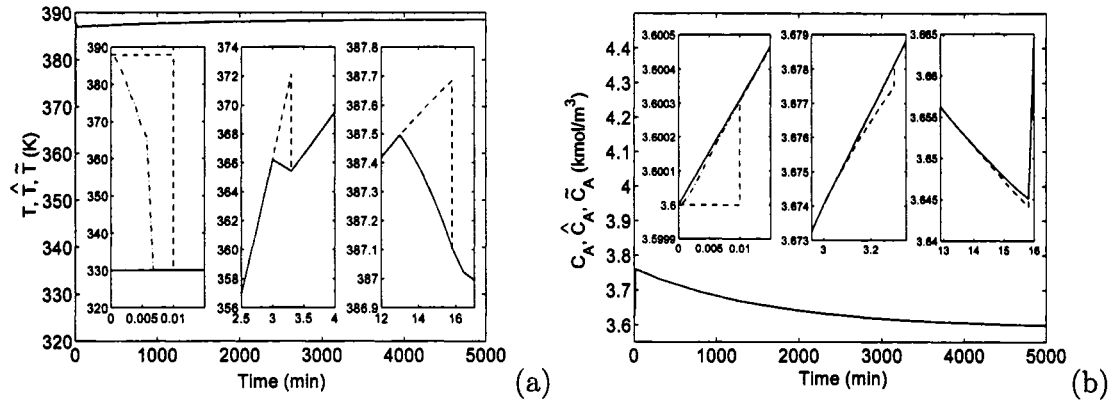


Figure 8.10: Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2.

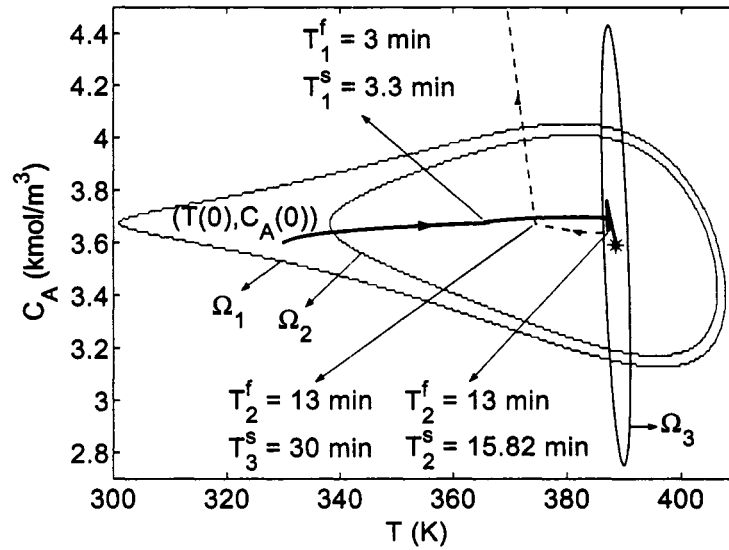


Figure 8.11: Evolution of the closed-loop state trajectory under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2, using an appropriate fault-detection filter (solid line) and in the absence of a fault-detection filter (dashed line).

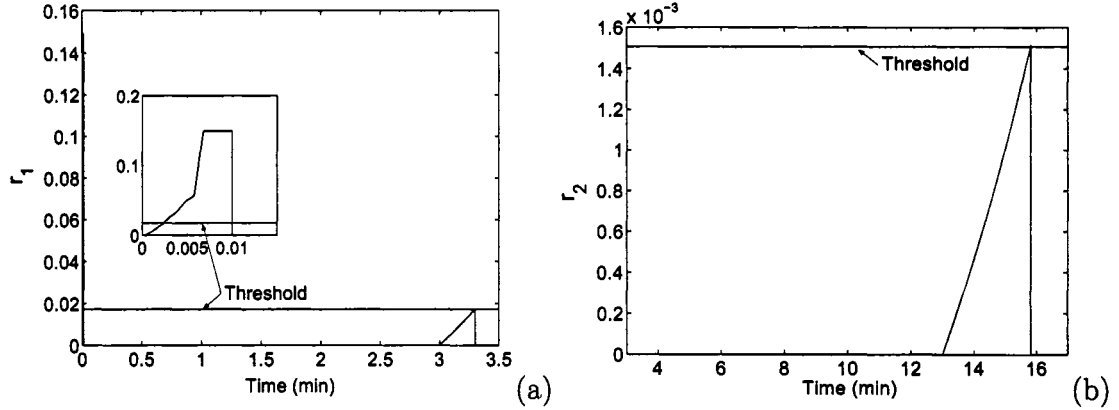


Figure 8.12: Evolution of the residual for (a) the first control configuration and (b) the second control configuration.

see also Fig.8.13 and dashed lines in Fig.8.14). In contrast, when the fault-detection filter is in place, it detects a fault at $t = 15.82$ minutes $\equiv T_2^s$ and when the supervisor switches to configuration 3, closed-loop stability is achieved (see solid line in Fig.8.11).

8.5 Conclusions

In this chapter, an integrated fault-detection and fault-tolerant control (FDFTC) structure, for nonlinear processes with input constraints subject to control actuator failures, was presented. Under the assumption of full state feedback, the FDFTC structure comprised of (1) a family of control configurations, each with a stabilizing feedback controller and an explicitly characterized stability region, (2) a fault-detection filter that detects faults by comparing the fault-free behavior of the closed-loop states with their actual behavior, and (3) a high-level supervisor that orchestrates switching between the control configurations, based on the stability regions, once a fault is detected. When measurements of the full state were not available, a nonlinear observer with sufficiently fast convergence properties was incorporated into the FDFTC structure to generate appropriate state estimates that were used to implement the state feedback controllers, the fault-detection filter and the switching logic. It was shown that by properly tuning

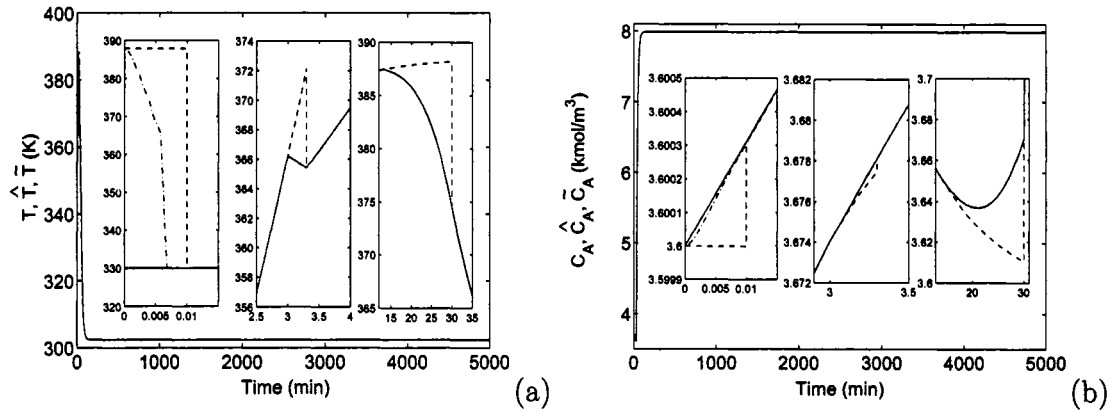


Figure 8.13: Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2 in the absence of a fault-detection filter.

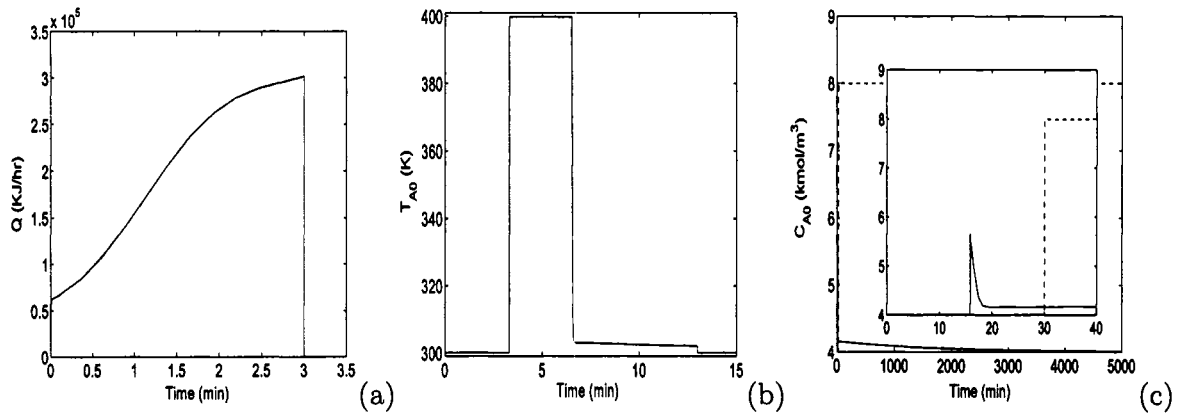


Figure 8.14: Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Eq.8.22 subject to failures in control systems 1 and 2 in the presence (solid lines) and absence (dashed lines) of a fault-detection filter.

the observer parameters and modifying the implementation of the filter, the effect of the estimation error on the filter's residual could be decoupled from the effect of faults, thus preventing unnecessary false alarms. Finally, simulation studies were presented to demonstrate the implementation of the proposed fault-tolerant control scheme.

Chapter 9

Predictive Control of Linear Parabolic PDEs with State and Control Constraints

9.1 Introduction

Transport-reaction processes that are characterized by significant spatial variations due to the underlying diffusion and convection phenomena. The dynamic models of transport-reaction processes over finite spatial domains typically consist of parabolic partial differential equation (PDE) systems whose spatial differential operators are characterized by a spectrum that can be partitioned into a finite (possibly unstable) slow part and an infinite stable fast complement [45]. The traditional approach to control of linear/quasi-linear parabolic PDEs involves the application of spatial discretization techniques to the PDE system to derive systems of ordinary differential equations (ODEs) that accurately describe the dynamics of the dominant (slow) modes of the PDE system. The finite-dimensional systems are subsequently used as the basis for the synthesis of finite-dimensional controllers (e.g., see [15, 185, 43]). A potential drawback of this approach, especially for quasi-linear parabolic PDEs, is that the num-

ber of modes that should be retained to derive an ODE system that yields the desired degree of approximation may be very large, leading to complex controller design and high dimensionality of the resulting controllers.

Motivated by these considerations, significant recent work has focused on the development of a general framework for the synthesis of low-order controllers for quasi-linear parabolic PDE systems – and other highly dissipative PDE systems that arise in the modeling of spatially-distributed systems – on the basis of low-order nonlinear ODE models derived through a combination of the Galerkin method (using analytical or empirical basis functions) with the concept of inertial manifolds [37]. Using these order reduction techniques, a number of control-relevant problems, such as nonlinear and robust controller design, dynamic optimization, and control under actuator saturation have been addressed for various classes of dissipative PDE systems (e.g., see [41, 14, 8, 63] and the book [37] for results and references in this area). In addition to these works, other recent studies in control of PDE systems include [212, 52]. The approaches proposed in the above works, however, do not address the issue of state constraints in the controller design. Operation of transport-reaction processes typically requires that the state of the closed-loop system be maintained within certain bounds to achieve acceptable performance (for example, requiring reactor temperature not to exceed a certain value or requiring a product concentration not to drop below some purity requirement). Handling both state and control constraints – the latter typically arising due to the finite capacity of control actuators – in the design of the controller, therefore, is an important consideration.

Model Predictive Control (MPC), also known as receding horizon control, is a popular control method for handling constraints (both on manipulated inputs and state variables) within an optimal control setting. In MPC, the control action is obtained by solving repeatedly, on-line, a finite-horizon constrained open-loop optimal control problem. The popularity of this approach stems largely from its ability to handle,

among other issues, multi-variable interactions, constraints on controls and states, and optimization requirements. Numerous research studies have investigated the properties of model predictive controllers and led to a plethora of MPC formulations that focus on a number of control-relevant issues, including issues of closed-loop stability, performance, implementation and constraint satisfaction (e.g., see [84, 3, 179, 141] for surveys of results and references in this area).

Most of the research in the area of predictive control, however, has focused on lumped-parameter processes modeled by ODE systems. Compared with lumped-parameter systems, the problem of designing predictive controllers for distributed parameter systems, modeled by PDEs, has received much less attention. Of the few results available on this problem, some have focused on analyzing the receding horizon control problem on the basis of the infinite-dimensional system using control Lyapunov functionals (e.g., [102]), while others have used spatial discretization techniques such as finite differences (e.g., [57]) to derive approximate ODE models (of possibly high-order) for use within the MPC design, thus leading to computationally expensive model predictive control designs that are, in general, difficult to implement on-line.

Motivated by the above considerations, we focus in this chapter on the development of a framework for the design of predictive controllers for linear parabolic PDEs with state and control constraints. The rest of the chapter is organized as follows. In section 9.2, a class of parabolic PDEs is described and formulated as an infinite-dimensional system, and the predictive control problem is formulated on the basis of the infinite-dimensional system. Then, in section 9.3, modal decomposition techniques are used to derive a finite-dimensional system that captures the dominant dynamics of the infinite-dimensional system and to express the state constraints for the infinite dimensional system in terms of the finite dimensional system state constraints. A number of MPC formulations, designed on the basis of different finite-dimensional approximations, are presented and compared. The closed-loop stability properties of the

infinite dimensional system under the low order MPC controller designs are analyzed and sufficient conditions, which guarantee stabilization and state constraint satisfaction for the infinite dimensional system under the reduced order MPC formulations, are derived. We also present other formulations which differ in the way the evolution of the fast eigenmodes is accounted for in the performance objective and state constraints. The impact of these differences on the ability of the predictive controller to enforce closed-loop stability and state constraints satisfaction in the infinite-dimensional system is analyzed. Finally, in section 9.4, the MPC formulations are applied, through simulations, to the problem of stabilizing the spatially-uniform unstable steady-state of a linear parabolic PDE subject to state and control constraints.

9.2 Preliminaries

9.2.1 Parabolic PDEs

To provide motivation for the class of infinite dimensional systems considered, we focus on a linear parabolic PDE with distributed control of the form:

$$\frac{\partial \bar{x}}{\partial t} = b \frac{\partial^2 \bar{x}}{\partial z^2} + c \bar{x} + w \sum_{i=1}^m b_i(z) u_i \quad (9.1)$$

with the following boundary and initial conditions:

$$\bar{x}(0, t) = 0, \quad \bar{x}(\pi, t) = 0, \quad \bar{x}(z, 0) = \bar{x}_0(z) \quad (9.2)$$

subject to the following input and state constraints:

$$u_i^{min} \leq u_i \leq u_i^{max}, \quad i = 1, \dots, m \quad (9.3)$$

$$\chi^{min} \leq \int_0^\pi r(z) \bar{x}(z, t) dz \leq \chi^{max} \quad (9.4)$$

where $\bar{x}(z, t)$ denotes the state variable, $z \in [0, \pi]$ is the spatial coordinate, $t \in [0, \infty)$ is the time, $u_i \in \mathbb{R}$ denotes the i -th constrained manipulated input; u_i^{min} and u_i^{max} are real numbers representing the lower and upper bounds on the i -th input, respectively, and χ^{min} and χ^{max} are real numbers representing the lower and upper state constraints,

respectively. The term $\frac{\partial^2 \bar{x}}{\partial z^2}$ denotes the second-order spatial derivative of $\bar{x}$; b , c and ω are constant coefficients with $b > 0$, and $\bar{x}_0(z)$ is a sufficiently smooth function of z . The function $b_i(z)$ is a known smooth function of z that describes how the control action, $u_i(t)$, is distributed in the spatial interval $[0, \pi]$. Whenever the control action is applied to the spatial domain at a single point z_a , with $z_a \in [0, \pi]$ (i.e., point actuation), the function $b_i(z)$ is taken to be nonzero in a finite spatial interval of the form $[z_a - \mu, z_a + \mu]$, where μ is a small positive real number, and zero elsewhere in $[0, \pi]$. The smooth function $r(z)$ is a “state constraint distribution” function that describes how the state constraint is enforced in the spatial domain $[0, \pi]$. Throughout this chapter, the notation $|\cdot|$ will be used to denote the standard Euclidian norm in $\mathbb{R}^n$, while the notation $|\cdot|_Q$ will be used to denote the weighted norm defined by $|\hat{x}|_Q^2 = \hat{x}' Q \hat{x}$, where Q is a positive-definite matrix and $\hat{x}'$ denotes the transpose of $\hat{x}$. A function $\beta(r, s) : [0, a) \times [0, \infty) \rightarrow [0, \infty)$ is said to be of class $\mathcal{KL}$ if, for each fixed $s \geq 0$, the mapping $\beta(r, s)$ is continuous, strictly increasing with respect to r and satisfies $\beta(0, s) = 0$, and, for each fixed r , the mapping $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \rightarrow 0$, as $s \rightarrow \infty$.

To proceed with the presentation of our results, we formulate the PDE of Eqs.9.1-9.2-9.3-9.4 as an infinite dimensional system in the state space $\mathcal{H} = L_2(0, \pi)$, with inner product and norm

$$(\omega_1, \omega_2) = \int_0^\pi \omega_1(z) \omega_2(z) dz, \quad \|\omega_1\|_2 = (\omega_1, \omega_1)^{\frac{1}{2}} \quad (9.5)$$

where ω_1, ω_2 are two elements of $L_2(0, \pi)$.

Defining the state function $x(t)$ on the state-space $\mathcal{H}$ as

$$x(t) = \bar{x}(z, t), \quad t > 0, \quad 0 < z < \pi, \quad (9.6)$$

the operator $\mathcal{A}$ as

$$\mathcal{A}\phi = b \frac{d^2 \phi}{dz^2} + c\phi, \quad 0 < z < \pi, \quad (9.7)$$

where $\phi(z)$ is a smooth function on $(0, \pi)$ with $\phi(0) = 0$ and $\phi(\pi) = 0$, with the following dense domain

$$\begin{aligned} \mathcal{D}(\mathcal{A}) = \{ \phi(z) \in L_2(0, \pi) : \phi(z), \frac{d\phi(z)}{dz} \text{ are absolutely continuous,} \\ \mathcal{A}\phi \in L_2(0, \pi), \phi(0) = 0 \text{ and } \phi(\pi) = 0 \}, \end{aligned} \quad (9.8)$$

the input operator as:

$$\mathcal{B}u = \sum_{i=1}^m b_i(\cdot) u_i, \quad (9.9)$$

and the state constraint as:

$$\chi^{min} \leq (r, x) \leq \chi^{max}, \quad (9.10)$$

the system of Eqs.9.1-9.2-9.3-9.4 can be written as:

$$\dot{x} = \mathcal{A}x + \mathcal{B}u, \quad x(0) = x_0, \quad (9.11)$$

$$u_i^{min} \leq u_i(t) \leq u_i^{max}, \quad (9.12)$$

$$\chi^{min} \leq (r, x) \leq \chi^{max} \quad (9.13)$$

The spectrum of $\mathcal{A}$ can be obtained by solving the following eigenvalue problem:

$$\mathcal{A}\phi_j = b \frac{d^2 \phi_j}{dz^2} + c \phi_j = \lambda_j \phi_j \quad (9.14)$$

subject to

$$\phi_j(0) = \phi_j(\pi) = 0 \quad (9.15)$$

where λ_j denotes an eigenvalue and ϕ_j denotes an eigenfunction. A direct computation of the solution of the above eigenvalue problem yields

$$\lambda_j = c - bj^2, \quad \phi_j(z) = \sqrt{\frac{2}{\pi}} \sin(jz), \quad j = 1, \dots, \infty \quad (9.16)$$

The spectrum of $\mathcal{A}$, $\sigma(\mathcal{A})$, is defined as the set of all eigenvalues of $\mathcal{A}$, i.e., $\sigma(\mathcal{A}) = \{\lambda_1, \lambda_2, \dots\}$. From the expression for the eigenvalues, it is clear that all the eigenvalues of $\mathcal{A}$ are real, and that, for a given b and c , only a finite number of unstable eigenvalues exist, and the distance between any two consecutive eigenvalues

(i.e., λ_j and λ_{j+1}) increases as j increases. Furthermore, $\sigma(\mathcal{A})$ can be partitioned as $\sigma(\mathcal{A}) = \sigma_1(\mathcal{A}) \cup \sigma_2(\mathcal{A})$, where $\sigma_1(\mathcal{A}) = \{\lambda_1, \dots, \lambda_m\}$ contains the first m (with m finite) “slow” eigenvalues (including all, if any, possibly unstable eigenvalues) and $\sigma_2(\mathcal{A}) = \{\lambda_{m+1}, \lambda_{m+2}, \dots\}$ contains the remaining “fast” stable eigenvalues. This implies that the dominant dynamics of the PDE can be described by a finite-dimensional system, and motivates the use of modal decomposition to derive a finite-dimensional system that captures the dominant (slow) dynamics of the PDE.

From the properties of $\mathcal{A}$ and its spectrum, it follows (Theorem 2.10 in [45]) that $\mathcal{A}$ generates a strongly continuous $\mathcal{C}_0$ -semigroup, $\mathcal{T}(t)$. Moreover, since $\mathcal{B}$ is a bounded operator, the system of Eq.9.11 has a mild solution (Theorem 2.31 in [45]) of the form:

$$x(t) = \mathcal{T}(t)x_0 + \int_0^t \mathcal{T}(t-\tau)\mathcal{B}u(\tau)d\tau \quad (9.17)$$

Remark 9.1: Note that while we use the PDE system of Eqs.9.1-9.4 to motivate and illustrate the development of the infinite dimensional system of Eq.9.11, our subsequent results are not limited to single PDEs of the form of Eqs.9.1-9.4. The results developed in this work apply to parabolic PDEs, with possibly other types of boundary condition (for example, mixed boundary condition), and also systems of parabolic PDEs, as long as they possess operators $\mathcal{A}$ for which the eigenspectrum can be partitioned into $\sigma_1(\mathcal{A})$ and $\sigma_2(\mathcal{A})$, where $\sigma_1(\mathcal{A})$ and $\sigma_2(\mathcal{A})$ possess the properties described above.

9.2.2 MPC formulation

Referring to the system of Eq.9.11, we consider the problem of asymptotic stabilization of the origin, subject to the control constraints of Eq.9.12 and the state constraint of Eq.9.13. The problem will be addressed within the MPC framework (see [141] for a review of various MPC algorithms for finite-dimensional systems) where the control, at state x and time t , is conventionally obtained by solving, on-line, a finite-horizon constrained optimal control problem of the form:

$$P(x, t) : \min\{J(x, t, u(\cdot)) \mid u(\cdot) \in S\} \quad (9.18)$$

$$\begin{aligned}
s.t. \quad \dot{x}(\tau) &= \mathcal{A}x(\tau) + \mathcal{B}u(\tau) \\
u(\tau) &\in \mathcal{U} \\
\chi^{min} &\leq (r, x(\tau)) \leq \chi^{max}, \quad \tau \in [t, t+T]
\end{aligned} \tag{9.19}$$

where $S = S(t, T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ , mapping $[t, t+T]$ into $\mathcal{U} := \{u \in \mathbb{R}^m : u_i^{min} \leq u_i \leq u_i^{max}, i = 1, \dots, m\}$, and T is the specified horizon. A control $u(\cdot)$ in S is characterized by the sequence $u[k]$, where $u[k] := u(k\Delta)$, and satisfies $u(t) = u[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. The performance index is given by

$$J(x, t, u(\cdot)) = \int_t^{t+T} [q\|x^u(\tau; x, t)\|_2^2 + |u(\tau)|_R^2] d\tau + F(x(t+T)) \tag{9.20}$$

where q is a strictly positive real number, $x^u(\tau; x, t)$ denotes the solution of Eq.9.11, due to control u , with initial state x at time t , and $F(\cdot)$ denotes the terminal penalty. The minimizing control $u^0(\cdot) \in S$ is then applied to the system over the interval $[k\Delta, (k+1)\Delta]$ and the procedure is repeated indefinitely. This defines an implicit model predictive control law

$$M(x) := u^0(t; x, t) \tag{9.21}$$

The predictive controller problem of Eqs.9.18-9.19 is formulated on the basis of the infinite dimensional system, and therefore, it leads to a predictive controller that is of higher-order and can not be readily implemented in practice. To overcome this problem, in the next section, we develop computationally efficient predictive control formulations that achieve stabilization of the system of Eq.9.11 subject to the control and state constraints of Eqs.9.12-9.13.

Remark 9.2: It is well known that even for the finite-dimensional system the control law defined by Eqs.9.18-9.21 is not necessarily stabilizing. For finite-dimensional systems, the issue of closed-loop stability is usually addressed by means of imposing suitable penalties and constraints on the state at the end of the optimization horizon (e.g., see [3, 141] for surveys of different approaches).

9.3 Predictive control of infinite dimensional systems

Initially, we apply modal decomposition techniques to the system of Eq.9.11 to derive a finite-dimensional system that captures the dominant dynamics of the infinite-dimensional system. The state constraints of the infinite-dimensional system are appropriately expressed in terms of constraints on the finite-dimensional system. The finite-dimensional system is used for construction of a low-order predictive controller and the closed-loop stability properties of the infinite dimensional system are analyzed and sufficient conditions, which guarantee stabilization and state constraint satisfaction for the infinite dimensional system under the reduced-order predictive controller. Other MPC formulations, designed on the basis of the finite-dimensional approximations, that differ in the way the state constraints are handled within the optimization problem are then presented and compared in terms of their ability to enforce closed-loop stability and ensure constraint satisfaction for the state of the infinite dimensional system.

9.3.1 Modal decomposition

In this section, we apply standard modal decomposition to the infinite-dimensional system of Eq.9.11 to derive a finite-dimensional system. Let $\mathcal{H}_s, \mathcal{H}_f$ be modal subspaces of $\mathcal{A}$, defined as $\mathcal{H}_s = \text{span}\{\phi_1, \phi_2, \dots, \phi_m\}$ and $\mathcal{H}_f = \text{span}\{\phi_{m+1}, \phi_{m+2}, \dots\}$ (the existence of $\mathcal{H}_s, \mathcal{H}_f$ follows from the properties of $\mathcal{A}$). Defining the orthogonal projection operators, P_s and P_f , such that $x_s = P_s x$, $x_f = P_f x$, the state x of the system of Eq.9.11 can be decomposed as

$$x = x_s + x_f = P_s x + P_f x \quad (9.22)$$

Applying P_s and P_f to the system of Eq.9.11 and using the above decomposition for x , the system of Eq.9.11 can be re-written in the following equivalent form

$$\begin{aligned}\frac{dx_s}{dt} &= \mathcal{A}_s x_s + \mathcal{B}_s u, \quad x_s(0) = P_s x(0) = P_s x_0 \\ \frac{dx_f}{dt} &= \mathcal{A}_f x_f + \mathcal{B}_f u, \quad x_f(0) = P_f x(0) = P_f x_0\end{aligned}\tag{9.23}$$

where $\mathcal{A}_s = P_s \mathcal{A}$, $\mathcal{B}_s = P_s \mathcal{B}$, $\mathcal{A}_f = P_f \mathcal{A}$, $\mathcal{B}_f = P_f \mathcal{B}$. In the above system, $\mathcal{A}_s$ is a diagonal matrix of dimension $m \times m$ of the form $\mathcal{A}_s = \text{diag}\{\lambda_j\}$ (λ_j are possible unstable eigenvalues of $\mathcal{A}_s$) and $\mathcal{A}_f$ is an unbounded differential operator which is exponentially stable (following from the fact that $\lambda_{m+1} < 0$ and the selection of $\mathcal{H}_s, \mathcal{H}_f$). In the remainder of this chapter, we will refer to the x_s - and x_f -subsystems in Eq.9.23 as the slow and fast subsystems, respectively. From the properties of $\mathcal{A}_s$ and $\mathcal{A}_f$ and the fact that $\mathcal{B}_s$ and $\mathcal{B}_f$ are bounded operators, it follows (Theorem 2.10 and 2.31 in [45]) that there exist C_0 -semigroups $\mathcal{T}_s$ and $\mathcal{T}_f$ such that the x_s - and x_f subsystems of Eq.9.23 admit, on the interval $t \in [0, \infty)$, the following mild solutions:

$$x_s(t) = \mathcal{T}_s(t)x_s(0) + \int_0^t \mathcal{T}_s(t-\tau)\mathcal{B}_s u(\tau)d\tau\tag{9.24}$$

$$x_f(t) = \mathcal{T}_f(t)x_f(0) + \int_0^t \mathcal{T}_f(t-\tau)\mathcal{B}_f u(\tau)d\tau\tag{9.25}$$

Furthermore, since $\mathcal{A}_f$ is a stable operator, the spectrum of $\mathcal{A}_f$ satisfies $\sup \{ \text{Re } \sigma(\mathcal{A}_f) \} < -\gamma$, for some $\gamma > |c - b(m+1)^2|$, and thus, $\mathcal{T}_f(t)$ satisfies ([45], p.74):

$$\|\mathcal{T}_f(t)\|_2 \leq M_0 e^{-\gamma t}, \quad t \geq 0\tag{9.26}$$

for some $M_0 > 0$.

9.3.2 MPC formulations: accounting for input and state constraints

We first present an MPC formulation, designed on the basis of the slow states of the system, that ensures stabilization of the infinite dimensional system. The MPC law in this case is obtained by solving, in a receding horizon fashion, the following optimization

problem

$$\min_u \left[\int_t^{t+T} [q_s \|x_s(\tau)\|_2^2 + |u(\tau)|_R^2] d\tau + F(x_s(t+T)) \right] \quad (9.27)$$

$$\begin{aligned} \text{s.t. } \dot{x}_s(\tau) &= \mathcal{A}_s x_s(\tau) + \mathcal{B}_s u(\tau) \\ u(\tau) &\in \mathcal{U} \\ \chi^{\min} &\leq (r, x_s(\tau)) \leq \chi^{\max}, \tau \in [t, t+T] \\ x_s(t+T) &= 0 \end{aligned} \quad (9.28)$$

To proceed, we assume that the predictive control law of Eqs.9.27-9.28, with a fixed horizon length T , is initially and successively feasible and achieves stabilization of the x_s -subsystem for all $x_s(0) \in \Omega_s \subset \mathcal{H}_s$. Note that the set Ω_s depends on the constraints on the states and inputs, the system dynamics and T (see remarks 9.5 and 9.6 for discussion on this issue). This assumption is precisely stated below.

Assumption 9.1: *There exists a set $\Omega_s \subset \mathcal{H}_s$ such that for all $x_s(0) \in \Omega_s$, the steady-state solution $x_s(t) = 0$ of the closed-loop system of Eq.9.24 under the MPC law of Eqs.9.27-9.28 is asymptotically stable in the sense that $x_s(t) \in \Omega_s$ for all $t \geq 0$ and satisfies $\|x_s(t)\|_2 \leq \beta(\|x_s(0)\|_2, t)$, where $\beta(\cdot, \cdot)$ is a class $\mathcal{KL}$ function.*

Proposition 9.1 below establishes that a predictive control law of Eqs.9.27-9.28, for which Assumption 9.1 holds, achieves also asymptotic stability of the closed-loop infinite dimensional system.

Proposition 9.1: *Consider the system of Eq.9.11 subject to the input and state constraints of Eqs.9.12-9.13, under the predictive controller of Eqs.9.27-9.28 for which Assumption 9.1 holds. If the initial condition of the infinite dimensional system, $x(0)$, is such that $x_s(0) \in \Omega_s$, then $x(t) = 0$ is an asymptotically stable solution of the closed-loop infinite dimensional system.*

Proof of Proposition 9.1: We first note that the control law of Eqs.9.27-9.28 is only a function of the state of the slow subsystem $u = M(x_s)$ and satisfies $\lim_{t \rightarrow \infty} u(t) = 0$ (this follows from the assumption that $x(t) = 0$ is a steady-state solution of the closed-loop system of Eqs.9.24-9.27-9.28, Assumption 9.1). Using the decomposition of Eq.9.23, the closed-loop system of Eq.9.11 under the MPC of Eqs. 9.27-9.28 can therefore be

written as a cascaded system of the form:

$$\frac{dx_s}{dt} = \mathcal{A}_s x_s + \mathcal{B}_s M(x_s) \quad (9.29)$$

$$\frac{dx_f}{dt} = \mathcal{A}_f x_f + \mathcal{B}_f M(x_s) \quad (9.30)$$

From Assumption 9.1, we have that since $x_s(0) \in \Omega_s$, $x_s(t)$ of Eq.9.29 satisfies $\|x_s(t)\| \leq \beta(\|x_s(0)\|_2, t)$ for all $t \geq 0$. We now show that the fast, x_f -subsystem of Eq.9.30, is also asymptotically stable, and thus, the infinite-dimensional closed-loop system of Eqs.9.11-9.12-9.13 under the control of Eqs.9.27-9.28 is asymptotically stable. To this end, we first note that since $\mathcal{A}_f$ is a stable operator which generates a C_0 -semigroup $\mathcal{T}_f$ that satisfies Eq.9.26 and $\mathcal{B}_f$ is a bounded operator (boundedness of $\mathcal{B}_f$ follows from the fact that $b(z) \in L_2(0, \pi)$), by taking the 2-norm in space of both sides of Eq.9.25, the following bound for the state of the x_f subsystem of Eq.9.30 can be written:

$$\|x_f(t)\|_2 \leq K e^{-\gamma t} \|x_f(0)\|_2 + \|\mathcal{B}_f\|_2 \left| \int_0^t e^{-\gamma(t-\tau)} u(\tau) d\tau \right| \quad (9.31)$$

where K is a positive real number and $u(\tau) = M(x_s(\tau))$. Further, since $u(t)$ is bounded (i.e., $|u(t)| \leq \bar{u}$ where $\bar{u} = \max\{|u^{max}|, |u^{min}|\}$) and $e^{-\gamma(t-\tau)} \geq 0$ for $0 \leq \tau \leq t$, Eq.9.31 can be written as:

$$\|x_f(t)\|_2 \leq K e^{-\gamma t} \|x_f(0)\|_2 + \|\mathcal{B}_f\|_2 \sup_{0 \leq \tau \leq t} |u(\tau)| \left| \int_0^t e^{-\gamma(t-\tau)} d\tau \right| \quad (9.32)$$

or

$$\|x_f(t)\|_2 \leq K e^{-\gamma t} \|x_f(0)\|_2 + M_2 \sup_{0 \leq \tau \leq t} |u(\tau)| \quad (9.33)$$

where $M_2 = \|\mathcal{B}_f\|_2/\gamma$. To use Eq.9.33 to prove that $\lim_{t \rightarrow \infty} \|x_f\|_2 = 0$, we need the following argument. First, we note that by taking the supremum over time of both sides of the inequality of Eq.9.33, we have:

$$\sup_{t \geq 0} \|x_f(t)\|_2 \leq K \|x_f(0)\|_2 + M_2 \bar{u} =: \bar{K} \quad (9.34)$$

The fact that the inequality of Eq.9.33 holds for every initial time, t_0 , yields that for $\forall t \geq t_0$:

$$\|x_f(t)\|_2 \leq K e^{-\gamma(t-t_0)} \|x_f(t_0)\|_2 + M_2 \sup_{\tau \geq t_0} |u(\tau)| \quad (9.35)$$

Taking $t_0 = \frac{t}{2}$ and using the bound of Eq.9.34 in Eq.9.35, we have

$$\|x_f(t)\|_2 \leq K e^{-\gamma \frac{t}{2}} \bar{K} + M_2 \sup_{\tau \geq \frac{t}{2}} |u(\tau)| \quad (9.36)$$

Taking the limit of both sides as $t \rightarrow \infty$ and using that $\lim_{t \rightarrow \infty} \sup_{\tau \geq t} u(\tau) = 0$ (this follows directly from $\lim_{t \rightarrow \infty} u(t) = 0$), we finally have

$$\lim_{t \rightarrow \infty} \|x_f(t)\|_2 \leq \lim_{t \rightarrow \infty} (K e^{-\gamma \frac{t}{2}} \bar{K}) + \lim_{t \rightarrow \infty} (M_2 \sup_{\tau \geq \frac{t}{2}} |u(\tau)|) = 0 \quad (9.37)$$

We therefore have that $\lim_{t \rightarrow \infty} \|x_f\|_2 = 0$, and thus, the infinite dimensional closed-loop system is asymptotically stable; this completes the proof of Proposition 9.1.

Remark 9.3: Note that the above formulation is designed on the basis of the slow subsystem only. The evolution of the fast states is neither accounted for in the cost function nor in the state constraints. While the resulting MPC controller enforces closed-loop stability of the infinite dimensional system, a potential drawback of this formulation is the fact that there is no guarantee that the state constraints for the infinite dimensional system will be satisfied. This may occur because the controller design neglects the dynamics of the fast modes, and, the satisfaction of $\chi^{min} \leq (r, x_s) \leq \chi^{max}$ does not guarantee that $\chi^{min} \leq (r, x_s + x_f) = (r, x) \leq \chi^{max}$. So, while the stabilization objective can be achieved independently of the fast subsystem, the additional objective of state constraints satisfaction requires that the MPC controller, even if designed on the basis of the slow subsystem, in some way accounts for the contribution of the fast states to the state of the infinite dimensional system.

We now present a modification to the MPC controller of Eqs.9.27-9.28, and a sufficient condition on the initial condition of the fast subsystem such that the resulting control law, when applied to the infinite dimensional system, achieves both the objectives of stabilization and state constraint satisfaction for the infinite dimensional system. The key idea is to revise (shrink) the state constraint enforcement in the controller design, to compensate for the contribution to the infinite dimensional system state by the fast modes. By invoking input to state boundedness of the fast subsys-

tem, we then derive appropriate bounds on the initial conditions for the fast subsystem which ensure state constraint satisfaction for the closed-loop infinite dimensional system. Proposition 2 below formalizes the input to state boundedness property and Theorem 1 states the necessary conditions for state constraint satisfaction of the infinite dimensional system. To this end, consider the MPC formulation of Eqs.9.27-9.28 with the states constraints χ^{max} , χ^{min} replaced by S^{max} and S^{min} as:

$$\min_u \left[\int_t^{t+T} [q_s \|x_s(\tau)\|_2^2 + |u(\tau)|_R^2] d\tau + F(x_s(t+T)) \right] \quad (9.38)$$

$$\begin{aligned} s.t. \quad \dot{x}_s(\tau) &= \mathcal{A}_s x_s(\tau) + \mathcal{B}_s u(\tau) \\ u(\tau) &\in \mathcal{U} \\ S^{min} &\leq (r, x_s(\tau)) \leq S^{max}, \tau \in [t, t+T] \\ x_s(t+T) &= 0 \end{aligned} \quad (9.39)$$

where $S^{max} \leq \chi^{max} - \alpha$ and $S^{min} \geq \chi^{min} + \alpha$ where α is a positive real number, to be determined later. We denote the set of initial conditions for which the predictive controller of Eqs.9.38-9.39 satisfies the conditions of Assumption 9.1, as Ω'_s .

Proposition 9.2: *Consider the system of Eq.9.11 and the input and state constraints of Eqs.9.12-9.13. There exist real numbers M_1^* and M_2^* such that $|(r, x_f(t))| \leq M_1^* \|x_f(0)\|_2 + M_2^* \bar{u}$, for all $t \geq 0$, where $\bar{u} = \max\{|u^{max}|, |u^{min}|\}$.*

Proof of Proposition 9.2: First, using Hölder's inequality [187], we get

$$\left| \int_0^\pi r(z) f(z, \tau) dz \right| \leq \left(\int_0^\pi r(z)^2 dz \right)^{\frac{1}{2}} \times \left(\int_0^\pi f(z, \tau)^2 dz \right)^{\frac{1}{2}} \quad (9.40)$$

Let $M_3 = \left(\int_0^\pi |r(z)|^2 dz \right)^{\frac{1}{2}}$. Note that since $r(z)$ is a smooth function in $[0, \pi]$, M_3 is a positive real number. Recognizing that the term on the left hand side of the inequality is $|(r, x_f)|$, and the second term on the right hand side is $\|x_f\|_2$, we have that

$$|(r, x_f)| \leq M_3 \|x_f\|_2 \quad (9.41)$$

From Eq.9.33, using that $\sup_{0 \leq \tau \leq t} |u(\tau)| \leq \bar{u}$, it follows that

$$\|x_f(t)\|_2 \leq M_1 \|x_f(0)\|_2 + M_2 \bar{u} \quad (9.42)$$

where $M_1 = K$ and $M_2 = \|\mathcal{B}_f\|_2/\gamma$. Substituting Eq.9.42 into Eq.9.41, we have

$$|(r, x_f(t))| \leq M_1^* \|x_f(0)\|_2 + M_2^* \bar{u} \quad (9.43)$$

where $M_1^* = M_1 M_3$ and $M_2^* = M_2 M_3$. This completes the proof of Proposition 9.2.

Theorem 9.1: *Consider the system of Eq.9.11 subject to the input and state constraints of Eqs.9.12-9.13 under the predictive control law of Eqs.9.38-9.39 for which Assumption 9.1 holds with $x_s(0) \in \Omega'_s$. Then, given that there exists a positive real number δ such that $\chi^{max} - \chi^{min} \geq 2(M_2^* \bar{u} + \delta)$, where M_2^* , $\bar{u}$ were defined in Proposition 9.2, there exist positive real numbers α and β such that if $x_s(0) \in \Omega'_s$ and $\|x_f(0)\|_2 \leq b$, then $x(t) = 0$ is an asymptotically stable solution of the closed-loop system and $\chi^{min} \leq (r, x(t)) \leq \chi^{max}$ for all $t \geq 0$.*

Proof of Theorem 9.1: Asymptotic stability of the closed-loop infinite dimensional system under the control law of Eqs.9.38-9.39 for which Assumption 9.1 holds with $x_s(0) \in \Omega'_s$ can be shown using similar arguments to the ones in the proof of Proposition 9.1. We now consider the problem of constraint satisfaction. First, we assume that there exists a positive real number δ such that $\chi^{max} - \chi^{min} \geq 2(M_2^* \bar{u} + \delta)$, where M_2^* , $\bar{u}$ were defined in Proposition 9.2. We also note that

$$(r, x_s + x_f) = (r, x_s) + (r, x_f) \quad (9.44)$$

and pick $\alpha = M_2^* \bar{u} + \delta$ and $\beta = \delta/M_1^*$. By picking $\|x_f(0)\|_2 \leq \beta$, we have that $-M_1^* \beta - M_2^* \bar{u} \leq (r, x_f(t)) \leq M_1^* \beta + M_2^* \bar{u}$ (Proposition 2), and moreover, $-\delta - M_2^* \bar{u} \leq (r, x_f(t)) \leq \delta + M_2^* \bar{u}$ and thus $-\alpha \leq (r, x_f(\tau)) \leq \alpha$.

Furthermore satisfaction of the constraint of $(r, x_s(\tau)) \leq S^{max} \leq \chi^{max} - \alpha$ implies

$(r, x_s(\tau)) + \alpha \leq \chi^{max}$. This finally implies that $(r, x_s(\tau)) + (r, x_f(\tau)) = (r, x(\tau)) \leq \chi^{max}$ (since $\alpha \geq (r, x_f(\tau))$).

Similarly satisfaction of the constraint of $(r, x_s(\tau)) \geq S^{min} \geq \chi^{min} + \alpha$ implies $(r, x_s(\tau)) - \alpha \geq \chi^{min}$ which implies $(r, x_s(\tau)) + (r, x_f(\tau)) \geq \chi^{min}$ (since $(r, x_f(\tau)) \geq -\alpha$) and finally $(r, x) \geq \chi^{min}$. This proves that for the above choices of α and b , the satisfaction of the constraint of $S^{min} \leq (r, x_s) \leq S^{max}$ implies $\chi^{min} \leq (r, x) \leq \chi^{max}$, which is the state constraint for the infinite dimensional system.

Note that the condition $\chi^{max} - \chi^{min} \geq 2(M_2^* \bar{u} + \delta)$ ensures that for this choice of α , we get $S^{max} \geq S^{min}$, which is necessary for the optimization problem to be feasible. This completes the proof the Theorem 1.

Remark 9.4: Note that the conditions imposed on the initial conditions, and the revision of the state constraints in the controller design, are ‘worst case’ corrections, and are therefore only sufficient conditions. It may happen, for instance, that the controller design of Theorem 1, when implemented in the closed-loop may achieve infinite dimensional state constraint satisfaction, even if the fast modes of the infinite dimensional system do not satisfy the required condition ($\|x_f(0)\|_2 \leq \beta$). Satisfaction of the revised state constraints in the controller design, however, when done according to Theorem 9.1, guarantees state constraint satisfaction for the infinite dimensional system.

Remark 9.5: Note that the set of initial conditions for which a given MPC formulation is guaranteed to be initially and successively feasible is in general a complex function of the inherent dynamics of the system $(\mathcal{A}, \mathcal{B})$, the constraints on the input and the states $(u^{min}, u^{max}, \chi^{min}, \chi^{max}, r(\cdot))$ and the controller parameters (T, Q, R) . Within the MPC framework, it is in general hard to come up with an explicit characterization of this set, and/or compare, for instances the sets Ω_s (of Proposition 9.1) and Ω'_s (of Theorem 9.1) or of the sets we subsequently define and this is not the focus of this

work. What Theorem 1, and other formulations that follow do, is only to provide sufficient conditions that need to be incorporated in the reduced order MPC formulation that, upon satisfaction, guarantee state constraint satisfaction and stabilization for the infinite dimensional system.

Remark 9.6: The problem of implementing various MPC formulations within a hybrid predictive control structure that provides an explicit characterization of the closed-loop stability region for linear systems with input constraints has been addressed in [75]. A similar approach can be utilized to provide an explicit characterization for the closed-loop stability region for finite dimensional input and state constrained systems. The use of the reduced order MPC formulations allows the use of these approaches because the set Ω_s only pertains to the set of initial conditions for a finite number of states, and can be obtained using the approach in [75]. Note that the initial conditions for the fast states only need to satisfy a bound, and hence are also characterized by a finite (in this case, one) number of conditions.

9.3.3 Higher-order MPC formulations

Since the MPC formulation of Theorem 9.1, instead of explicitly accounting for the fast mode dynamics, accounts, in some sense, for the ‘worst case’ dynamics, the formulation may be conservative in terms of restricting the set of initial conditions for which stabilization and state constraint satisfaction for the infinite dimensional system is accomplished. To alleviate this conservatism, we present two MPC formulations which explicitly account for the evolution of the x_f -subsystem.

One way to account for the effect of the fast states on the state constraints of the infinite dimensional system, is incorporating the fast states explicitly into the state constraints equation. The control action in this case is computed by solving the following optimization problem

$$\min_u \left[\int_t^{t+T} \left[q_s \|x_s(\tau)\|_2^2 + |u(\tau)|_R^2 \right] d\tau + F(x_s(t+T)) \right] \quad (9.45)$$

$$\begin{aligned}
s.t. \quad \dot{x}_s(\tau) &= \mathcal{A}_s x_s(\tau) + \mathcal{B}_s u(\tau) \\
\dot{x}_f(\tau) &= \mathcal{A}_f x_f(\tau) + \mathcal{B}_f u(\tau) \\
u(\tau) &\in \mathcal{U} \\
\chi^{min} &\leq (r, x_s(\tau) + x_f(\tau)) \leq \chi^{max} \\
x_s(t+T) &= 0, \tau \in [t, t+T]
\end{aligned} \tag{9.46}$$

Note that for the set of initial conditions defined by Ω'_i , for which this formulation is initially and successively feasible, stabilization and state constraints satisfaction for the infinite dimensional system is achieved. Stabilization of the closed-loop system infinite-dimensional system under the formulation of Eqs.9.45-9.46 can be proved using a similar argument to the one in the proof of proposition 9.1. The implementation of the above controller, however, will require computation of the fast mode dynamics, which can only be done approximately in practice. The key feature of this formulation is that it underscores the fact that even when using a sufficiently high number of modes to simulate the dynamics of the fast modes, the fast modes need not be a part of the cost function, thereby keeping the computation requirement low.

Remark 9.7: The formulation of Eqs.9.45-9.46 represents a more straightforward approach to enforcing state constraint satisfaction, that of incorporating them exactly as constraints in the optimization problem. The drawback of doing this, though, is that the set of initial conditions for which this problem is feasible is infinite dimensional, and impossible to compute or even estimate. The realization that the stability of the slow states is sufficient to enforce stabilization of the infinite dimensional system allows the use of only the slow modes in the cost function and the stability constraint, thereby substantially reducing the computational requirement. In practice, the evolution of fast modes can be accounted for in the state constraint satisfaction by including a sufficiently high, but finite, number of fast modes.

In the formulation of Eq.9.46, the fast modes dynamics have been accounted exactly in the state constraints, which requires that the predictive controller integrates the dynamics of the fast modes, which might be computationally intensive. We now present another formulation that approximates the effect of the fast dynamics by exploiting

the two time-scale separation between the slow and fast subsystems and deriving an approximate model that describes the evolution of the fast subsystem. We define $\epsilon := \frac{|Re\{\lambda_1\}|}{|Re\{\lambda_{m+1}\}|}$ and multiply the x_f -subsystem of Eq.9.23 by ϵ to obtain the following system [37]:

$$\begin{aligned}\frac{dx_s}{dt} &= \mathcal{A}_s x_s + \mathcal{B}_s u \\ \epsilon \frac{dx_f}{dt} &= \mathcal{A}_{f\epsilon} x_f + \epsilon \mathcal{B}_f u\end{aligned}\tag{9.47}$$

where $\mathcal{A}_{f\epsilon}$ is an unbounded differential operator defined as $\mathcal{A}_{f\epsilon} = \epsilon \mathcal{A}_f$. Introducing the fast time scale $\bar{\tau} = \frac{t}{\epsilon}$, the system of Eq.9.47 takes the form:

$$\begin{aligned}\frac{dx_s}{d\bar{\tau}} &= \epsilon(\mathcal{A}_s x_s + \mathcal{B}_s u) \\ \frac{dx_f}{d\bar{\tau}} &= \mathcal{A}_{f\epsilon} x_f + \epsilon \mathcal{B}_f u\end{aligned}\tag{9.48}$$

Setting $\epsilon = 0$, we get

$$\begin{aligned}\frac{d\bar{x}_s}{d\bar{\tau}} &= 0 \\ \frac{d\bar{x}_f}{d\bar{\tau}} &= \mathcal{A}_{f\epsilon} \bar{x}_f\end{aligned}\tag{9.49}$$

From the properties of $\mathcal{A}_{f\epsilon}$, we have that the solution of x_f -subsystem of Eq.9.23 can be approximated by $\bar{x}_f(t) = e^{\mathcal{A}_f t} \bar{x}_f(0)$ (note that from the properties of $\mathcal{A}_f$, it follows that $\mathcal{T}_f(t) = e^{\mathcal{A}_f t}$, [44]- p.153). Based on this two-time-scale analysis, we consider the following MPC formulation:

$$\min_u \left[\int_t^{t+T} [q_s \|x_s(\tau)\|_2^2 + |u(\tau)|_R^2] d\tau + F(x_s(t+T)) \right]\tag{9.50}$$

$$\begin{aligned}s.t. \quad \dot{x}_s(\tau) &= \mathcal{A}_s x_s(\tau) + \mathcal{B}_s u(\tau) \\ u(\tau) &\in \mathcal{U} \\ \bar{S}^{min} &\leq (r, x_s(\tau) + e^{\mathcal{A}_f(\tau-t)} x_f(t)) \leq \bar{S}^{max}\end{aligned}\tag{9.51}$$

where $\tau \in [t, t+T]$, and constraints are given as $\bar{S}^{min} = \chi^{min} + M_2^* \bar{u}$ and $\bar{S}^{max} = \chi^{max} - M_2^* \bar{u}$, where M_2^* , $\bar{u}$ were defined in Proposition 9.2. Once again we denote the set of initial conditions for which the predictive controller of Eqs.9.50-9.51 achieves stabilization of the $x(t) = 0$ solution of the closed-loop infinite dimensional systems by Ω' . Note that, as opposed to Theorem 9.1, the set of initial conditions include the

slow as well as the fast states because the fast states appear explicitly (even though ‘approximately’) in the constraints in the optimization problem. State constraint satisfaction for the infinite dimensional system is achieved by revising the state constraints in the controller formulation by the worst case error (induced due to neglecting the effect of the input on the evolution of the fast modes) in the prediction of the fast state dynamics. We formalize this idea in the following theorem.

Theorem 9.2: *Consider the system of Eq.9.11, the input and state constraints of Eqs.9.12-9.13, under the control law of Eqs.9.50-9.51. Then, if $x(0) \in \Omega'$, the $x(t) = 0$ is an asymptotically stable solution of the closed-loop system of Eqs.9.11-9.50-9.51 and $\chi^{\min} \leq (r, x(t)) \leq \chi^{\max}$ for all $t \geq 0$.*

Proof of Theorem 9.2: The fact that the control law of Eqs.9.50-9.51 achieves stabilization of the closed-loop infinite dimensional system can be proved using similar arguments to the one in the proof of Proposition 9.1. We focus on constraint satisfaction.

Satisfaction of the constraint $(r, x_s + e^{\mathcal{A}_f(\tau-t)}x_f(t)) \leq \bar{S}^{\max} = \chi^{\max} - M_2^*\bar{u}$ implies $(r, x_s(\tau)) + (r, e^{\mathcal{A}_f(\tau-t)}x_f(t)) + M_2^*\bar{u} \leq \chi^{\max}$. Note that $(r, x_s(\tau)) + (r, x_f(\tau)) \leq (r, x_s(\tau)) + (r, e^{\mathcal{A}_f(\tau-t)}x_f(t)) + M_2^*\bar{u}$ (see proof of proposition 2). Hence, $(r, x_s(\tau)) + (r, e^{\mathcal{A}_f(\tau-t)}x_f(t)) + M_2^*\bar{u} \leq \chi^{\max}$ implies $(r, x(\tau)) \leq \chi^{\max}$.

Similarly, satisfaction of the constraint $(r, x_s(\tau) + e^{\mathcal{A}_f(\tau-t)}x_f(t)) \geq \bar{S}^{\min} = \chi^{\min} + M_2^*\bar{u}$ implies $(r, x_s(\tau)) + (r, e^{\mathcal{A}_f(\tau-t)}x_f(t)) - M_2^*\bar{u} \geq \chi^{\min}$. Note, once again that $(r, x_f(\tau)) \geq (r, e^{\mathcal{A}_f(\tau-t)}x_f(t)) - M_2^*\bar{u}$ (see proof of proposition 2). Therefore, $(r, x_s(\tau)) + (r, e^{\mathcal{A}_f(\tau-t)}x_f(t)) - M_2^*\bar{u} \geq \chi^{\min}$ implies $(r, x(\tau)) \geq \chi^{\min}$. This completes the proof of Theorem 2.

9.4 Simulation example

In this section, we demonstrate and compare, through computer simulations, the implementation of the various MPC formulations discussed in the previous section. To this end, we consider the parabolic PDE of Eq.9.1 with $b = 1$, $c = 1.66$, $w = 2$ and two control actuators ($m = 2$) with the following distribution functions $b_i(z) = 1/\mu$ for $z \in [z_{ai} - \mu, z_{ai} + \mu]$ and $b_i(z) = 0$ elsewhere in $[0, \pi]$, where $\mu = 0.005$ is a small positive real number and $z_{a1} = \pi/3$ and $z_{a2} = 2\pi/3$. For these values, it was verified that the operating steady-state, $\bar{x}(z, t) = 0$, is an unstable one. The control objective is to stabilize the state profile at the unstable zero steady-state by manipulating $u_i(t)$ subject to the input and state constraints of Eqs.9.12-9.13 with $u_i^{min} = -3$, $u_i^{max} = 3$, for $i = 1, 2$, $\chi^{min} = -3.0$, $\chi^{max} = 3.0$. $r(z)$ is the state constraint distribution function, chosen to be $r(z) = 1/m$ for $z \in [z_c - \nu, z_c + \nu]$, with $m = 0.0035$, $\nu = 0.0018$ and $z_c = 1.156$, and zero elsewhere. The solution of the eigenvalue problem of the spatial differential operator of Eq.9.14 is:

$$\lambda_j = 1.66 - j^2, \phi_j(z) = \sqrt{\frac{2}{\pi}} \sin(j z), \quad j = 1, \dots, \infty \quad (9.52)$$

For this system, we consider the first two eigenmodes to be the dominant ones. To simplify the presentation of the results, we will work with the amplitudes of the eigenmodes of the PDE of Eq.9.11. Specifically, using standard modal decomposition, we derive the following high-order ODE system that describes the temporal evolution of the amplitudes of the first l eigenmodes:

$$\begin{aligned} \dot{a}_s(t) &= A_s a_s(t) + B_s u(t) \\ \dot{a}_f(t) &= A_f a_f(t) + B_f u(t) \end{aligned} \quad (9.53)$$

where $a_s(t) = [a_1(t) \ a_2(t)]'$, $a_f(t) = [a_3(t) \ a_4(t) \ \dots \ a_l(t)]'$, $a_i(t) \in \mathbb{R}$ is the modal amplitude of the i -th eigenmode, the notation a'_s denotes the transpose of a_s , l is chosen to be 50, $u(t) = [u_1(t) \ u_2(t)]'$, the matrices A_s and A_f are diagonal matrices, given by $A_s = \text{diag}\{\lambda_i\}$, for $i = 1, 2$ and $A_f = \text{diag}\{\lambda_i\}$, for $i = 3, \dots, l$. B_s and B_f are a 2×2 and $(l - 2) \times m$ matrices, respectively whose (i, j) -th element is given

by $B_{ij} = (b_j(z), \phi_i(z))$. Note that $\bar{x}(z, t) = \sum_{i=1}^l a_i(t) \phi_i(z)$, $x_s(t) = a_1(t) \phi_1 + a_2(t) \phi_2$, $x_f(t) = \sum_{i=3}^{50} a_i(t) \phi_i$ and that $(x(t), \phi_i) = a_i(\phi_i, \phi_i)$. Using these projections, the state constraints of Eq.9.13 can be expressed as constraints on the modal amplitudes as follows:

$$\chi^{min} \leq \sum_{i=1}^2 a_i(t) \phi_i(z_c) + \sum_{i=3}^l a_i(t) \phi_i(z_c) \leq \chi^{max} \quad (9.54)$$

We now proceed with the design and implementation of the different predictive control formulations presented in the previous section. In the first scenario, we use the a_s -subsystem in Eq.9.53 as the basis for the predictive controller design (the a_f -subsystem is neglected). For this case, we consider an MPC formulation of the form of Eqs.9.27-9.28 with the following objective function and constraints:

$$\min_u \left[\int_t^{t+T} [q_s |a_s(\tau)|^2 + |u(\tau)|_R^2] d\tau \right] \quad (9.55)$$

$$\begin{aligned} s.t. \quad \dot{a}_s(\tau) &= A_s a_s(\tau) + B_s u(\tau) \\ u_{min} &\leq u_i(\tau) \leq u_{max}, \quad i = 1, 2 \\ \chi^{min} &\leq C_s a_s(\tau) \leq \chi^{max}, \quad \tau \in [t, t+T] \end{aligned} \quad (9.56)$$

where $C_s = [\phi_1(z_c) \ \phi_2(z_c)]$ is a row vector, $q_s = 8.79$, $R = rI$, with $r = 0.01$, and $T = 0.007$. To ensure stability, we also impose a terminal equality constraint of the form $a_s(t+T) = 0$ on the optimization problem. The resulting quadratic program is solved using the MATLAB subroutine QuadProg. The control action is then implemented on the 50-th order model of Eq.9.53. Figure 9.1 shows the closed-loop state under the implementation of the MPC Eqs.9.55-9.56 stabilizes the PDE state at the unstable zero steady-state starting from the initial condition $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 3.15 \sin(3z) + 3.15 \sin(4z)$, By examining Figure 9.3 (solid line), we observe that the integral constraint $R(z_c, t) = \int_0^\pi r(z) \bar{x}(z, t) dz$ violates the lower constraint for some time. The violation of the state constraint is a consequence of neglecting the contribution of the a_f states to the state of the PDE in the MPC formulation.

We now revise the constraints in the previous MPC formulation, and consider the

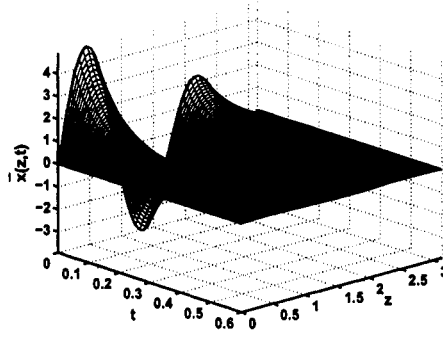


Figure 9.1: Closed-loop state profile under the MPC formulation of Eqs.9.55-9.56 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 3.15 \sin(3z) + 3.15 \sin(4z)$.

following objective function and constraints (analyzed in Theorem 9.1):

$$\min_u \left[\int_t^{t+T} [q_s |a_s(\tau)|^2 + |u(\tau)|_R^2] d\tau \right] \quad (9.57)$$

$$\begin{aligned} s.t. \quad \dot{a}_s(\tau) &= A_s a_s(\tau) + B_s u(\tau) \\ u_{min} &\leq u_i(\tau) \leq u_{max}, \quad i = 1, 2 \\ S^{min} &\leq C_s a_s(\tau) \leq S^{max}, \quad \tau \in [t, t+T] \end{aligned} \quad (9.58)$$

where $C_s = [\phi_1(z_c) \ \phi_2(z_c)]$ is a row vector, and q_s , R , r and T are the same to the ones used in the first scenario, and $S^{max} = 3 - \alpha$ and $S^{min} = -3 + \alpha$. Following the result of Theorem 1, we have verified that $\delta = 1$ satisfies $\chi^{max} - \chi^{min} \geq 2(M_2^* \bar{u} + \delta)$ where $\bar{u} = 3$ and $M_2^* = M_2 M_3 = M_3 \|\mathcal{B}_f\|_2 / \gamma = 0.477$, and pick $\alpha = M_2^* \bar{u} + \delta = 2.43$ and $\beta = \delta / M_1^* = 1.0 / 1.003 = 0.99$

Picking the initial condition $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 0.95 \sin(3z)$ which satisfies $\|x_f(0)\|_2 \leq 0.99$, implementation of the predictive controller of Eqs.9.57-9.58 results in stabilization and satisfaction of the state constraint of the closed-loop system (Figure 9.2 and dashed lines in Figures 9.3-9.4). Note that, since the control action is computed on the basis of the slow states – and since the initial conditions for the slow states are the same as in the previous scenario – the controller implements the same control action as before (i.e., the solid and the dashed lines overlap in Figure 9.4). The closed-loop state profile, however, stays within the constraints because of

appropriate initialization of the fast modes and modification of the state constraint in the predictive controller of Eqs.9.57-9.58.

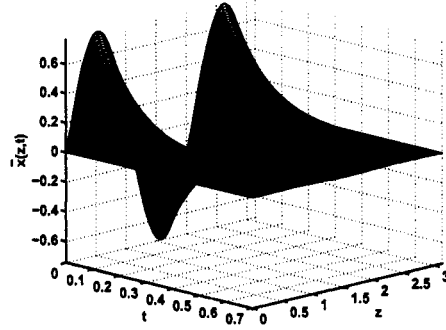


Figure 9.2: Closed-loop state profile under the MPC formulation of Eqs.9.57-9.58 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 0.95 \sin(3z)$.

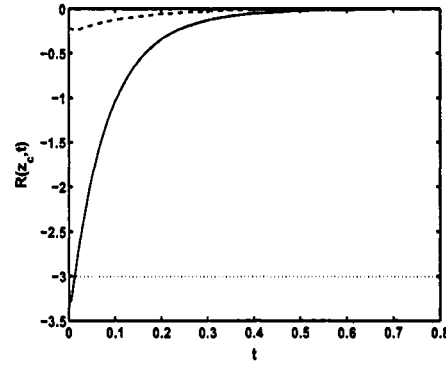


Figure 9.3: $R(z_c, t) = \int_0^\pi r(z) \bar{x}(z, t) dz$ under the MPC formulation of Eqs.9.55-9.56 (solid line) and under the MPC formulation of Eqs.9.57-9.58 (dashed line). The dotted line represents the lower state constraint.

We now consider higher-order MPC formulations. First, we demonstrate the implementation of the MPC formulation of Eqs.9.45-9.46 where the PDE state constraints are exactly accounted for in the controller design. In this case, the objective function and constraints are given by

$$\min_u \left[\int_t^{t+T} \left[q_s |a_s(\tau)|^2 + |u(\tau)|_R^2 \right] d\tau \right] \quad (9.59)$$

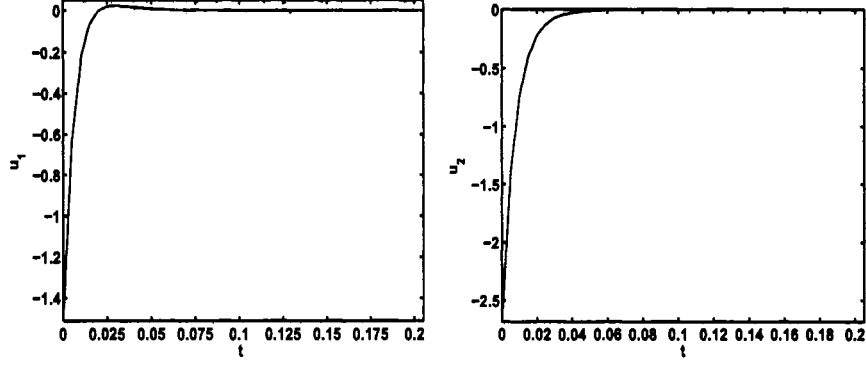


Figure 9.4: Manipulated input profiles for the first and second control actuators applied at $z_{a_1} = \pi/3$ and $z_{a_2} = 2\pi/3$ under the MPC formulation of Eqs.9.55-9.56 (solid line) and under the MPC formulation of Eqs.9.57-9.58 (dashed line); note that the dashed and solid line coincide because of the same initial conditions of the a_s -states.

$$\begin{aligned}
 s.t. \quad \dot{a}_s(\tau) &= A_s a_s(\tau) + B_s u(\tau) \\
 \dot{a}_f(\tau) &= A_f a_s(\tau) + B_f u(\tau) \\
 u_{min} &\leq u_i(\tau) \leq u_{max}, \quad i = 1, 2 \\
 \chi^{min} &\leq C_s a_s(\tau) + C_f a_f(\tau) \leq \chi^{max}
 \end{aligned} \tag{9.60}$$

where $\tau \in [t, t + T]$, $C_f = [\phi_3(z_c) \cdots \phi_{50}(z_c)]$ is a row vector and the MPC tuning parameters have the same values used in the previous two cases. The results are shown in Figure 9.5 and Figure 9.7 (dashed lines), where we see that starting from the initial condition $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 2.8 \sin(3z) + 2.85 \sin(4z)$ the predictive controller of Eqs.9.59-9.60 successfully stabilizes the system at the zero steady-state and the PDE state constraints is satisfied for all times. The corresponding manipulated input profiles are given in Figure 9.8.

Finally, we demonstrate the implementation of the MPC formulation of Eqs.9.50-9.51 (analyzed in Theorem 2). Using the approximation of Eq.9.49 in the formulation of Eqs.9.59-9.60 yields the following objective function and constraints:

$$\min_u \left[\int_t^{t+T} [q_s |a_s(\tau)|^2 + |u(\tau)|_R^2] d\tau \right] \tag{9.61}$$

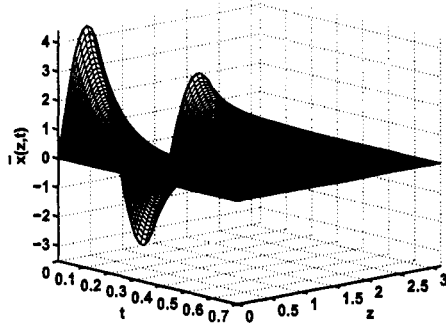


Figure 9.5: Closed-loop state profile under the MPC formulation of Eqs.9.59-9.60 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 2.8 \sin(3z) + 2.85 \sin(4z)$.

$$\begin{aligned}
 s.t. \quad & \dot{a}_s(\tau) = A_s a_s(\tau) + B_s u(\tau) \\
 & u_{min} \leq u_i(\tau) \leq u_{max}, \quad i = 1, 2 \\
 & \bar{S}^{min} \leq C_s a_s(\tau) + C_f \exp(A_f(\tau - t)) a_f(t) \leq \bar{S}^{max}
 \end{aligned} \tag{9.62}$$

where $\tau \in [t, t + T]$. $\bar{S}^{max}$ and $\bar{S}^{min}$ are calculated by using the value $M_2^* \bar{u} = 1.431$, as follows $\bar{S}^{max} = \chi^{max} - M_2^* \bar{u} = 1.569$ and $\bar{S}^{min} = \chi^{min} + M_2^* \bar{u} = -1.569$. The above formulation does not require solving the state evolution equation for the a_f -subsystem at each time step; instead it uses an explicit (approximate) expression, $a_f(\tau) = e^{A_f(\tau-t)} a_f(t)$, to account for the dynamics of the fast subsystem which contribute to the PDE state constraints. The initial conditions in this scenario are picked as ($\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 1.45 \sin(3z) + 1.5 \sin(4z)$); note that for the initial condition $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 2.8 \sin(3z) + 2.85 \sin(4z)$, the formulation of Eqs.9.61-9.62 is not feasible. The receding horizon implementation of the predictive controller implies that for subsequent computations, this expression is used with the updated value of $a_f(t)$. Note, however, that this does not imply that the optimization problem needs to solve the fast mode dynamics; it only means that at the new initial condition, the optimization problem in the predictive controller is solved starting from this updated system state, in line with the standard receding horizon implementation of predictive controllers. The resulting predictive controller, when implemented on the system of Eq.9.1 successfully stabilizes the zero steady-state and enforces PDE state constraints satisfaction (see Figure 9.6 and solid lines Figures

9.7-9.8).

Remark 9.8: As a final note, we want to demonstrate that even if $\bar{x}(z, 0)$ does not violate the state constraints, these constraints can be violated for some time $t \geq 0$. To this end, we pick $\chi^{min} = -0.035$ and $\chi^{max} = 2$, and $u_i^{min} = -10$, $u_i^{max} = 10$, and use the a_s -subsystem in Eq.9.53 as the basis for the predictive controller design (the a_f -subsystem is neglected). For this case, we consider the predictive controller of Eqs.9.55-9.62 where $C_s = [\phi_1(z_c) \ \phi_2(z_c)]$ is a row vector, $q_s = 8.79$, $R = rI$, with $r = 0.01$, and $T = 0.007$. To ensure stability, we impose a terminal equality constraint of the form $a_s(t + T) = 0$ on the optimization problem. The control action is then implemented on the 50-th order model of Eq.9.53. Figure 9.9 shows state constraint profile starting from the initial condition $\bar{x}(z, 0) = 0.04 \sin(z) + 0.0005 \sin(2z) + 0.07 \sin(3z)$, that does not violates state constraints. It is clear that the predictive controller successfully stabilizes the state at the zero steady-state and that the state violates the lower constraint for some time.

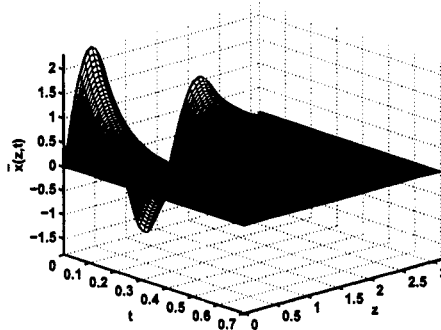


Figure 9.6: Closed-loop state profile under the MPC formulation of Eqs.9.61-9.62 with $\bar{x}(z, 0) = 0.02 \sin(z) + 0.01 \sin(2z) + 1.45 \sin(3z) + 1.5 \sin(4z)$.

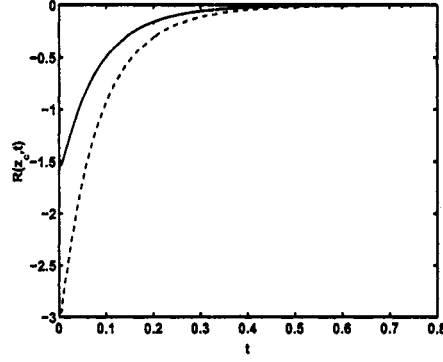


Figure 9.7: $R(z_c, t) = \int_0^\pi r(z)\bar{x}(z, t)dz$ under the MPC formulation of Eqs.9.59-9.60, (dashed line) and under the MPC formulation of Eqs.9.61-9.62 (solid line).

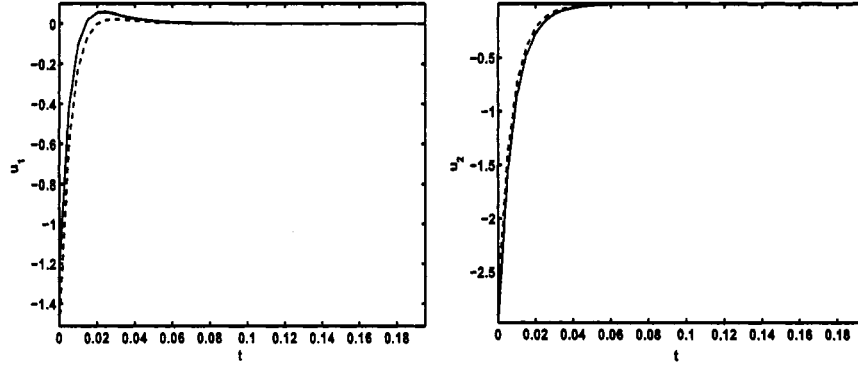


Figure 9.8: Manipulated input profiles for the first and second control actuators applied at $z_{a1} = \pi/3$ and $z_{a1} = 2\pi/3$ under the MPC formulation of Eqs.9.59-9.60 (dashed line) and under the MPC formulation of Eqs.9.61-9.62 (solid line).

9.5 Conclusions

In this chapter, we focused on linear parabolic PDEs with state and control constraints and developed predictive control strategies on the basis of lower-order ODE models obtained through application of appropriate order reduction techniques to the infinite-dimensional system. In order to account for the effect of both the slow and fast modes of the PDE in the cost function of the optimal control problem and the state constraints, we developed various MPC formulations which are distinct with respect to the number

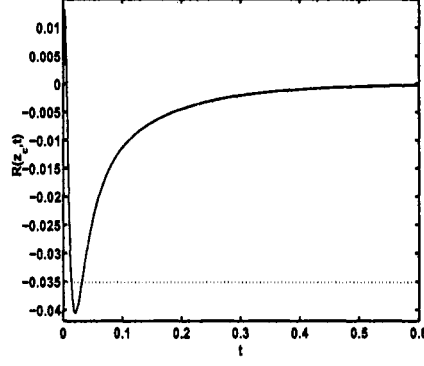


Figure 9.9: $R(z_c, t) = \int_0^\pi r(z)\bar{x}(z, t)dz$ under the MPC formulation of Eqs.9.55-9.56 with $\bar{x}(z, 0) = 0.04 \sin(z) + 0.0005 \sin(2z) + 0.07 \sin(3z)$ and state constraints $-0.035 \leq R(z_c, t) \leq 2$ with $[u_{max}, u_{min}] = [-10, 10]$.

of modal states employed in the cost function and/or state constraints formulation. Namely, the number of modal states utilized in the cost function and number of modal states utilized in the state constraints may differ in the various MPC formulations which affects the performance of the proposed formulation and point out to trade-offs which are necessary to be made with respect to state constraints satisfaction and number of modal states accounted for in the formulation. The developed predictive control algorithms were numerically validated using a linearized diffusion-reaction example.

Chapter 10

Predictive Control of Particle Size Distribution in Particulate Processes

In this chapter, we focus on the development and application of predictive-based strategies for control of particle size distribution (PSD) in continuous and batch particulate processes described by population balance models (PBMs). The control algorithms are designed on the basis of reduced-order models, utilize measurements of principle moments of the PSD, and are tailored to address different control objectives for the continuous and batch processes. For continuous particulate processes, we develop a hybrid predictive control strategy to stabilize a continuous crystallizer at an open-loop unstable steady-state. The hybrid predictive control strategy employs logic-based switching between model predictive control (MPC) and a fall-back bounded controller with a well-defined stability region. The strategy is shown to provide a safety net for the implementation of MPC algorithms with guaranteed stability region. For batch particulate processes, the control objective is to achieve a final PSD with desired characteristics subject to both manipulated input and product quality constraints. An optimization-based predictive control strategy that incorporates these constraints ex-

plicitly in the controller design is formulated and applied to a seeded batch cooling crystallizer example. The strategy is shown to be able to reduce the total volume of the fines by 13.4% compared to a linear cooling strategy, and is shown to be robust with respect to modeling errors.

10.1 Introduction

Particulate processes are prevalent in a number of process industries including agricultural, chemical, food, minerals, and pharmaceuticals. By some estimates, 60% of the products in the chemical industry are manufactured as particulates with an additional 20% using powders as ingredients. Examples of particulate processes include the crystallization of proteins for pharmaceutical applications, the emulsion polymerization reactors for the production of latex, and the titania powder aerosol reactors used in the production of white pigments. One of the key attributes of particulate systems is the co-presence of a continuous phase and a dispersed phase, which leads to the occurrence of physico-chemical phenomena such as particle nucleation, growth, coagulation, and breakage which are absent in homogeneous processes and lead to a distributed characterization of the physical and chemical properties of the particulate product such as particle size, shape, morphology, porosity, molecular weight, etc.

It is now well understood that the physico-chemical and mechanical properties of materials made with particulates are strongly dependent on the characteristics of the corresponding particle size distribution (PSD). For example, a nearly mono-disperse PSD is required for titania pigments to obtain the maximum hiding power per unit mass. Also, in coatings the product's composition, molecular weight and particle size distributions often need to be maintained in specific ranges to ensure the coating has a desired level of film formation, film strength, and gloss. In all of these instances, the PSD provides the critical link between the product quality indices and the operating process variables; and, therefore, the ability to effectively manipulate the PSD is essen-

tial for our ability to control the end product quality in these processes. In this light, the problem of synthesizing and implementing high-performance model-based feedback control systems on particulate processes to achieve PSDs with desired characteristics has significant industrial value.

The mathematical models of particulate processes are typically obtained through the application of population, material and energy balances and consist of systems of nonlinear partial integro-differential equations that describe the evolution of the PSD, coupled with systems of nonlinear ordinary differential equations (ODEs) that describe the evolution of the state variables of the continuous phase. There is an extensive literature on population balance modeling, numerical solution, and dynamical analysis of particulate processes, see, for example, [181, 166, 137, 81, 86, 175, 95, 118, 103, 183, 177]; see also [40] for further details and references. Early work on control of particulate processes focused mainly on the understanding of fundamental control-theoretic properties of PBMs [205], and the application of conventional control schemes to crystallizers and emulsion polymerization processes [206, 189, 54], and the references therein. More recently, the realization that PBMs – owing to their infinite-dimensional nature – cannot be used directly for the synthesis of practically implementable controllers, has motivated significant research work on the development of a general order reduction procedure, based on combination of the method of weighted residuals and approximate inertial manifolds, which allows deriving low-order ODE approximations that capture the dominant dynamics of particulate processes and can, therefore, serve as an appropriate basis for the design of low-order controllers that can be readily implemented in practice [34]. This approach subsequently laid the foundation for the development of a systematic framework for solving a number of important control problems for particulate processes, including the problem of dealing with the highly nonlinear behavior, e.g., owing to complex growth, nucleation, agglomeration and breakage mechanisms, and the Arrhenius dependence of nucleation laws on solute concentration in crystallizers [34], the problem of model uncertainty [35], and the problem of control under

actuator constraints [64].

In the operation of particulate processes, constraints typically arise due to physical limitations on the capacity of control actuators and/or desired restrictions on the process state variables, such as temperature and certain properties of the PSD (e.g., crystal concentration and total particle size), in order to meet some safety or product quality requirements. In current industrial practice, the achievement of optimal performance, subject to input and state constraints, relies to a large extent on the use of MPC policies which are well known for their ability to handle multi-variable interactions, constraints, and optimization requirements, all in a consistent, systematic manner. Unlike open-loop model-based optimal control policies (where the optimal operating conditions are calculated off-line), in MPC, the control action is computed by solving repeatedly, on-line, a constrained optimization problem at each sampling time. Owing to this, MPC has the ability to suppress the influence of external disturbances and tolerate model inaccuracies (because of the use of feedback) and force the system to follow the optimal trajectory that respects constraints on the operating conditions.

In this chapter, we focus on the development and application of predictive algorithms for control of PSDs in continuous and batch particulate processes described by PBMs. The control algorithms are designed on the basis of finite-dimensional models that capture the dominant dynamics of the particulate processes and are tailored to address different control objectives for the batch and continuous processes. The controllers utilize real-time measurements of the principal moments of the PSD (such measurements, for example, can be obtained from PSD measurements made by light scattering techniques as in the case of crystallization processes), as well as measurements of the process temperature and of the concentrations of the continuous-phase species. For continuous particulate processes, we consider the control objective of asymptotic stabilization under constraints and develop a hybrid predictive control methodology that employs logic-based switching between MPC and a fall-back bounded controller

with a well-defined stability region. The hybrid predictive control strategy provides a safety net for the implementation of MPC algorithms to particulate processes with guaranteed stability regions. The strategy is successfully used to stabilize a continuous crystallizer at an open-loop unstable steady-state. For batch particulate processes, the control objective is to achieve PSD with desired characteristics subject to both control and product quality constraints. An optimization-based predictive control strategy that incorporates these constraints explicitly in the controller design is formulated and applied to a seeded batch crystallizer of potassium sulfate crystals. The strategy is shown to be able to reduce the volume of fines by 13.4% comparing to linear cooling strategy, and to possess a robustness margin with respect to modeling errors.

10.2 Predictive control of continuous particulate processes

10.2.1 A continuous crystallizer: modeling and dynamics

Crystallization is a particulate process which is widely used in industry for the production of many products including fertilizers, proteins, and pesticides. The fact that the shape of the crystal-size distribution influences significantly the necessary liquid-solid separation, as well as the properties of the product, implies that crystallization requires a population balance in order to be accurately described, analyzed, and controlled. Continuous crystallizers typically exhibit highly oscillatory behavior which suggests the use of feedback control to ensure stable operation and attain a crystal-size distribution with desired characteristics. Under the assumptions of isothermal operation, constant volume, mixed suspension, nucleation of crystals of infinitesimal size, and mixed product removal, a dynamic model for a continuous crystallizer can be derived from a population balance for the particle phase and a mass balance for the

solute concentration of the following form [121, 103]:

$$\begin{aligned}\frac{\partial n}{\partial \bar{t}} &= -\frac{\partial(R(\bar{t})n)}{\partial r} - \frac{n}{\tau} + \delta(r-0)Q(\bar{t}) \\ \frac{dc}{d\bar{t}} &= \frac{(c_0 - \rho)}{\bar{\epsilon}\tau} + \frac{(\rho - c)}{\tau} + \frac{(\rho - c)}{\bar{\epsilon}} \frac{d\bar{\epsilon}}{d\bar{t}}\end{aligned}\quad (10.1)$$

where $n(r, \bar{t})$ is the density of crystals of radius $r \in [0, \infty)$ at time $\bar{t}$ in the suspension, τ is the residence time, c is the solute concentration in the crystallizer, c_0 is the solute concentration in the feed, and $\bar{\epsilon} = 1 - \int_0^\infty n(r, \bar{t}) \frac{4}{3}\pi r^3 dr$ is the volume of liquid per unit volume of suspension. $R(\bar{t})$ is the growth rate, $\delta(r-0)$ is the standard Dirac function, and $Q(\bar{t})$ is the nucleation rate. The term $\delta(r-0)Q(\bar{t})$ accounts for the production of crystals of infinitesimal (zero) size via nucleation. $R(\bar{t})$ and $Q(\bar{t})$ are assumed to follow McCabe's law and Volmer's nucleation law, respectively:

$$\begin{aligned}R(\bar{t}) &= k_1(c - c_s) \\ Q(\bar{t}) &= \bar{\epsilon}k_2 \exp \left[-\frac{k_3}{\left(\frac{c}{c_s} - 1\right)^2} \right]\end{aligned}\quad (10.2)$$

where k_1 , k_2 , and k_3 are constants and c_s is the concentration of solute at saturation.

To study the dynamic behavior of the crystallizer in question, a second-order accurate finite-difference spatial discretization scheme with 1,000 discretization points was used to obtain the solution of the system of Eqs.10.1-10.2 (simulations of the system using more discretization points led to identical results). The values of the process parameters used in the simulations can be found in [34].

The solid lines in Figure 10.1 show the open-loop profiles of the total crystal concentration and the solute concentration obtained by solving the distributed parameter model of Eqs.10.1-10.2. It is clear that the crystallizer exhibits highly oscillatory behavior, which is the result of the interplay between growth and nucleation caused by the relative nonlinearity of the nucleation rate as compared to the growth rate (compare the nonlinear dependence of $Q(\bar{t})$ and $R(\bar{t})$ on c in Eq.10.2). To establish that the dynamics of the crystallizer are characterized by a small number of degrees of freedom, the method of moments is applied to the system of Eqs.10.1-10.2 to derive a

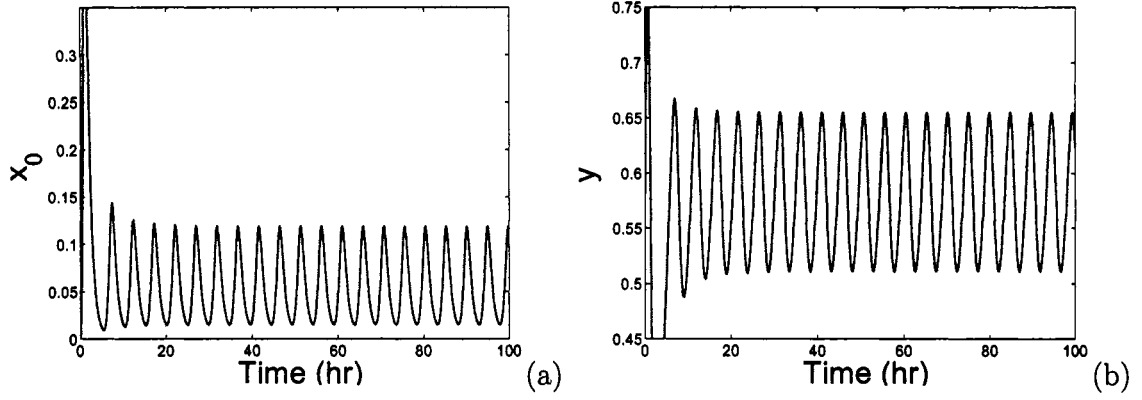


Figure 10.1: Open-loop profiles of (a) crystal concentration and (b) solute concentration obtained from the distributed parameter model.

reduced-order ODE model. Because the nucleation and growth rates are assumed to be independent of particle size, this allows closure of the moments equations, which results in the reduced-order moments model being an exact replication of the evolution of the PBM model. It is noted that the method of moments has been extensively used in the past to analyze the dynamics of particulate processes, e.g., [99, 169].

Defining the j th moment of $n(r, \bar{t})$ as:

$$\mu_j = \int_0^\infty r^j n(r, \bar{t}) dr, \quad j = 0, \dots, \quad (10.3)$$

multiplying the population balance in Eq.10.1 by r^j , and integrating over all particle sizes, the following system of infinite ordinary differential equations, which describes the rate of change of the moments of the particle size distribution and the solute

concentration, is obtained:

$$\begin{aligned}
\frac{d\mu_0}{d\bar{t}} &= -\frac{\mu_0}{\tau} + \left(1 - \frac{4}{3}\pi\mu_3\right) k_2 e^{-\frac{k_3}{\left(\frac{c}{c_s} - 1\right)^2}} \\
\frac{d\mu_1}{d\bar{t}} &= -\frac{\mu_1}{\tau} + \nu k_1 (c - c_s) \mu_0 \\
\frac{d\mu_2}{d\bar{t}} &= -\frac{\mu_2}{\tau} + \nu k_1 (c - c_s) \mu_1 \\
\frac{d\mu_3}{d\bar{t}} &= -\frac{\mu_3}{\tau} + \nu k_1 (c - c_s) \mu_2 \\
\frac{d\mu_j}{d\bar{t}} &= -\frac{\mu_j}{\tau} + \nu k_1 (c - c_s) \mu_{j-1}, \quad j = 4, 5, 6, \dots, \\
\frac{dc}{d\bar{t}} &= \frac{c_0 - c - 4\pi k_1 \tau (c - c_s) \mu_2 (\rho - c)}{\tau \left(1 - \frac{4}{3}\pi\mu_3\right)}
\end{aligned} \tag{10.4}$$

On the basis of the system of Eq.10.4, it is clear that the moments of order four and higher do not affect those of order three and lower, and moreover, the state of the infinite dimensional system is bounded when μ_3 and c are bounded, and it converges to a globally exponentially stable equilibrium point when $\lim_{t \rightarrow \infty} \mu_3 = c_1$ and $\lim_{t \rightarrow \infty} c = c_2$, where c_1, c_2 are constants. This implies that the dominant dynamics of the process of Eq.10.1 can be adequately captured by the fifth-order moments model which includes the dynamics of the first four moments and those of the solute concentration. Furthermore, when the following set of dimensionless variables and parameters is introduced:

$$\begin{aligned}
x_0 &= 8\pi\sigma^3\mu_0, \quad x_1 = 8\pi\sigma^2\mu_1, \quad x_2 = 4\pi\sigma\mu_2, \quad x_3 = \frac{4}{3}\pi\mu_3, \dots, \\
t &= \frac{\bar{t}}{\tau}, \quad \sigma = k_1\tau(c_{0s} - c_s), \quad Da = 8\pi\sigma^3k_2\tau, \\
F &= \frac{k_3c_s^2}{(c_{0s} - c_s)^2}, \quad \alpha = \frac{(\rho - c_s)}{(c_{0s} - c_s)}, \quad y = \frac{(c - c_s)}{(c_{0s} - c_s)}, \quad u = \frac{(c_0 - c_{0s})}{(c_{0s} - c_s)}
\end{aligned} \tag{10.5}$$

The resulting moments model takes the form:

$$\begin{aligned}
\frac{dx_0}{dt} &= -x_0 + (1 - x_3)Da \exp\left(\frac{-F}{y^2}\right) \\
\frac{dx_1}{dt} &= -x_1 + yx_0 \\
\frac{dx_2}{dt} &= -x_2 + yx_1 \\
\frac{dx_3}{dt} &= -x_3 + yx_2 \\
\frac{dy}{dt} &= \frac{1 - y - (\alpha - y)yx_2}{1 - x_3} + \frac{u}{1 - x_3}
\end{aligned} \tag{10.6}$$

where x_i , $i = 0, 1, 2, 3$, are dimensionless moments of the crystal size distribution, y is dimensionless concentration of the solute in the crystallizer, and u is a dimensionless concentration of the solute in the feed (the reader may refer to [64] for a detailed derivation of the moments model, and to [40] for further results and references in this area).

The stability properties of the fifth-order model of Eq.10.6 have been studied thoroughly in [103], see also [121], where it is shown that the global phase space of this model has a unique unstable steady-state surrounded by a stable periodic orbit, and that the linearization of the system of Eq.10.1 around the unstable steady-state includes two isolated complex conjugate eigenvalues with a positive real part.

10.2.2 Hybrid predictive controller design

Having obtained a low-order ODE model that captures the dominant dynamics of the continuous crystallizer, we proceed in this section to address the controller synthesis problem on the basis of the low-order model. The control objective is to stabilize the crystallizer at an unstable steady-state (which corresponds to a desired PSD) using constrained control action. MPC is a popular method for handling constraints within an optimal control setting. In MPC, the control action is obtained by solving repeatedly, on-line, a finite-horizon constrained open-loop optimal control problem. When the system is linear, the cost quadratic, and the constraints convex, the MPC op-

timization problem reduces to a quadratic program for which efficient software exists and, consequently, a number of control-relevant issues have been explored, including issues of closed-loop stability, performance, implementation and constraint satisfaction. The crystallizer, however, exhibits highly nonlinear behavior that must be accounted for when designing the controller. While several nonlinear model predictive control (NMPC) schemes have been proposed in the literature (see [141] for a survey of results in this area), the practical implementation of MPC is limited by: (1) the computational difficulties of solving a nonlinear (typically nonconvex) optimization problem at each time step, and (2) the difficulty of characterizing, *a priori*, the set of initial conditions starting from where a given NMPC controller is guaranteed to be feasible and/or stabilize the closed-loop nonlinear system.

To overcome these difficulties, we have recently developed [62] a hybrid predictive control structure that provides a safety net for the implementation of predictive control algorithms. The central idea is to use a bounded analytical nonlinear controller, with an explicitly characterized stability region, as a fall-back controller, and embed the operation of MPC within its stability region. In the event that the given predictive controller (which can be based on linear or nonlinear models) is unable to stabilize the closed-loop system (e.g., due to failure of the optimization algorithm, poor choice of the initial condition, insufficient horizon length, etc.), supervisory switching from MPC to the bounded controller safeguards closed-loop stability.

In order to proceed with the hybrid predictive controller design, we initially re-write the moments model of Eq.10.6 in a more compact form:

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g(\tilde{x}(t))\tilde{u}(t), \quad |\tilde{u}| \leq u_{max} \quad (10.7)$$

where $\tilde{x} = [\tilde{x}_0 \ \tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3 \ \tilde{y}]'$, $\tilde{x}_i = x_i - x_i^s$, $i = 0, 1, 2, 3$, $\tilde{y} = y - y^s$, $\tilde{u} = u - u^s$, $u_{max} \geq 0$ denotes the bound on the manipulated input, the superscript at x_i^s refers to the unstable steady-state at which we would like to asymptotically stabilize the system.

In order to provide the necessary background for our main results in section 10.2.3,

we will briefly review in the remainder of this section the design procedure for, and the stability properties of, both the bounded and model predictive controllers, which constitute the basic components of our hybrid control scheme. For clarity of presentation, we will focus only on the state feedback problem where measurements of $x(t)$ are assumed to be available for all t .

Bounded Lyapunov-based control

Consider the system of Eq.10.7, for which a control Lyapunov function (CLF), $V(\tilde{x})$, is available, for more details on the existence and construction of CLFs, see [80]. Using the control Lyapunov function, we construct, using the results in [127], see also [66, 69], the following continuous bounded control law:

$$u(\tilde{x}) = -k(\tilde{x})L_gV(\tilde{x}) := b(\tilde{x}) \quad (10.8)$$

where

$$k(\tilde{x}) = \begin{cases} \frac{L_fV(\tilde{x}) + \sqrt{(L_fV(\tilde{x}))^2 + (u_{max}L_gV(\tilde{x}))^4}}{(L_gV(\tilde{x}))^2 \left[1 + \sqrt{1 + (u_{max}L_gV(\tilde{x}))^2} \right]}, & L_gV(\tilde{x}) \neq 0 \\ 0, & L_gV(\tilde{x}) = 0 \end{cases} \quad (10.9)$$

where $L_fV(\tilde{x}) = \frac{\partial V(\tilde{x})}{\partial \tilde{x}}f(\tilde{x})$, and $L_gV(\tilde{x}) = \frac{\partial V(\tilde{x})}{\partial \tilde{x}}g(\tilde{x})$. An estimate of the constrained stability region of the above controller can be obtained using the level sets of V , i.e.,

$$\Omega = \{\tilde{x} \in \mathbb{R}^n : V(\tilde{x}) \leq c^{max}\} \quad (10.10)$$

where $c^{max} > 0$ is the largest number for which every nonzero element of Ω is fully contained in the set:

$$\Phi = \{\tilde{x} \in \mathbb{R}^n : L_fV(\tilde{x}) < u_{max}|L_gV(\tilde{x})|\} \quad (10.11)$$

Model predictive control

In this section, we consider model predictive control of the system under control constraints described by Eq.10.7. In the literature, several MPC formulations are currently

available. For the sake of a concrete illustration, we briefly describe here the traditional formulation (we note that any other MPC formulation can be used; see [62] for further details on this issue). For this case, the control action in MPC at state $\tilde{x}$ and time t is conventionally obtained by solving, on-line, a finite horizon optimal control problem of the form

$$P(\tilde{x}, t) : \min\{J(\tilde{x}, t, \tilde{u}(\cdot)) | \tilde{u}(\cdot) \in S\}, \quad s.t. \quad \dot{\tilde{x}} = f(\tilde{x}) + g(\tilde{x})\tilde{u} \quad (10.12)$$

where $S = S(t, T)$ is the family of piecewise continuous functions, with period Δ , mapping $[t, t + T]$ into $\mathcal{U} := \{\tilde{u} \in \mathbb{R} : |\tilde{u}| \leq \square_{\mathbb{R}^m}\}$ and T is the specified horizon. The constraint in Eq.10.12 is a nonlinear model describing the time evolution of the states $\tilde{x}$. A control $\tilde{u}(\cdot)$ in S is characterized by the sequence $\{\tilde{u}[k]\}$ where $\tilde{u}[k] := \tilde{u}(k\Delta)$, and satisfies $\tilde{u}(t) = \tilde{u}[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. The performance index is given by

$$J(\tilde{x}, t, \tilde{u}(\cdot)) = \int_t^{t+T} [\|\tilde{x}^u(s; \tilde{x}, t)\|_Q^2 + \|\tilde{u}(s)\|_R^2] ds + F(\tilde{x}(t+T)) \quad (10.13)$$

where $\|\cdot\|_Q$ refers to the weighted norm, defined by $\|\tilde{x}\|_Q^2 = \tilde{x}'Q\tilde{x}$ for all $\tilde{x} \in \mathbb{R}^n$, Q and R are strictly positive-definite, symmetric matrices and $\tilde{x}^u(s; \tilde{x}, t)$ denotes the solution of Eq.10.7, due to control $\tilde{u}$, with initial state $\tilde{x}$ at time t and $F(\cdot)$ denotes the terminal penalty. The minimizing control $\tilde{u}^0(\cdot) \in S$ is then applied to the plant over the interval $[k\Delta, (k+1)\Delta]$ and the procedure is repeated indefinitely. This defines an implicit model predictive control law

$$M(\tilde{x}) = \operatorname{argmin}(J(\tilde{x}, t, \tilde{u}(\cdot))) = \tilde{u}^0(t; \tilde{x}, t) \quad (10.14)$$

A hybrid predictive control strategy: switching logic design

In this section, we describe a switching strategy that brings together the MPC and bounded controllers in a way that guarantees asymptotic closed-loop stability and provides a safety net for the implementation of MPC. To this end, consider the constrained nonlinear system of Eq.10.7, with any initial condition $\tilde{x}(0) \in \Omega$, where Ω was defined

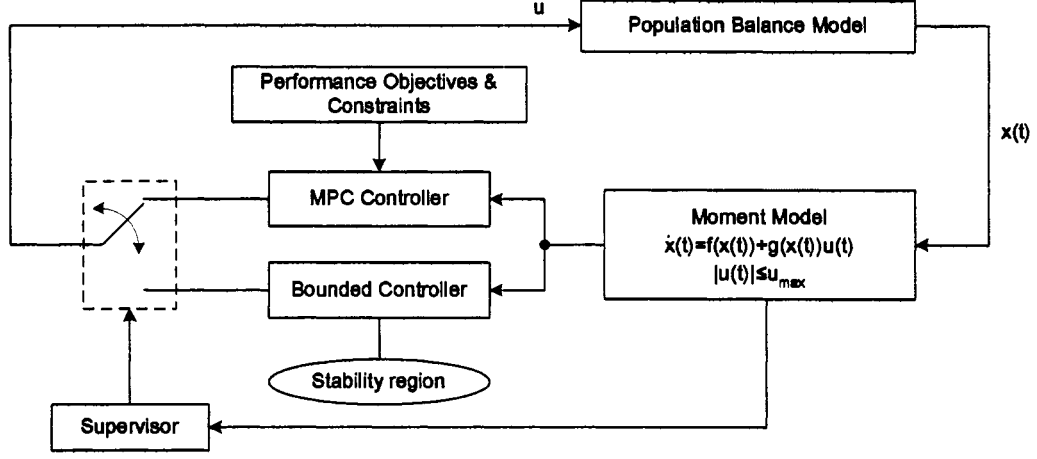


Figure 10.2: Closed-loop system under hybrid predictive control.

in Eq.10.10, under the model predictive controller of Eqs.10.12-10.14. Also let $\bar{T} \geq 0$ be the earliest time for which either the closed-loop state, under MPC, satisfies

$$L_f V(\tilde{x}(\bar{T})) + L_g V(\tilde{x}(\bar{T}))M(\tilde{x}(\bar{T})) \geq 0 \quad (10.15)$$

or the MPC algorithm fails to prescribe any control move. Then, the switching rule given by

$$\tilde{u}(t) = \begin{cases} M(\tilde{x}(t)), & 0 \leq t < \bar{T} \\ b(\tilde{x}(t)), & t \geq \bar{T} \end{cases} \quad (10.16)$$

guarantees that the origin of the switched closed-loop system of Eqs.10.7-10.16 is asymptotically stable (see [62] for a detailed proof as well as possible extensions of the switching scheme).

The hybrid predictive control structure consists of the predictive controller, the bounded nonlinear controller with its estimated region of closed-loop stability, and a high-level supervisor that orchestrates the switching between the two controllers. A schematic representation of the hybrid predictive control structure is shown in Figure 10.2. The implementation procedure is outlined below:

1. Consider the population balance model of Eqs.10.1-10.2 and use the model reduction procedure, based on the method of moments, to derive the finite dimensional

ODE model of Eq.10.6 that captures the dominant dynamics of the continuous crystallizer.

2. Given the constraints on the manipulated input and an appropriate CLF, design the bounded controller, on the basis of the system of Eq.10.6, using Eqs.10.8-10.9, and compute the stability region using Eqs.10.10-10.11.
3. Given the performance objective and constraints, construct the MPC optimization algorithm, on the basis of the system of Eq.10.6, and choose the MPC parameters.
4. Initialize the closed-loop system under MPC, with the initial condition $x(0)$ belonging to the set Ω .
5. Monitor the temporal evolution of the closed-loop trajectory of x (by checking Eq.10.15 at each time) and denote the earliest time that either Eq.10.15 holds or the MPC algorithm prescribes no control move as $\bar{T}$.
6. If such a $\bar{T}$ exists, discontinue MPC implementation, switch to the bounded controller and implement it for all future times.

10.2.3 Application to control of PSD in a continuous crystallizer

In this section, we demonstrate the application of the hybrid predictive control strategy to the continuous crystallizer of Eqs.10.1-10.2. The control objective is to suppress the oscillatory behavior of the crystallizer and stabilize it at an unstable steady-state that corresponds to a desired PSD by manipulating the inlet solute concentration. To achieve this objective, we assume that the first four moments, as well as the solute concentration, can be measured on line. Following the proposed methodology, we initially use the moments model of Eq.10.6 to design the controllers. The values of the dimensionless model parameters in Eq.10.6 are chosen to be: $F = 3.0$, $\alpha = 40.0$ and $Da = 200.0$. The dimensionless solute feed concentration, u , is subject to the

constraints: $-1 \leq u \leq 1$ (which correspond to the following constraint on the inlet solute concentration; $980 \text{ kg/m}^3 \leq c_0 \leq 1000 \text{ kg/m}^3$). The desired steady-state is $x^s = [x_0^s \ x_1^s \ x_2^s \ x_3^s \ y^s]' = [0.065 \ 0.040 \ 0.024 \ 0.015 \ 0.612]'$, and $u^s = 0.2$.

To facilitate the design of the bounded controller and construction of the CLF, we initially re-write the moments model of Eq.10.6 in deviation variable form – thus translating the steady-state to the origin – to obtain the system of Eq.10.7 which we transform into the normal form. To this end, we define the auxiliary output variable, $\bar{y} = h(x) = \tilde{x}_0$, and introduce the invertible coordinate transformation: $[\xi' \ \eta']' = \Pi(x) = [\tilde{x}_0 \ f_1(\tilde{x}) \ \tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3]'$, where $\xi = [\xi_1 \ \xi_2]' = [\tilde{x}_0 \ f_1(\tilde{x})]'$, $\bar{y} = \xi_1$, $f_1(\tilde{x}) = -\tilde{x}_0 + (1 - \tilde{x}_3)Da \exp(-F/\bar{y}^2)$, and $\eta = [\eta_1 \ \eta_2 \ \eta_3]' = [\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3]'$. The state-space description of the system in the transformed coordinates takes the form:

$$\begin{aligned} \dot{\xi} &= A\xi + b l(\xi, \eta) + b \alpha(\xi, \eta)u \\ \dot{\eta} &= \Psi(\eta, \xi) \end{aligned} \quad (10.17)$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b = [0 \ 1]'$, $l(\xi, \eta) = L_f^2 h(\Pi^{-1}(\xi, \eta))$ is the second-order Lie derivative of the scalar function, $h(\cdot)$, along the vector field $f(\cdot)$, and $\alpha(\xi, \eta) = L_g L_f h(\Pi^{-1}(\xi, \eta))$ is the mixed Lie derivative. The forms of $f(\cdot)$ and $g(\cdot)$ can be obtained by re-writing the system of Eq.10.6 in the form of Eq.10.7, and are omitted for brevity.

The partially-linear ξ -subsystem in Eq.10.17 is used to design a bounded controller that stabilizes the full interconnected system of Eq.10.17 and, consequently, the original system of Eq.10.6. For this purpose, a quadratic function of the form, $V_\xi = \xi' P \xi$, is used as a CLF in the controller synthesis formula of Eqs.10.8-10.9, where the positive-definite matrix, P , is chosen to satisfy the Riccati matrix equality: $A'P + PA - P b b' P = -\bar{Q}$ where $\bar{Q}$ is a positive-definite matrix. An estimate of the region of constrained closed-loop stability for the full system is obtained by defining a composite Lyapunov function of the form $V_c = V_\xi + V_\eta$, where $V_\eta = \eta' P_\eta \eta$ and P_η is a positive-definite matrix, and choosing a level set of V_c , Ω_c , for which $\dot{V}_c < 0$ for all x in Ω_c . The two-dimensional projections of the stability region are shown in Figure 10.3 for all possible

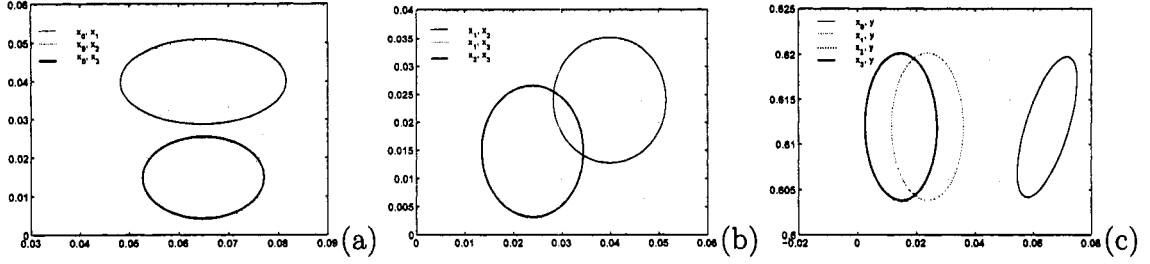


Figure 10.3: Two dimensional projections of the stability region for the 10 distinct combinations of the states of the reduced order system of Eq.10.6.

combinations of the system states.

In designing the predictive controller, a linear MPC formulation, with a terminal equality constraint of the form $x(t + T) = 0$, is chosen (based on the linearization of the reduced order model of Eq.10.6 around the unstable equilibrium point). The parameters in the objective function of Eq.10.13 are taken to be: $Q = qI$, with $q = 1$, $R = rI$, with $r = 1.0$, and $F = 0$. We also choose a horizon length of $T = 0.25$ in implementing the predictive controller. The resulting quadratic program is solved using the MATLAB subroutine QuadProg, and the full nonlinear closed-loop system is integrated using finite-differences.

In the first set of simulation runs, we tested the ability of the predictive controller to stabilize the crystallizer starting from the initial condition, $x(0) = [0.066 \ 0.041 \ 0.025 \ 0.015 \ 0.560]'$. The result is shown by the solid lines in Figure 10.4(a)–(e) where it is seen that the predictive controller, with a horizon length of $T = 0.25$, is able to stabilize the closed-loop system at the desired equilibrium point. Starting from the initial condition $x(0) = [0.033 \ 0.020 \ 0.013 \ 0.0075 \ 0.570]'$, however, the predictive controller yields no feasible solution. If the terminal equality constraint is removed, to make MPC feasible, we see from the dashed lines in Figure 10.4(a)–(e) that the resulting control action cannot stabilize the closed-loop system, and leads to a stable limit cycle. Figure 10.5(a) shows the resulting sustained oscillations in the PSD under MPC without terminal constraints. On the other hand, when the hybrid predictive controller is implemented, the

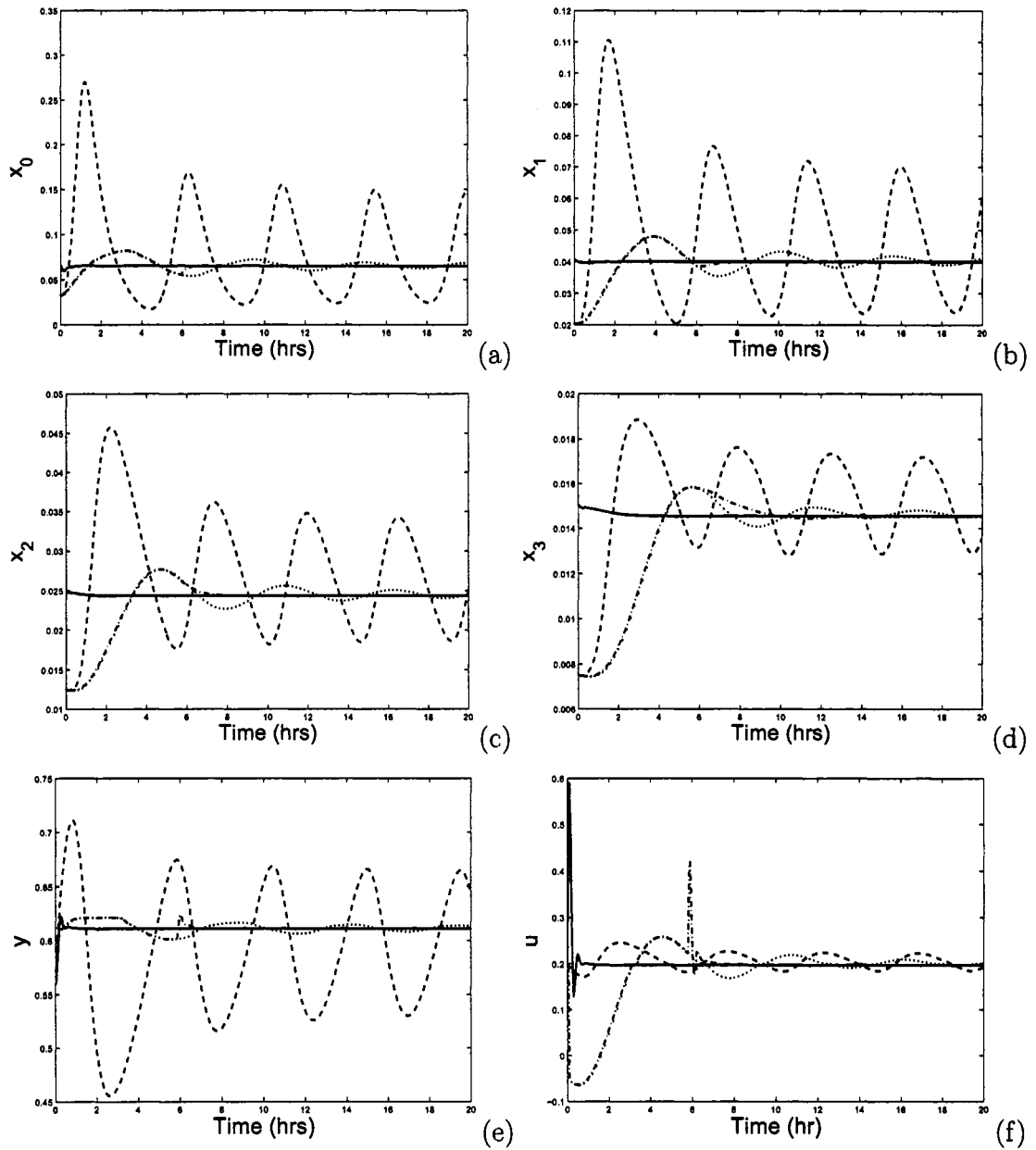


Figure 10.4: Continuous crystallizer example: closed-loop profiles of the dimensionless crystallizer moments (a)-(d), the solute concentration in the crystallizer (e) and the manipulated input (f) under MPC with stability constraints (solid lines), under the bounded controller (dotted lines), and using the hybrid predictive controller (dash-dotted lines).

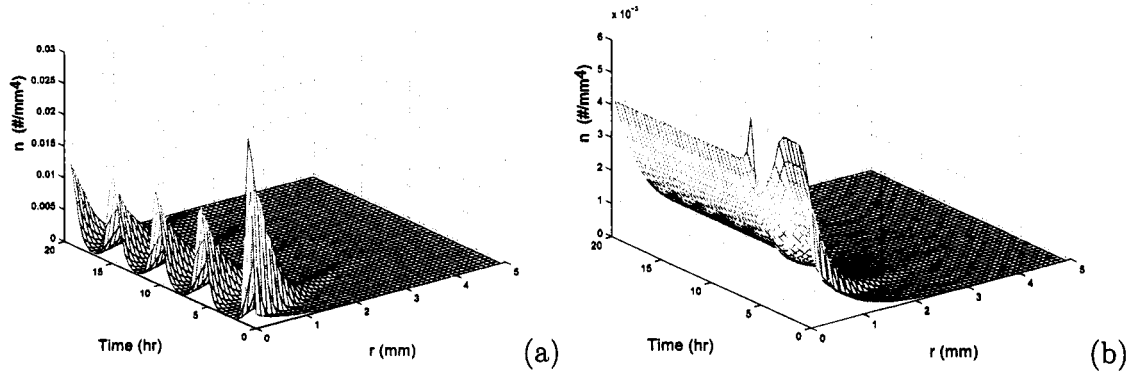


Figure 10.5: Closed-loop profiles of the PSD: (a) under MPC without terminal constraints; (b) under the hybrid predictive controller.

supervisor detects initial infeasibility of MPC and implements the bounded controller in the closed loop. As the closed-loop states evolve under the bounded controller and get closer to the desired steady-state, the supervisor finds (at $t = 5.8$) that the MPC becomes feasible and, therefore, implements it for all future times. Note that despite the “jump” in the control action profile as we switch from the bounded controller to MPC at $t = 5.8$, (see the difference between dotted and dash-dotted profiles in Figure 10.4(f)), the moments of the PSD in the crystallizer continue to evolve smoothly (dash-dotted lines in Figure 10.4(a)–(e)). The supervisor finds that MPC continuous to be feasible and is implemented in closed-loop to stabilize the closed-loop system at the desired steady-state. The dotted lines in Figure 10.4(a)–(e) shows the simulation results of the closed-loop system under the bounded controller only. Compared with the simulation results under the bounded controller, the hybrid predictive controller (dash-dotted lines) stabilizes the system much faster, and achieves a better performance, reflected in a lower value of the performance index (0.1282 *vs* 0.1308). The evolution of PSD under the hybrid predictive controller, shown in Figure 10.5(b), illustrates clearly how the hybrid controller stabilizes the PSD at the desired steady state. The manipulated input profiles for the three scenarios are shown in Figure 10.4(f).

10.3 Predictive control of batch particulate processes

Batch crystallization differs from continuous crystallization in that the withdrawal of product for the batch process is made only once at the end of the batch run. It is commonly used in the chemical, pharmaceutical, photographic, and many other industries as manufacturing process to prepare a wide variety of crystalline products. Compared with continuous particulate processes, batch particulate processes have several desirable features [229]. For instance, in batch particulate processes, the equipment is relatively simple and flexible, and requires a relatively lower level of maintenance. Batch particulate processes are particularly applicable to chemical systems difficult to process, such as processes with toxic or highly viscous properties. Also, experiments on batch particulate processes can be used to examine a large number of operational variables in a short time. Systems that are difficult to operate continuously may conveniently be investigated in a batch-wise manner with relatively minimum development time and investment. For capacity requirements less than 500 kg/h, batch particulate processes are usually more economically advantageous. Furthermore, if the product requires a relatively narrow PSD, a batch particulate process is clearly a better choice.

10.3.1 A seeded batch crystallizer: modeling and dynamics

As an example, we consider the seeded batch cooling crystallizer studied in [180, 184], which produces crystals of potassium sulfate. The population balance model of this crystallizer, describes the evolution of the crystal size distribution, $n(r, t)$, under the joint effects of nucleation (B is the crystal nucleation rate) and crystal growth (G is the crystal growth rate). The evolution of the solute concentration, C , and reactor temperature, T , are described by two ODEs. The process model has the following

form:

$$\begin{aligned}
\frac{\partial n(r, t)}{\partial t} + G(t) \frac{\partial n(r, t)}{\partial r} &= 0, \quad n(0, t) = \frac{B(t)}{G(t)} \\
\frac{dC}{dt} &= -3\rho k_v G(t) \mu_2(t) \\
\frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j) - \frac{\Delta H}{C_p} 3\rho k_v G(t) \mu_2(t)
\end{aligned} \tag{10.18}$$

where ρ is the density of crystals, k_v is the volumetric shape factor, U is the overall heat-transfer coefficient, A is the total heat-transfer surface area, M is the mass of solvent in the crystallizer, C_p is the heat capacity of the solution, T_j is the jacket temperature, ΔH is the heat of reaction, and $\mu_2 = \int_0^\infty r^2 n(r, t) dr$ is the second moment of the PSD. The nucleation rate, $B(t)$, and the growth rate, $G(t)$, are given by

$$\begin{aligned}
B(t) &= k_b e^{-E_b/RT} \left(\frac{C - C_s(T)}{C_s(T)} \right)^b \mu_3, \\
G(t) &= k_g e^{-E_g/RT} \left(\frac{C - C_s(T)}{C_s(T)} \right)^g
\end{aligned} \tag{10.19}$$

where E_b is the nucleation activation energy, E_g is the growth activation energy, b and g are exponents relating nucleation rate and growth rate to supersaturation, C_s is the saturation concentration of the solute, and $\mu_3 = \int_0^\infty r^3 n(r, t) dr$ is the third moment of the PSD. The values of the process parameters are given in Table 11.1. Eq.10.20 is used to calculate the saturation and metastable concentrations corresponding to the solution temperature T . These two concentrations represent the constraints on the solution concentration, i.e., $C_s \leq C \leq C_m$ that must hold during the whole batch run. The initial seed distribution of the seeded batch crystallizer is assumed to be a parabolic distribution, from 250 to 300 μm , and the maximum density of initial seed distribution, which is $2/\mu m \cdot g$ solvent, occurs at 275 μm , i.e., $n(r, 0) = 0.0032(300 - r)(r - 250)$, for $250\mu m \leq r \leq 300\mu m$, and $n(r, 0) = 0$, for $r < 250\mu m$ and $r > 300\mu m$.

$$\begin{aligned}
C_s(T) &= 6.29 \times 10^{-2} + 2.46 \times 10^{-3}T - 7.14 \times 10^{-6}T^2 \\
C_m(T) &= 7.76 \times 10^{-2} + 2.46 \times 10^{-3}T - 8.10 \times 10^{-6}T^2
\end{aligned} \tag{10.20}$$

To study the dynamic behavior of the crystallizer, a second-order accurate finite difference scheme with 1500 discretization points is used to obtain the solution of the system

Table 10.1: Parameter values for the seeded batch cooling crystallizer of Eqs.11.4–10.19.

b	$=$	1.45		g	$=$	1.5	
k_b	$=$	285.0	$1/(s \mu m^3)$	k_g	$=$	1.44×10^8	$\mu m/s$
E_b/R	$=$	7517.0	K	E_g/R	$=$	4859.0	K
U	$=$	1800	$kJ/m^2 \cdot hr \cdot K$	A	$=$	0.25	m^2
ΔH	$=$	44.5	kJ/kg	C_p	$=$	3.8	$kJ/K \cdot kg$
M	$=$	27.0	kg	ρ	$=$	2.66×10^{-12}	$g/\mu m^3$
k_v	$=$	1.5		t_f	$=$	30	min

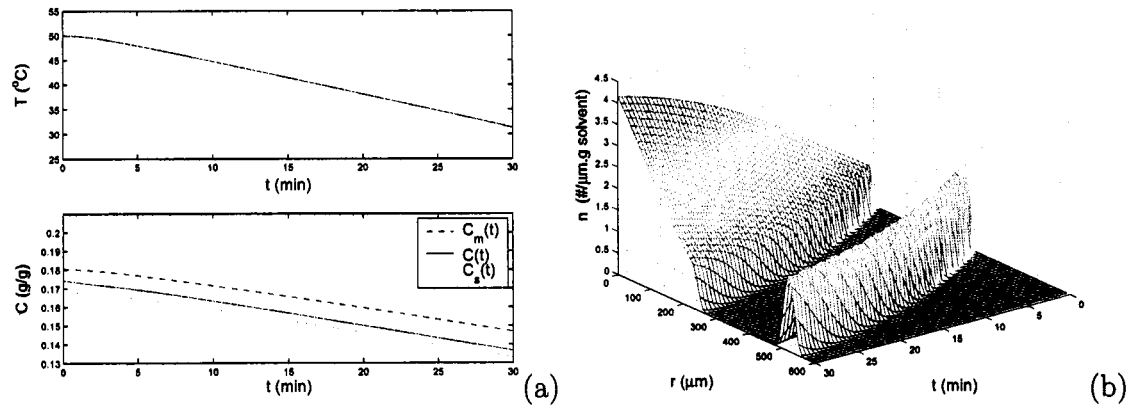


Figure 10.6: Simulation results for the linear cooling strategy: (a) reactor temperature and concentration profiles (dashed and dotted lines represent the upper and lower constraints on the concentration respectively); (b) the evolution of crystal size distribution.

of Eq.11.4. Figure 10.6 shows the evolution of the reactor temperature, T , the solution concentration, C , and the PSD under a linear cooling strategy (where the jacket temperature, T_j , is cooled down linearly from 50°C to 30°C). From Figure 10.6(b), it is clear that there is a gap between the crystals formed by nucleation and those growing from the seeds during the whole reaction period. Based on this observation, we developed two moments models to simulate the dynamics of the crystals formed by nucleation and the crystals growing from the seeds separately. The two models are given by Eqs. 10.21 and 10.22, respectively. The mass and energy balances in these

two models are described by Eq. 10.23.

$$\begin{aligned}\frac{d\mu_0^n}{dt} &= B(t) \\ \frac{d\mu_i^n}{dt} &= iG(t)\mu_{i-1}^n(t), \quad i = 1, 2, 3\end{aligned}\tag{10.21}$$

$$\begin{aligned}\mu_0^s &= k_4 \\ \frac{d\mu_i^s}{dt} &= iG(t)\mu_{i-1}^s(t), \quad i = 1, 2, 3\end{aligned}\tag{10.22}$$

$$\begin{aligned}\frac{dC}{dt} &= -3\rho k_v G(t)(\mu_2^n(t) + \mu_2^s(t)) \\ \frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j) - \frac{\Delta H}{C_p}3\rho k_v G(t)(\mu_2^n(t) + \mu_2^s(t))\end{aligned}\tag{10.23}$$

where μ_i^n and μ_i^s ($i=0,1,2,3$) are the first four moments of the PSD of the crystals formed by nucleation and the crystals growing from the seeds, respectively, and are defined by

$$\begin{aligned}\mu_i^n &= \int_0^{r_g} r^i n(r, t) dr \\ \mu_i^s &= \int_{r_g}^{\infty} r^i n(r, t) dr, \quad i = 0, 1, 2, 3.\end{aligned}\tag{10.24}$$

The superscript, n , stands for nucleation, and s stands for seed. r_g is the radius at the middle of the gap between two groups of crystals. This characteristic radius is a function of time, and can be determined only from the population balance model. Since the model does not consider crystal breakage or agglomeration during the crystallization, μ_0^s , which is the total number of the crystals growing from the seeds, remains constant k_4 during the crystallization period. Figure 10.7 shows a comparison between μ_i^n and μ_i^s , ($i=0,3$), based on simulations of the two moments models. The purpose of deriving two moments models instead of one is to facilitate the design of the predictive controller for the PSD of each group of crystals discussed in the next section.

10.3.2 Predictive controller design: accounting for state and input constraints

Unlike the control of continuous particulate processes, asymptotic stabilization is not an issue in the control of batch processes. More important is the objective of achieving

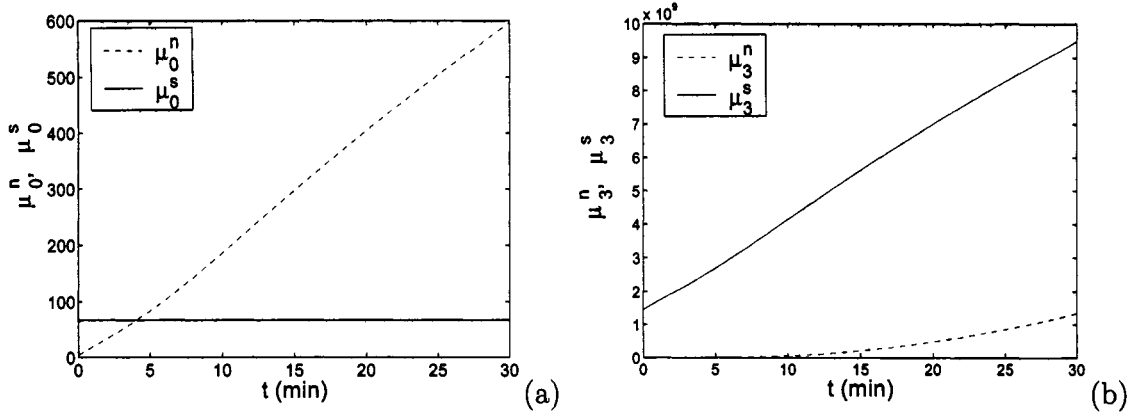


Figure 10.7: Simulation results for the linear cooling strategy: (a) the zeroth and (b) third moments of the PSD of the crystals formed by nucleation (dashed lines) and the those growing from the seeds (solid lines).

a desired particle size distribution at the end of the batch and satisfying state and control constraints during the whole batch run. Significant previous work has focused on PSD control in the batch crystallizers [180, 234]. [155] derived an open-loop optimal control strategy where the objective function involves maximization of the crystal size and the cooling curve is the decision variable. [151] developed a method for assessing parameter uncertainty and studied its effects on the open-loop optimal control strategy, which maximized the weight mean size of the product. [242] developed an on-line optimal control methodology for a seeded batch cooling crystallizer to improve the product quality expressed in terms of the mean size and the width of the distribution. [100] designed a hierarchical multiobjective strategy to control the PSD in semi-batch emulsion polymerization. In these previous works, most efforts were focused on the open-loop optimal control of the batch crystallizer, i.e., the optimal operating condition was calculated off-line and based on mathematical models. The successful application of such a control strategy relies, to a large extent, on the accuracy of the models.

In this section, we focus on developing a closed-loop predictive control system to minimize the total volume of fines (i.e., small crystals formed by nucleation) in the

final product. In the operation of industrial crystallizers, the fines usually cause difficulties in downstream processing equipment (e.g., filtration) and affect both product quality and process economics. Such effects are especially important in a seeded batch crystallizer, since its final products mainly grow from the seeds rather than from the crystal formed by nucleation. Excessive fines may also require a relatively long batch run time to achieve the desired final size of the product. Some experimental studies on fines destruction for batch crystallizers have been reported. [104] first described the application of fines destruction in batch crystallization of potassium sulfate solutions. [190] implemented a feedback control fines dissolution strategy in order to maintain the fines slurry density at some constant value over the batch run. These studies demonstrate the experimental feasibility of dramatically reducing the amount of fines in the final product based on the reactor design.

In this work, a closed-loop predictive control scheme is developed to control the seeded batch cooling crystallizer described by Eq.11.4. The control objective is to minimize the volume of fines in the final product, (i.e., the third moment of the crystals formed by nucleation, μ_3^n), by manipulating the jacket temperature, T_j . The principle moments are calculated from the on-line measured PSD, n , which can be obtained by measurement techniques such as the laser light scattering method. The concentration and reactor temperature are also assumed to be measured in real time. In the closed-loop control structure, the PBM, together with the mass and energy balances of Eq.11.4, are used to describe the process while the reduced-order model of Eqs.10.21–10.22 is used within the MPC for the purpose of prediction. Therefore, the PBM is utilized to simulate the value of the state variables (n, C, T) at $t_i = i\Delta t$, where $i=1,2,\dots,m$, $m = t_f/\Delta t$, and t_f is the length of the crystallization period. Thus, the values of the moments ($\mu_j^n(t_i), \mu_j^s(t_i), j=0,1,2,3$) are calculated from the PSD at every t_i , $n(r, t_i)$. The values of $\mu_j^n(t_i), \mu_j^s(t_i), C(t_i), T(t_i)$, and the optimal trajectory of T_j solved at the previous time step, are used as the initial values for the reduced-order model in MPC to solve an optimization problem for a horizon length of $t_f - t_i$. The

first step of the solution (T_j) is implemented to generate the value of state variables at the end of next time step, t_{i+1} . This procedure is repeated every Δt until the end of the batch run.

Manipulated input limitations and concentration specifications are incorporated as input and state constraints on the optimization problem, which takes the form

$$\begin{aligned}
 \min \quad & \mu_3^n(t_f) \\
 \text{s.t.} \quad & T_{j,min} \leq T_j \leq T_{j,max}, \\
 & C_s \leq C \leq C_m, \\
 & \left| \frac{dT_j}{dt} \right| \leq k_5, \\
 & \mu_3^s(t_f) \geq V_1
 \end{aligned} \tag{10.25}$$

where μ_3^n , μ_3^s , T_j and C are obtained by solving the moments models of Eqs.10.21-10.22. $T_{j,min}$ and $T_{j,max}$ are the constraints on the manipulated variable, T_j , and set to 30°C and 50°C in the simulation. C_s and C_m , which can be calculated from Eq.10.20, are the constraints on the solution concentration. The constant, k_5 , (chosen to be 2°C/min) is the maximum gradient of the jacket temperature. V_1 , chosen as 8.3301×10^9 in the simulations, denotes the lower bound on the total volume of the crystals growing from the seeds. Such constraint on $\mu_3^s(t_f)$ represents a desirable quality of the final product. In the simulation, Δt and t_f are chosen as 30 seconds and 30 min, respectively. The optimization problem is solved using the steepest descent method, which is a gradient-based optimization method and converges faster when compared with other optimization methods such as SQP. Considering the high dimensionality of the nonlinear optimization problem, fast convergence is critical to real time implementation of the feasibility of the predictive control policy. Furthermore, the optimization results based on the steepest descent method and SQP were compared, and only a small difference was observed, while the SQP method took a considerably longer time to converge than the steepest descent method.

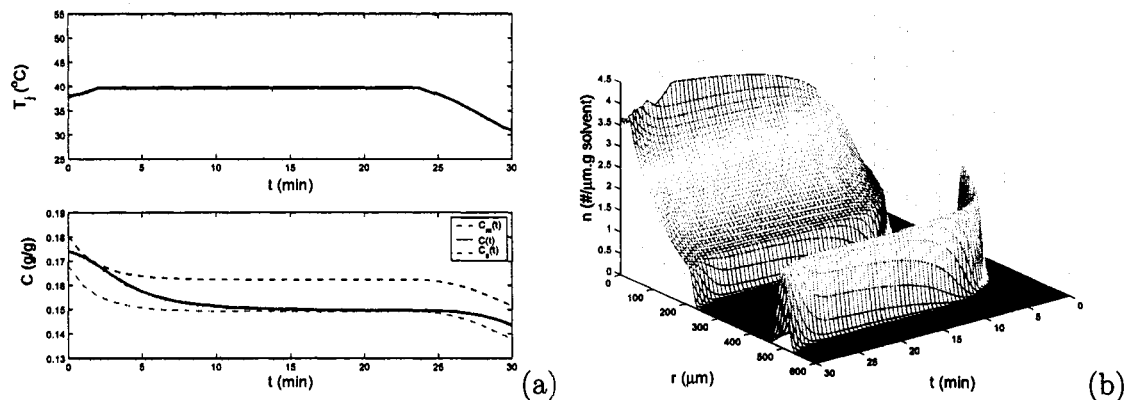


Figure 10.8: Closed-loop simulation results for matched models: (a) Jacket temperature and concentration profiles (dashed and dotted lines represent the upper and lower constraints on the concentration, respectively); (b) the evolution of crystal size distribution.

10.3.3 Closed-loop simulation results

To illustrate the effect of model accuracy on the ability of the predictive controller to control the crystallizer, we considered two different cases. In the first case, it is assumed that the same model parameters are used in the MPC model (i.e., the moments model) and the process model (i.e., population balance model). In another case, a process-model mismatch is allowed by changing the value of the parameter b , which is the exponent relating nucleation rate to supersaturation, from its nominal value of 1.45 to 1.35 in the moments model used in MPC. The simulation results of these two cases were also compared with the open-loop simulation results obtained under the linear cooling strategy.

Figure 10.8 shows the closed-loop simulation results for the case of a perfect process-model match. Figure 10.8(a) shows the optimal trajectory of the jacket temperature and solution concentration while Figure 10.8(b) shows the evolution of the PSD. Clearly, the constraints on the jacket temperature and concentration are respected during the evolution of the closed-loop profiles. When a process-model mismatch is considered, MPC is unable to satisfy the state and control constraints unless the time

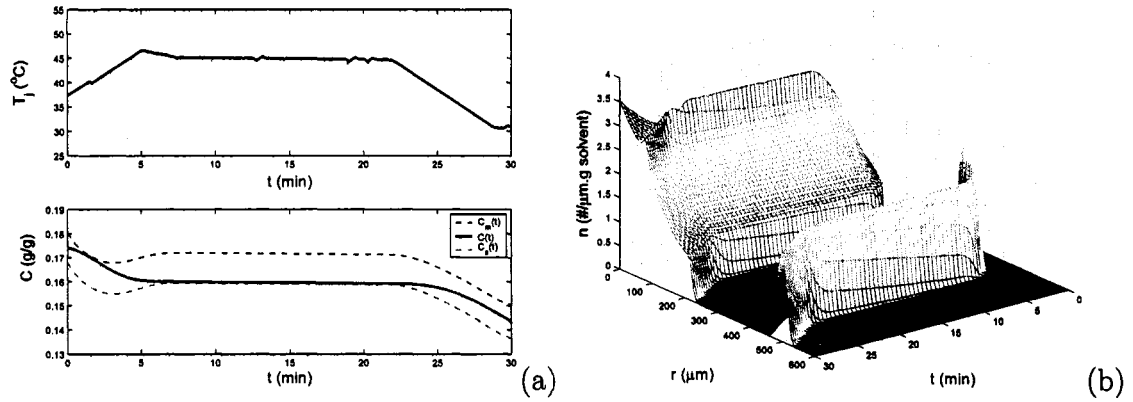


Figure 10.9: Closed-loop simulation results for mismatched models: (a) Jacket temperature and concentration profiles (dashed and dotted lines represent the upper and lower constraints on the concentration, respectively); (b) the evolution of crystal size distribution.

step of the closed-loop control system (the hold time in MPC implementation) is lowered from 30 to 15 seconds, indicating that tighter feedback control is necessary in the case of imperfect model to avoid violation of the constraints. The closed-loop simulation results for the case of a process-model mismatch are shown in Figure 10.9. Comparing Figure 10.8 and Figure 10.9, it is clear that the predictive controller produces a different manipulated input trajectory to enforce constraint satisfaction when the model used in MPC design does not perfectly match the process. It is also noticed that the off-line computed optimal trajectory of T_j based on the mismatched model fails to satisfy the constraints on the solute concentration during the batch run. Such results indicate that feedback control is necessary to guarantee the process operating within the constraints.

Table 10.2: Comparison between the simulation results under four different control strategies.

Control Strategy	μ_3^n ($\mu m^2/g$ solvent)	μ_3^s ($\mu m^2/g$ solvent)
Open-loop with linear cooling	8.9174×10^8	8.3304×10^9
MPC with objective to maximize μ_3^s/μ_3^n	1.7828×10^9	1.0545×10^{10}
MPC with matched moments model	7.7209×10^8	8.3301×10^9
MPC with mismatched moments model	7.8655×10^8	8.3301×10^9

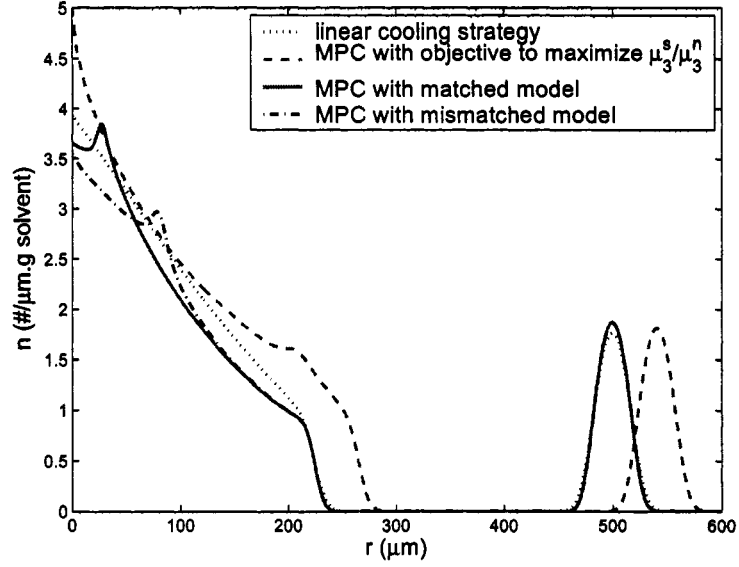


Figure 10.10: Comparison of the final crystal size distributions under 4 different control strategies.

Table 11.3 lists the value of μ_3^n and μ_3^s obtained under four different control strategies: (1) Open-loop with linear cooling, (2) MPC with an objective to maximize μ_3^s/μ_3^n , (3) MPC with the objective of Eq.11.9 under a perfect process-model match, and (4) MPC with the objective of Eq.11.9 under a process-model mismatch. Figure 10.10 compares the final crystal size distribution under these 4 different control strategies. Comparing the results of control strategies (1), (3) and (4), it is clear that MPC can lower the volume of fines (nucleation formed crystals) by 13.4% compared with the linear cooling strategy, while the crystals growing from the seeds in final product still satisfy the product quality requirement. It is also observed that, in the case of process-model mismatch, the predictive control strategy (with a smaller time step) is able to successfully control the system and lower the volume of fines by 11.8% compared with the linear cooling strategy. To demonstrate the effect of the choice of optimization objective in the control strategy on the volume of fines, we carried out simulations with the same control structure, but with a different optimization objective which is to maximize the ratio μ_3^s/μ_3^n . Comparing the results in Table 11.3, it is clear that, when

MPC with this objective is applied, even though the volume of crystals growing from the seeds increases by 26.6% (relative to the linear cooling strategy) which is favorable to product quality, the volume of fines actually increases dramatically by 129.2%. Such large difference on the crystal size distribution of final products can be more clearly seen in Figure 10.10. It shows that the control strategy (2) leads to larger crystals both formed by nucleation and grown from seeds compared to the other 3 control strategies, which allow to produce less crystals formed by nucleation while satisfying the minimum requirement of the volume of the crystals grown from seeds. This result suggests that maximizing the ratio μ_3^s/μ_3^n , while favoring product quality, does so at the expense of a significant increase in the volume of fines which is undesirable. In contrast, the optimization problem of Eq.11.9 provides a more meaningful formulation whereby the volume of fines is explicitly minimized, and the product quality is accounted for as a constraint on the optimization problem. This significant difference in the impact on μ_3^s and μ_3^n between the two optimization formulations suggests that the optimization objective should be selected carefully, when attempting to achieve a desired PSD, in order to strike a balance between enhancing product quality and minimizing the difficulties caused by a large volume of fines. Finally, to study the importance of incorporating measurements in the control system, we carried out a simulation of the predictive control system where the values of the state variables at each time step are not updated using measurements from the process, but instead, are updated using the values generated by simulating the moments model with $b = 1.35$. The simulation results (not shown in the manuscript due to space limitation) show that the state constraints are violated at the end of the open-loop simulation run because of the discrepancy between the PBM and the mismatched moments model, and the lack of measurement feedback.

10.4 Conclusions

In this chapter, we focused on the development and application of predictive algorithms for control of PSDs in continuous and batch particulate processes described by PBMs. The control algorithms were designed on the basis of finite-dimensional models that capture the dominant dynamics of the particulate processes and were tailored to address different control objectives for the continuous and batch processes. Closed-loop simulations have demonstrated the effectiveness of the proposed control algorithms.

Chapter 11

Predictive Control of Crystal Size Distribution in Protein Crystallization

This chapter focuses on the modeling, simulation and control of a batch protein crystallization process that is used to produce the crystals of tetragonal hen egg-white (HEW) lysozyme. First, a model is presented that describes the formation of infinitesimal-size protein crystals via nucleation and the subsequent growth of the crystals via condensation, and predicts the temporal evolution of the crystal size distribution in the size range of $0-300\text{ }\mu\text{m}$. To this end, existing experimental data are used to develop empirical models of the nucleation and growth mechanisms of the tetragonal HEW lysozyme crystal. The developed growth and nucleation rate expressions are used within a population balance model to simulate a batch crystallization process that produces the tetragonal HEW lysozyme crystals. Then, model reduction techniques are used to derive a reduced-order moments model for the purpose of controller design. Online measurements of the solute concentration and reactor temperature are assumed to be available, and a Luenberger-type observer is developed to estimate the moments of the crystal size distribution based on the available measurements. A predictive con-

troller, that uses the available state estimates, is designed to achieve the objective of maximizing the volume-averaged crystal size while respecting constraints on the manipulated input variables (which reflect physical limitations of control actuators) and on the process state variables (which reflect performance considerations). Simulation results demonstrate that the proposed predictive controller is able to increase the volume-averaged crystal size by 30% and 8.5% compared to Constant Temperature Control (CTC) and Constant Supersaturation Control (CSC) strategies, respectively, while reducing the number of fine crystals produced. Furthermore, a comparison of the crystal size distributions (CSDs) indicates that the product achieved by the proposed predictive control strategy has larger total volume and lower polydispersity compared to the CTC and CSC strategies. Finally, the robustness of the proposed method with respect to plant-model mismatch is evaluated. The proposed method is demonstrated to successfully achieve the task of maximizing the volume-averaged crystal size in the presence of plant-model mismatch, and is found to be robust in comparison to open-loop optimal control strategies.

11.1 Introduction

Proteins play a vital role in most biological processes. In addition to constructing large-scale biological structures, such as muscle fibers, smaller protein molecules can function as antibodies, which help the immune system to destroy invading substances like viruses and bacteria, and enzymes, which can catalyze the synthesis of complex compounds, the transformation of complex substances into simpler ones, or the generation of energy in organisms. A protein molecule is a chain of amino acids that are linked by peptide bonds formed by dehydration synthesis. Many pharmaceuticals act by binding to and blocking an active site (the active site is a region on the protein composed of some of the protein's amino acids which have a specific three dimensional arrangement to which a molecule can bind) of a protein [193]. The three-dimensional arrangement of

amino acids, especially, at the active site, determines the specific biological function of the protein molecule.

X-ray and neutron diffraction techniques are the only available methods that can be used to obtain structural information of proteins with molecular weight over 20,000. To be able to study the structure of proteins using these techniques, large protein crystals of high structural perfection, typically with diameters of several hundred microns, are needed. Extensive research studies, using model proteins, such as ferritin, insulin, haemoglobin and lysozyme, that are able to crystallize easily under normal operating conditions, have focussed on growing large protein crystals of high structural perfection under various operating conditions, including low gravity [28], different pressure conditions [156], flow condition of the solution [173], purity [226], temperature [46] and concentrations of precipitants and buffers [172, 123]. In addition, numerous research studies have considered the problem of modeling of protein nucleation [82, 171] and growth [61, 119, 77]. The reader may also refer to [143, 196, 231, 225] for excellent reviews on this subject. Among these model proteins, the tetragonal form of HEW lysozyme is most popular and widely used [30]. HEW lysozyme is composed of 129 amino acids, with a molecular weight of 14,388. It is a naturally occurring enzyme, and has antibacterial activity against gram-positive bacteria.

The experimental studies not only benefit the determination of the protein structure, but also provide, through an understanding of the nucleation and growth mechanisms of protein crystallization, a way of determining the operating conditions necessary to achieve protein crystals of desired properties, which can often be expressed in the form of a desired size distribution. In the pharmaceutical industry, the size distribution of the protein crystal is a very critical variable and in many applications a predetermined, typically narrow, crystal size distribution is necessary in order to guarantee a desired drug delivery performance.

Protein crystals intrinsically grow much slower than most inorganic crystals at the

same supersaturation [194]. This is probably one of the reasons why it has been widely believed that implementation of advanced control algorithms is not important for growing perfect crystals in protein crystallization. However, experimental results show that even in protein crystallization processes, certain growth conditions lead to crystal defect formation (for example, the structural defect density increases when the growth rate is relatively high [152]), which necessitates appropriately choosing the operating conditions. Furthermore, for a given choice of operating conditions, it is critical to implement real-time control to mitigate the affect of disturbances that might drive the operating conditions off the desired values.

While numerous experimental studies of the nucleation and growth mechanisms of protein crystallization have been carried out, very few results on control of protein crystallization are available. The feasibility and effectiveness of temperature control in protein crystallization was discussed in [195], where a series of preliminary experiments were conducted to demonstrate the ability of temperature control to manipulate the supersaturation in the protein crystallization. In [51], a method for control of nucleation in small protein solutions was developed and focused on control of a relatively smaller number of protein crystals. In [228], an experiment system was developed to control a crystallization cell, and focused on the development of the control system, including the hardware and software for measurement and communication, rather than on the control algorithm. A Constant Supersaturation Concentration (CSC) control strategy was developed in [200] to achieve large protein crystals. Although this strategy shows significant advantage in increasing the crystal size compared to the case where, for instance, the temperature in the crystallizer is held steady, it is not optimal with respect to maximizing the crystal size. Furthermore, being an open-loop control strategy, it is not robust with respect to modeling errors and disturbances affecting the operation of the crystallizer.

Motivated by the above considerations, this chapter focuses on the modeling, sim-

ulation and control of a batch protein crystallization process that is used to produce the crystals of tetragonal hen egg-white (HEW) lysozyme. First, a model is presented that describes the formation of infinitesimal-size protein crystals via nucleation and the subsequent growth of the crystals via condensation, and predicts the temporal evolution of the crystal size distribution in the size range of $0 - 300 \mu m$. To this end, existing experimental data are used to develop empirical models of the nucleation and growth mechanisms of the tetragonal HEW lysozyme crystal. The developed growth and nucleation rate expressions are used within a population balance model to simulate a batch crystallization process that produces the tetragonal HEW lysozyme crystals. Then, model reduction techniques are used to derive a reduced-order moments model for the purpose of controller design. Online measurements of the solute concentration and reactor temperature are assumed to be available, and a Luenberger-type observer is developed to estimate the moments of the crystal size distribution based on the available measurements. A predictive controller, that uses the available state estimates, is designed to achieve the objective of maximizing the volume-averaged crystal size while respecting constraints on the manipulated input variables (which reflect physical limitations of control actuators) and on the process state variables (which reflect performance considerations). Simulation results demonstrate that the proposed predictive controller is able to increase the volume-averaged crystal size by 30% and 8.5% compared to Constant Temperature Control (CTC) and Constant Supersaturation Control (CSC) strategies, respectively, while reducing the number of fine crystals produced. Furthermore, a comparison of the crystal size distributions (CSDs) indicates that the product achieved by the proposed predictive control strategy has larger total volume and lower polydispersity compared to the CTC and CSC strategies. Finally, the robustness of the proposed method with respect to plant-model mismatch is evaluated. The proposed method is demonstrated to successfully achieve the task of maximizing the volume-averaged crystal size in the presence of plant-model mismatch, and is found to be robust in comparison to open-loop optimal control strategies.

11.2 Modeling of nucleation and growth rate of tetragonal lysozyme crystal

The formation of a single crystal can be modeled as comprising of two sequential processes: crystal nucleation and growth. Experimental studies [61, 77, 82, 113, 171] have considered various operating conditions and developed empirical models for the nucleation and growth of the tetragonal HEW lysozyme crystal. Owing to large differences in operating conditions, significant discrepancies exist in the nucleation and growth data obtained in these works. In the current study, we focus on one operating condition (fixed buffer, pH value and salt concentration; these values are selected based on the availability of experimental results), and develop an empirical model of the nucleation and growth mechanisms by using the existing experimental results available for these operating conditions. In the remainder of this section, experimental results on the solubility, nucleation and growth of the tetragonal HEW lysozyme crystal are analyzed and an empirical model of the nucleation and growth mechanisms is developed.

11.2.1 Solubility of the tetragonal HEW lysozyme crystals

The main driving force for the crystal nucleation and growth is understood to be supersaturation, i.e., the concentration of the solution in excess of the saturation concentration (solubility). The solubility of lysozyme at different pH and precipitant concentrations has been studied by [27, 194, 199]. In [27], the solubility experiments covered a pH range from 4.0 to 5.4, NaCl concentration (weight/volume) range from 2.0% to 7.0% and temperature range from 4 to 25°C. The solubility of HEW lysozyme was also studied in [194], for pH at 4.5, NaCl concentration at 2.1%, 2.5% and 3.0%, and a temperature range from 11 to 30°C. By implementing a more accurate measurement technique (two-beam interferometry), the solubility in a solution with pH of 4.5 and NaCl concentration of 2.5% was measured in [199].

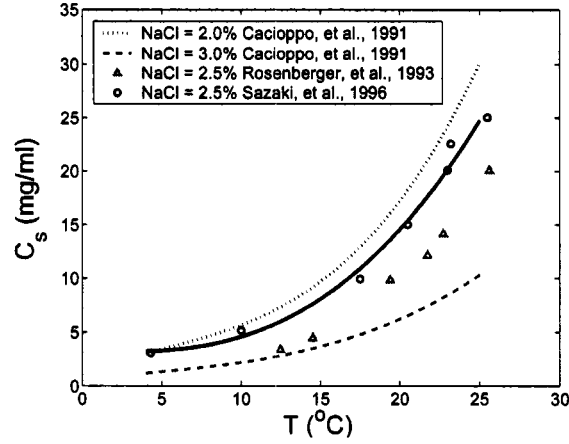


Figure 11.1: Solubility data of tetragonal lysozyme crystal: dotted and dashed lines show the data in [27], $\triangle$ shows data in [194], $\circ$ shows data in [199], and the solid line shows the empirical model of Eq.11.1).

Since purer samples and more advanced measurement techniques were used in [199], this solubility data, shown as circles in Figure 11.1, is considered to be more accurate and used to derive parameters of the empirical model in our work. A third-order polynomial is used to fit the data, and the fitted curve is shown as the solid curve in Figure 11.1. Also shown in Figure 11.1 is the solubility data in [27, 194, 199] which exhibit trends similar to the data set obtained in [199], with solubility being low at low temperature and increasing significantly with increasing temperature. The resulting expression for the solubility, $C_s(T)$, is as follows:

$$C_s(T) = 1.0036 \times 10^{-3}T^3 + 1.4059 \times 10^{-2}T^2 - 0.12835T + 3.4613 \quad (11.1)$$

11.2.2 Modeling of the nucleation rate of the tetragonal HEW lysozyme crystals

For biological molecules, crystals are most often grown with no seeding, where primary nucleation must precede crystal growth. Thus, an understanding of the nucleation

process and the effect of various parameters on nucleation is essential for successful production of protein crystals. Recently, significant progress has been made in understanding the nucleation and crystallization of globular proteins, including the formation of compositional and structural crystal defects [196].

Insight into the interactions of protein macro-ions in solution, obtained from light scattering, small angle X-ray scattering and osmotic pressure studies shows that the nucleation of globular proteins is governed by the same principles as those of small molecules. In particular, supersaturation, the difference between solute concentration and solubility, is understood to be the main driving force for both nucleation and growth of the protein crystals. Changes of operating conditions, such as lowering temperature, increasing salt concentration, pH or protein concentration influence nucleation [105] via influencing the solubility. Therefore, to develop an empirical model, it is assumed that the homogenous nucleation rate of the tetragonal HEW lysozyme crystal is only a function of solute concentration and supersaturation. The following two-parameter empirical expression is used as an empirical model to describe the crystal nucleation rate $B(t)$ [82, 23]:

$$B(t) = k_a C \exp\left(-\frac{k_b}{\sigma^2}\right) \quad (11.2)$$

where σ , the supersaturation, is defined as $\sigma = \ln(C/C_s)$, where C is the solute concentration and C_s is the solubility and k_a and k_b are parameters that are obtained using experimental results.

Different techniques, including light scattering [23], temperature jump [82] and microcalorimetry [46] have been implemented to measure the nucleation rate of the tetragonal HEW lysozyme crystal. We initially use the measurements of concentration and temperature to compute the solubility according to Eq.11.1 and the supersaturation. The experimentally obtained values of the nucleation rate, and the computed concentration and supersaturation values, are then used to obtain the parameters k_a and k_b in Eq.11.2. Both sets of data yield the same values for $k_a = 1044.4/min.cm^3$

and $k_b = 51.33$ and a single straight line serves as the best fit for both sets of data (see Figure 11.2).

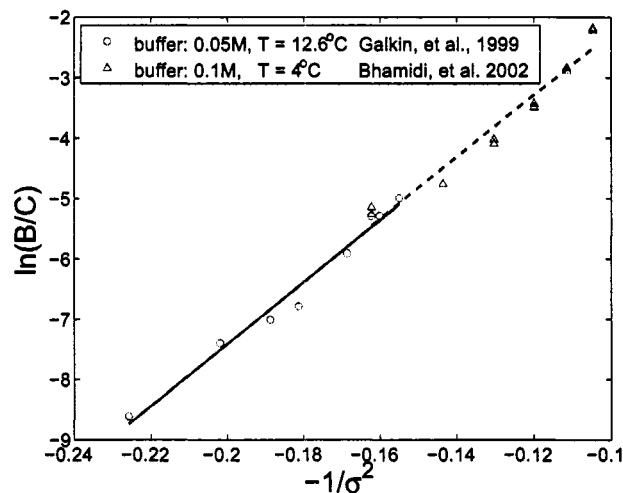


Figure 11.2: Comparison of the experimental results on the nucleation rate of tetragonal HEW lysozyme crystal: ○ show the data in [82], △ show the data in [23], and the empirical model of Eq.11.2 is shown by the solid line.

11.2.3 Modeling of the growth rate of tetragonal HEW lysozyme crystals

Figure 11.3 shows the structure of a tetragonal HEW lysozyme crystal, which shows well-defined facets similar to inorganic crystals. This observation indicates that growth occurs via the spreading of layers from growth step sources such as dislocations and 2D nuclei [60]. The microstructure of the faces of protein crystals has been revealed by implementing advanced measurement techniques, such as electron microscopy and atomic force microscopy (AFM). Experimental results have shown that the growth step generation at screw dislocations outcrops and 2D nucleation islands is the dominant growth mechanism in the growth of tetragonal lysozyme crystals [60]. Experimental studies have also shown that screw dislocation and 2D nucleation are the rate-limiting steps for the growth of tetragonal HEW lysozyme crystals [61, 60, 225]. Detailed spiral growth and 2D nucleation theories have been developed [21]. Experiments in [226] also

showed the existence of a critical supersaturation, above which the crystal growth is dominated by 2D nucleation. When the supersaturation is higher than about 1.6, the increase in the growth rate with increasing supersaturation is much faster than at low supersaturation. This is consistent with the results in [61] which concludes that the growth rates at high and low supersaturation are dominated by two different mechanisms. The experimental results suggest that dislocations may be involved in the lysozyme crystal growth at low supersaturation, and 2D nucleation could be dominant at high supersaturation which leads to equal growth rates on all crystal facets, thereby leading to the formation of symmetrical crystals. Furthermore, it was found that orthorhombic lysozyme crystals, rather than tetragonal crystals, tend to be formed at low supersaturation [77]. This suggests that the supersaturation should be maintained within appropriate bounds to obtain crystals of desired morphology. Empirical and

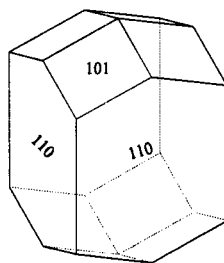


Figure 11.3: Structure of a tetragonal HEW lysozyme crystal.

theoretical models of crystal growth mechanisms have been developed and compared to experimental results [61, 123, 119]. These works studied the growth rate of the tetragonal HEW lysozyme crystal under various operating conditions, such as different precipitation concentration, solute concentration and temperature. All of them concluded the power-law dependence of the growth rate on supersaturation σ . Some experimental results of the studies of the growth rate of the tetragonal HEW lysozyme crystal are shown in Figure 11.4 [172, 61, 119]. The growth rate data corresponding to the supersaturation are plotted on log-log scale. It is clear that the growth rate changes exponentially with the supersaturation, and that the value of the exponents

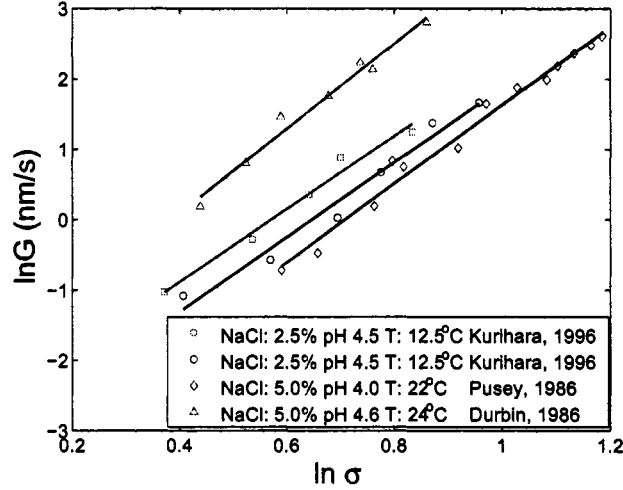


Figure 11.4: Comparison of the experimental results on the growth rate of tetragonal HEW lysozyme crystal: $\diamond$ shows the data in [172], $\triangle$ shows data in [61], and $\square$ and $\circ$ show the data in [119].

obtained from different experiments are very close. In our work, we use the following empirical model to describe the growth rate of the tetragonal HEW lysozyme crystals as a function of supersaturation:

$$G(t) = k_g \sigma^g \quad (11.3)$$

where k_g is the pre-exponential factor of the growth rate. Since the composition of the solution considered in the present study is the same as that in [119]; the experimental data of [119], shown as the circles in Figure 11.4, are used to compute the values of k_g and g in Eq.11.3; this yields $k_g = 3.1451 \times 10^{-9} \text{ cm/min}$ and $g = 5.169$.

11.3 Modeling of a batch crystallizer for protein crystallization

The mathematical models of particulate processes, including the protein crystallization process in question, are typically obtained through the application of population, material and energy balances and consist of systems of nonlinear partial integro-differential

equations that describe the evolution of the particle size distribution (PSD), coupled with systems of nonlinear ordinary differential equations (ODEs) that describe the evolution of the state variables of the continuous phase. There is an extensive literature on population balance modeling, numerical solution, and dynamical analysis of particulate processes, see, for example, [176]; see also [40] for further details and references.

For the batch crystallizer considered in this chapter, a population balance model can be used to describe the evolution of the crystal size distribution (CSD), $n(r, t)$. The evolution of the solute concentration, C , and crystallizer temperature, T , are described by two ODEs. The process model has the following form:

$$\begin{aligned}\frac{\partial n(r, t)}{\partial t} + G(t) \frac{\partial n(r, t)}{\partial r} &= 0, \quad n(0, t) = \frac{B(t)}{G(t)} \\ \frac{dC}{dt} &= -24\rho k_v G(t) \mu_2(t) \\ \frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j)\end{aligned}\tag{11.4}$$

where $G(t)$ is the growth rate, $B(t)$ is the nucleation rate, ρ is the density of crystals, k_v is the volumetric shape factor, U is the overall heat-transfer coefficient, A is the total heat-transfer surface area, M is the mass of solvent in the crystallizer, C_p is the heat capacity of the solution and $\mu_2 = \int_0^\infty r^2 n(r, t) dr$ is the second moment of the CSD. The nucleation rate, $B(t)$, and the growth rate, $G(t)$, are given by Eqs. 11.2 and 11.3, respectively, where the parameters obtained from experimental results are used, and reported in Table 11.1. Note that because of the tetragonal form of the crystals and the existence of about 46% of solvent in each crystal [122], the volumetric shape factor, k_v , is set equal to 0.54. To simulate the population balance model of Eq.11.4, a second-order accurate finite difference scheme with sufficient (3000) discretization points is used (while not shown here for the sake of brevity, it was verified that simulations with higher number of discretization points yield results that are almost indistinguishable from those obtained using 3000 discretization points). Figure 11.5 shows the evolution of the solute concentration (C), the supersaturation (σ) and the CSD at a constant crystallizer temperature ($T = 15^\circ\text{C}$).

Table 11.1: Parameter values for the batch crystallizer model of Eqs.11.3 and 11.4.

k_a	$1044.4/(\text{min cm}^3)$	k_g	$3.1451 \times 10^{-9} \text{cm/min}$
k_b	51.33	g	5.169
k_v	0.54	ρ	$1.40 \times 10^3 \text{mg/cm}^3$
U	$1800 \text{kJ/m}^2 \cdot \text{hr} \cdot \text{K}$	A	0.25m^2
M	10kg	C_p	$4.13 \text{kJ/K} \cdot \text{kg}$

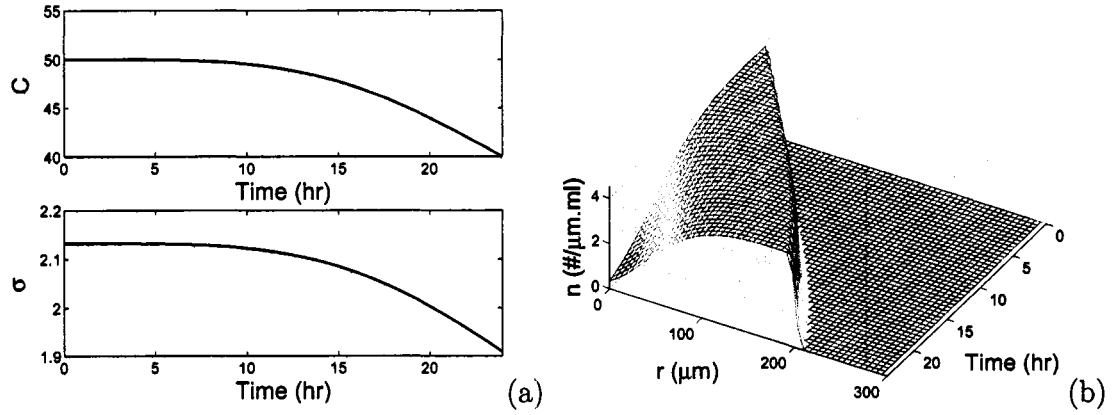


Figure 11.5: Open-loop simulation results of the solute concentration (C), the supersaturation (σ) and the CSD at a constant crystallizer temperature ($T = 15^\circ\text{C}$).

11.4 Predictive control of the batch protein crystallizer

Early work on control of particulate processes focused mainly on the understanding of fundamental control-theoretic properties of PBMs and the application of conventional control schemes to crystallizers and emulsion polymerization processes, see, e.g., [189], and the references therein. More recently, the realization that PBMs – owing to their infinite-dimensional nature – cannot be used directly for the synthesis of practically implementable model-based controllers, has motivated significant research work on the development of a general order reduction procedure, based on combination of the method of weighted residuals and approximate inertial manifolds, which allows deriving low-order ODE approximations that capture the dominant dynamics of particulate processes and can, therefore, serve as an appropriate basis for the design of low-order controllers that can be readily implemented in practice, see, for example, [34]. This approach subsequently laid the foundation for the development of a systematic framework for solving a number of important control problems for particulate processes, including the problem of dealing with the highly nonlinear behavior, e.g., owing to complex growth, nucleation, agglomeration and breakage mechanisms, and the Arrhenius dependence of nucleation laws on solute concentration in crystallizers [34], the problem of control under model uncertainty [35], and the problem of control under actuator constraints [64].

The crystal size distribution or the moments of the crystal size distribution are often not available as online measurements. Solute concentration and temperature, on the other hand, are more readily available as measurements and can be used to generate estimates of the moments which are required for implementation of feedback control strategies based on the reduced order moments model. Also, given the availability of a reduced order moments model and estimates of the moments, desirable product characteristics need to be appropriately expressed in terms of constraints on the crystal size distribution, and physical limitations on the manipulated variable (jacket temper-

ature) must be accounted for in the control design. In the remainder of this section we first use model reduction techniques to develop a reduced order model suitable for the purpose of real-time control. Next, we present the design of a state estimator, to provide estimates of the states of the reduced order model using available measurements. Then we present a predictive controller design that uses the state estimates for the purpose of real-time control, while incorporating performance considerations as appropriate constraints on the moments of the crystal size distribution and accounting for the constraints on the manipulated input variable.

11.4.1 Model reduction

Owing to the fact that the dominant dynamics of the crystallizer are characterized by a small number of degrees of freedom [34], the method of moments [99] (see also [40, 209, 136]) is applied to the system of Eq.11.4 to derive an approximate ODE model. Defining the i th moment of $n(r, t)$ as:

$$\mu_i = \int_0^\infty r^i n(r, t) dr, \quad i = 0, 1, \dots, \infty \quad (11.5)$$

multiplying the population balance in Eq.11.4 by r^i , and integrating over all crystal sizes, the following infinite set of ordinary differential equations, which describes the rate of change of the moments of the crystal size distribution, solute concentration and temperature, is obtained:

$$\begin{aligned} \frac{d\mu_0}{dt} &= B(t) \\ \frac{d\mu_i}{dt} &= iG(t)\mu_{i-1}(t), \quad i = 1, 2, \dots, \infty \\ \frac{dC}{dt} &= -24\rho k_v G(t)\mu_2(t) \\ \frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j) \end{aligned} \quad (11.6)$$

Note that in Eq.11.6, the ODEs describing the dynamics of the first N moments, where N is any positive integer greater than or equal to 3, the solute concentration and the

crystallizer temperature are independent of the moments of order $N + 1$ and higher. This implies that a set of ordinary differential equations, which include the first N moments and the evolution of the solute concentration and crystallizer temperature, would provide an accurate description of the evolution of the first N moments, the solute concentration and the crystallizer temperature. In this case, the nucleation and growth rates are independent of crystal size and this allows closure of the moment equations (note that closure can still be achieved if the nucleation and growth mechanism are dependent on certain moments of the crystal size distribution). In cases where the moments do not close, a finite set of ordinary differential equations describing the evolution of the moments would provide an approximation that gets increasingly better as the number of moments included in the reduced-order model is increased.

As will be seen in section 11.4.3, the control objective will require computation of μ_3 and μ_4 , hence N is chosen as 4 and the following reduced-order model is used for the purpose of controller design:

$$\begin{aligned}
\frac{d\mu_0}{dt} &= B(t) \\
\frac{d\mu_i}{dt} &= iG(t)\mu_{i-1}(t), \quad i = 1, 2, 3, 4 \\
\frac{dC}{dt} &= -24\rho k_v G(t)\mu_2(t) \\
\frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j)
\end{aligned} \tag{11.7}$$

11.4.2 State estimator design

In this section, we present an observer design that uses measurements of the solute concentration, C , and temperature T and the reduced order moments model, to generate estimates of the moments. Note that building an observer for a nonlinear system of the form of Eq.11.7 with guaranteed convergence properties is in general a difficult task. In this chapter, an extended Luenberger-type observer is used to estimate the values of the moments of the CSD and takes the following form:

Table 11.2: Parameter values for the Luenberger-type observer of Eq.11.8.

L_0	-0.4	L_1	0.05
L_2	0.001	L_3	1.7×10^{-5}
L_4	3×10^{-7}	L_5	-0.1

$$\begin{aligned}
 \frac{d\hat{\mu}_0}{dt} &= \hat{B}(t) + L_0(C_m - \hat{C}) \\
 \frac{d\hat{\mu}_i}{dt} &= i\hat{G}(t)\hat{\mu}_{i-1}(t) + L_i(C_m - \hat{C}), \quad i = 1, \dots, 4 \\
 \frac{d\hat{C}}{dt} &= -24\rho k_v \hat{G}(t)\hat{\mu}_2(t) + L_5(C_m - \hat{C})
 \end{aligned} \tag{11.8}$$

where C_m is the online measurement of the solute concentration, $\hat{B}(t)$ and $\hat{G}(t)$ are the nucleation and growth rates computed using the online measurement of T and values of the estimates of $\hat{\mu}_i$ and $\hat{C}$, and L_i , $i = 0, \dots, 5$ are the observer gains (these values were obtained via running open-loop simulations and comparing the evolution of the state with the state estimates for different choices of the observer gains), reported in Table 11.2. Figures 11.6a-f show a comparison between the evolution of the solute concentration (solid line in Figure 11.6a) and the moments μ_i , $i = 0, 1, \dots, 4$ (solid lines in Figures 11.6b-f) and the estimates of the solute concentration $\hat{C}$ and the moments, for the perfect model case (dotted lines), and for two instances of plant-model mismatch, where the parameter g in the moment model is 90% and 110% of its nominal value, respectively (denoted by $\circ$ and $*$), under open-loop simulations. In all instances, the effect of measurement noise, often found in experimental readings, is simulated by introducing a white noise with variance of 0.5 mg/ml in the measurement of the solute concentration. Furthermore, to evaluate the convergence of the estimates to the true state values, the initial value of the estimated solute concentration, $\hat{C}$, is set to 40 mg/ml, while the true value is 50 mg/ml.

Note that for the batch crystallization considered in this chapter, the initial values of the moments at the beginning of the batch run are identically equal to zero, because

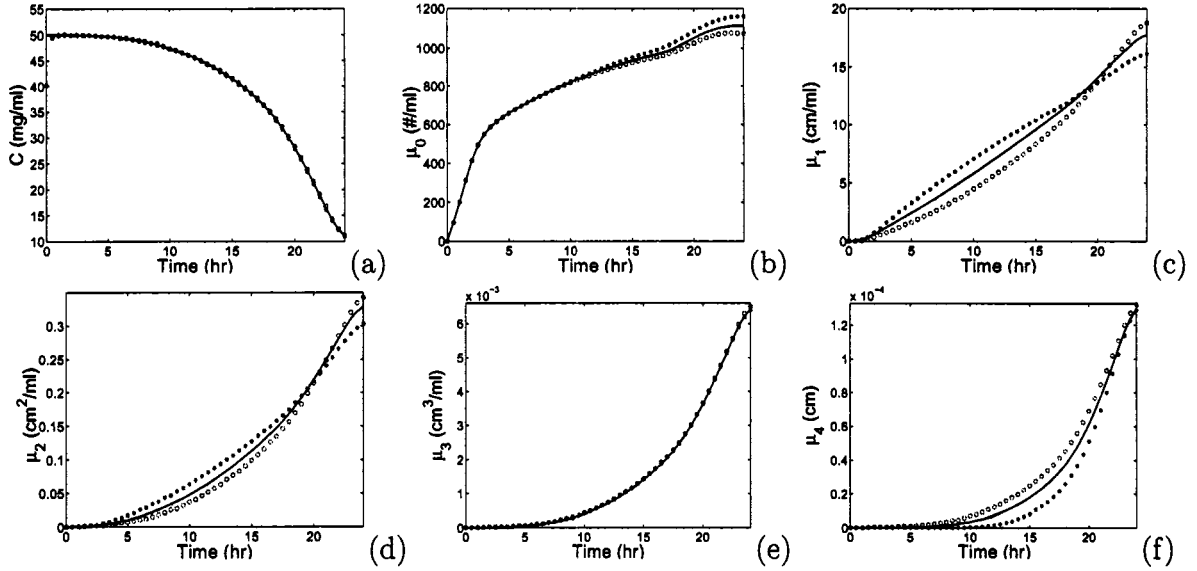


Figure 11.6: Comparison between the true value and the estimates of C (the solute concentration) and $\mu_i, i = 0, 1, \dots, 4$.

there are no crystals initially inside the crystallizer. In the case of a perfect model, the state estimates converge quickly to the true values, and in both cases of plant-model mismatch, the observer generates satisfactory estimates of the states (see Figure 11.6a-f).

11.4.3 Predictive controller formulation

Significant previous work has focused on CSD control in batch crystallizers, e.g., [180, 234]. In [155], an open-loop optimal control strategy was derived, where the objective function involves maximization of the crystal size, and the cooling curve is the decision variable. In [151], a method was developed for assessing parameter uncertainty and studied its effects on the open-loop optimal control strategy, which maximized the weight mean size of the product. An on-line optimal control methodology was developed for a seeded batch cooling crystallizer in [234, 242]. Most of these works focused on open-loop optimal control of the batch crystallizer, i.e., the optimal operating condition is calculated off-line and implemented during the batch operation.

The successful achievement of the control objective in such a control strategy depends heavily on the accuracy of the models, and disturbances affecting the batch operation may easily drive the process off the desired trajectory.

The volume-averaged crystal size is an important parameter that characterizes the crystal size distribution [242], especially for pharmaceuticals. A larger volume-averaged crystal size facilitates downstream processing of the crystals, such as filtration. As mentioned previously, experimental results show low supersaturation leads to the cessation of the crystal growth, while needle like, instead of tetragonal crystals, form if the supersaturation is too high and this necessitates maintaining the supersaturation (and similarly, temperature) within an acceptable range. Constraints on the jacket temperature reflect the physical limitations in supplying the coolant at a prescribed temperature.

Based on these considerations, we present in the remainder of this section a predictive controller formulation that uses the available estimates of the process states for the purpose of feedback control to maximize the volume-averaged crystal size, while respecting the constraints on the reactor temperature, solute concentration and the jacket temperature. Specifically, at time t_i , the control trajectory is computed by

solving an optimization problem of the form:

$$\begin{aligned}
\min \quad & -\frac{\mu_4(t_f)}{\mu_3(t_f)} \\
s.t. \quad & \frac{d\mu_0}{dt} = k_a C \exp\left(-\frac{k_b}{\sigma^2}\right) \\
& \frac{d\mu_i}{dt} = i k_g \sigma^g \mu_{i-1}(t), \quad i = 1, \dots, 4 \\
& \frac{dC}{dt} = -24 \rho k_v k_g \sigma^g \mu_2(t) \\
& \frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) \\
& \mu_i(t_i) = \hat{\mu}_i(t_i) \\
& C(t_i) = \hat{C}(t_i) \\
& t_i \leq t \leq t_f, \\
& T_{min} \leq T \leq T_{max}, \\
& T_{j\ min} \leq T_j \leq T_{j\ max}, \\
& \sigma_{min} \leq \sigma \leq \sigma_{max}, \\
& \left| \frac{dC_s}{dt} \right| \leq k_1
\end{aligned} \tag{11.9}$$

$$n(0, t) \leq n_{fine}, \forall \ t \geq t_f/2 \tag{11.10}$$

where T_{min} and T_{max} are the constraints on the crystallizer temperature, T , and are specified as 4°C and 22°C, respectively. $T_{j\ min}$ and $T_{j\ max}$ are the constraints on the manipulated variable, T_j , and are specified as 3°C and 22°C, respectively. The constraints on the supersaturation σ are $\sigma_{min} = 1.73$ and $\sigma_{max} = 2.89$. The constant, k_1 , (chosen to be 0.065mg/ml·min) specifies the maximum rate of change of the saturation concentration C_s . n_{fine} is the largest allowable number of nuclei at any time instant during the second half of the batch run, and is set to be 5/ $\mu m.ml$. Previous work has shown that the objective of maximizing the volume-averaged crystal size can result in a large number of fines in the final product [130]. The constraint of Eq.11.10, by restricting the number of nuclei formed at any time instant during the second half of the batch run limits the fines in the final product.

In the closed-loop control structure, the PBM, together with the mass and energy balances of Eq.11.4, are used to simulate the batch crystallization process. Measure-

ments of the solute concentration and the crystallizer temperature are assumed to be available, and are used by the Luenberger-type observer of Eq.11.8 to generate estimates of the moments. These estimates are used as initial conditions of the states in the moments model, which is used in the predictive controller to compute the control trajectory. Specifically, the PBM is utilized to simulate the value of the state variables (n, C, T) at $t_i = i\Delta t$, where $i=1,2,\dots,m$, $m = t_f/\Delta t$, and t_f is the length of the crystallization period. The measurements of C and T are used to compute estimates of $\hat{\mu}_j(t_i)$. These estimates are used as initial values of the states in the reduced order moments model, and used within the optimization problem over a horizon length of $t_f - t_i$ to compute the optimal control trajectory. The first step of the solution (T_j) is implemented in the closed-loop, and this procedure is repeated every Δt until the end of the batch run. In the simulations, Δt and t_f are chosen as 5 minutes and 24 hours, respectively. Eq. 11.4, that describes the evolution of the batch crystallizer, is solved by a second-order accurate finite difference scheme with 3000 spatial discretization points and 7200 temporal discretization points and the optimization problem is solved using sequential quadratic programming (SQP).

11.5 Closed-loop simulation results

In this section, we first demonstrate the effect of the unavailability of measurements of the moments on the evolution of the closed-loop system. To this end, we compare the following scenarios: 1) state feedback control (i.e., the measurements of the moments are available); the control trajectory initially computed by the optimization problem is shown by the dotted line, and the closed-loop control trajectory under state feedback is shown by the dash-dotted line in Figure 11.7(a), and 2) output feedback control; the control trajectory initially computed by the optimization problem is shown by the dashed line 11.7(a), and the closed-loop control trajectory under output feedback control is shown by the solid line in Figure 11.7(a) (the corresponding supersaturation

profiles are shown in Figure 11.7(b)).

Note that in the case of state feedback control, the control trajectory computed initially coincides with that under the closed-loop implementation (the dotted and dash-dotted lines overlap), since the tail of the control trajectory continues to be the solution to the optimization problem at the successive times. In the case of output feedback control, however, the control trajectory is initially computed based on the available (incorrect) estimates of the state variables, and leads to state constraints violation if implemented for the entire duration of the batch run (see dashed lines in Figure 11.7(b), where the supersaturation hits the lower constraint at $t = 18.2$ hours).

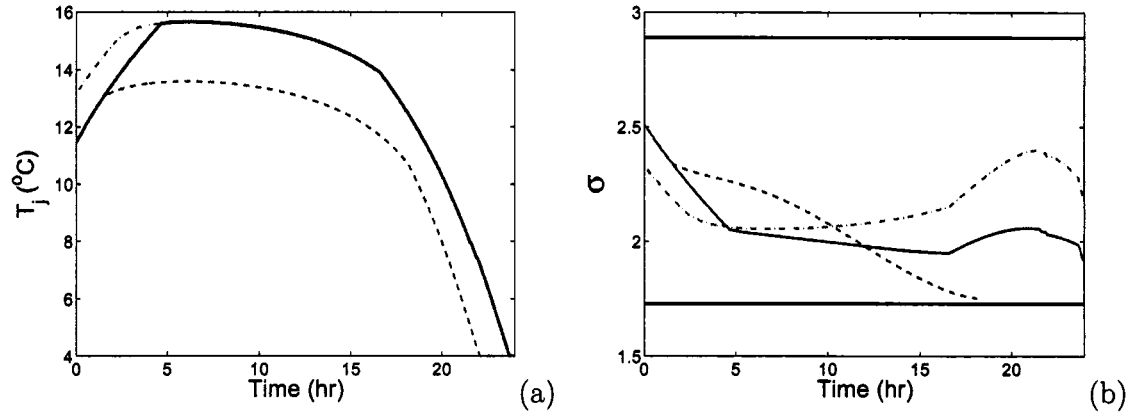


Figure 11.7: (a) Jacket temperature and (b) supersaturation profiles under 1) state feedback control: dotted (implementation of the initially computed manipulated input trajectory) and dash-dotted (closed-loop state feedback control) lines, and 2) output feedback control; dashed (implementation of the initially computed manipulated input trajectory) and solid (closed-loop output feedback control) lines.

Next, we compare the performance of the proposed predictive controller to that of three other control strategies, Constant Temperature Control (CTC), Constant Supersaturation Control (CSC) and open-loop optimal control strategy. To study the ability of the proposed predictive control strategy to maximize the performance objective while avoiding the formation of a large number of fines in the final product, the predictive controller of Eq.11.9 is implemented with the additional constraint (Eq.11.10) on the

fines in the final product. Also, to evaluate the robustness of the proposed method with respect to plant-model mismatch, we consider a 10% error in the nominal value of the parameter g - which is the exponent relating growth rate to supersaturation - in the moments model used in the predictive controller.

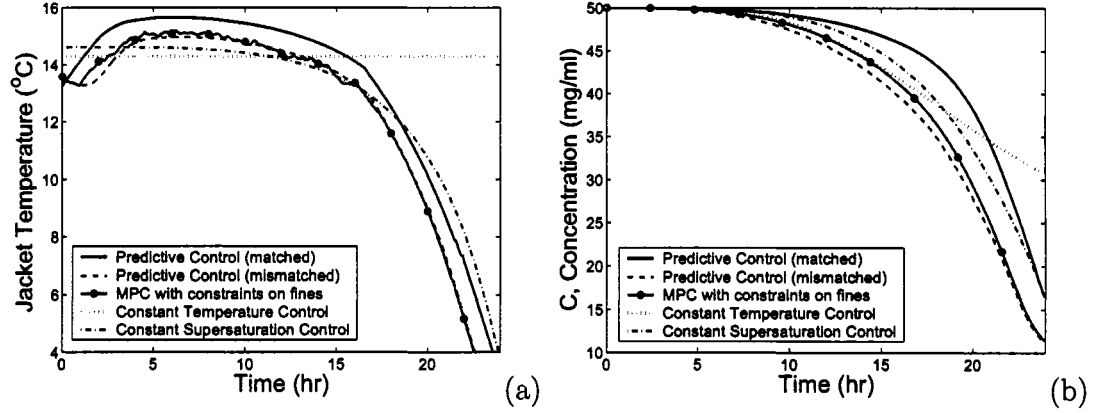


Figure 11.8: Comparison of the simulation results for (a) jacket temperature and (b) solute concentration under four different control strategies.

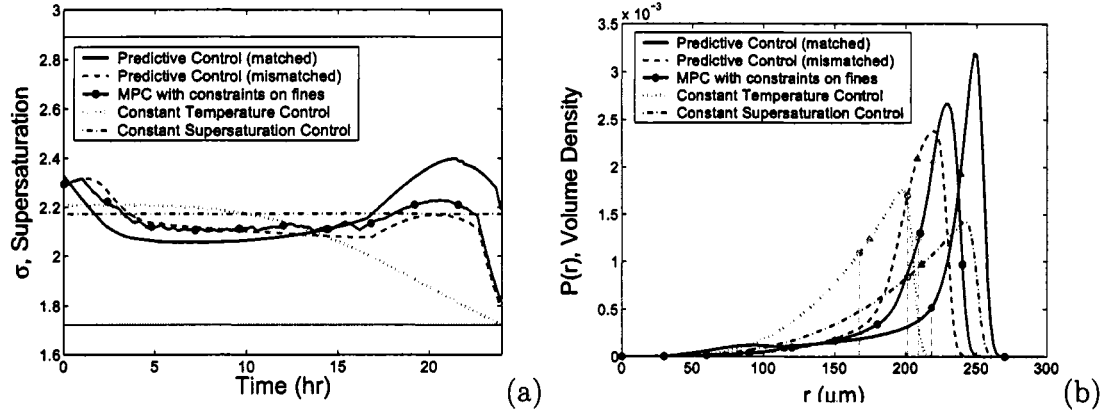


Figure 11.9: Comparison of the simulation results for (a) supersaturation and (b) volume density under four different control strategies.

In Figures 11.8 and 11.9, four important variables, the jacket temperature, the solute concentration, the supersaturation, and the volume distribution of the CSD ($P(r)$) at the end of the batch run, are shown. The volume distribution of the CSD,

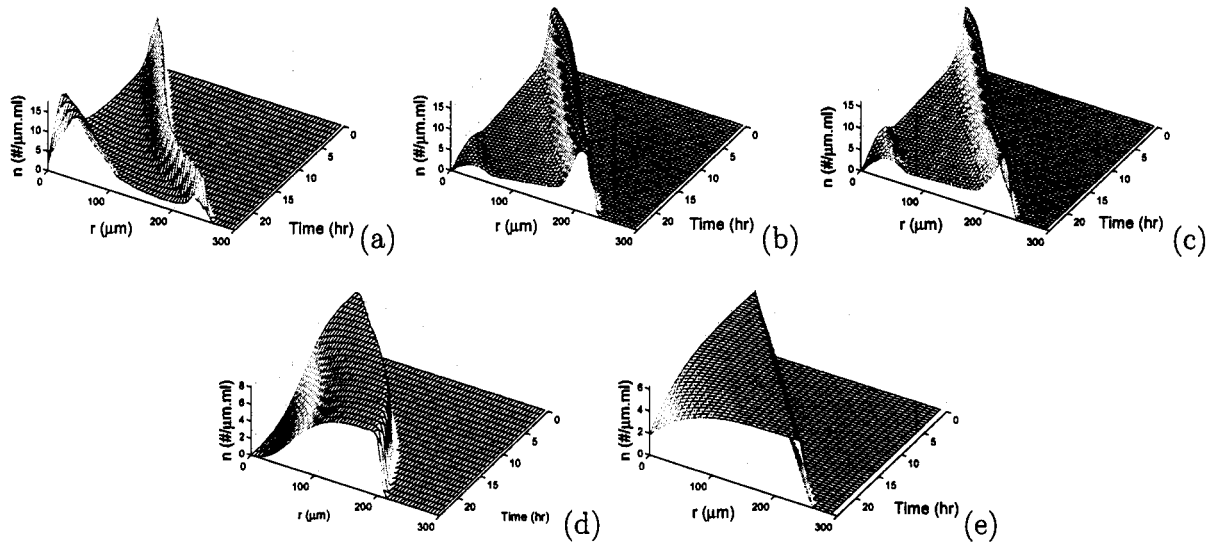


Figure 11.10: Evolution of crystal size distribution under four different control strategies: (a) Predictive control with matched model, (b) Predictive control with mismatched model, (c) Predictive control with constraint on fines, (d) constant temperature control, (e) constant supersaturation control.

$P(r)$, defined as:

$$P(r) = \frac{r^3 n(r, t_f)}{\int_0^\infty r^3 n(r, t_f) dr} \quad (11.11)$$

where t_f is the time at the end of the batch run, is an important variable in determining the product characteristics. The two horizontal lines at 1.73 and 2.89 in Figure 11.9(a) are the lower and upper constraints on the supersaturation. The evolution of the CSD under different control strategies is shown in Figure 11.10.

Figure 11.8(a) shows the comparison of the manipulated variable trajectories, the jacket temperature T_j , for the five scenarios. Note that while the trajectories of the jacket temperature under CSC (dash-dotted line) and under the predictive controller with a matched model (solid line) are close; the slight difference causes the behavior of CSD evolution under these two different control strategies to be significantly different (shown in Figure 11.10(a) and (e)), indicating the necessity to implement control to achieve a desired product quality.

Since the supersaturation σ is the main driving force for the crystal nucleation and growth, the evolution of supersaturation under various control strategies, shown in Figure 11.9(a), can be subsequently used to explain the behavior of the CSD evolution. Specifically, the solid line in Figure 11.9(a) shows that the predictive controller computes a manipulated input trajectory that results in the supersaturation first decreasing, then staying almost constant at a low level, and finally increasing at the end of the batch run. In view of the fact that the control objective is to maximize the volume-averaged crystal size at the end of the batch run, i.e., to maximize the volume of big crystals while minimizing the volume of small crystals, the result can be understood as follows: after high initial levels of supersaturation, that results in a number of nuclei formed, a drop in the supersaturation results in the nucleation rate dropping more drastically than the growth rate (note that while the nucleation rate in Eq.11.2 shows exponential dependence on the supersaturation, the growth rate in Eq.11.3 exhibits a power law dependence), and favors growth of the initially formed crystals instead of

formation of new crystals. Towards the end of the batch run, it is advantageous to increase the growth rate (by increasing the supersaturation) to maximize the size of the crystals in the batch reactor, even at the cost of forming a small number of new nuclei because the net result is a favorable increase in the volume-averaged crystal size.

Note that towards the end of the batch run, the solute concentration has depleted due to the formation of crystals. Supersaturation, however, is the solute concentration in excess of the solubility, and may be increased by decreasing the solubility, which is a function of the reactor temperature. Therefore, the desired increase in the supersaturation at the end of the batch run is achieved by lowering the jacket temperature (to lower the reactor temperature, which lowers the solubility, and hence, increases the supersaturation), until it hits the lower constraint (4 °C) on the jacket temperature.

The closed-loop simulation results for the case of a process-model mismatch are shown by the dashed line in Figures 11.8, 11.9 and 11.10(b). Comparing Figure 11.10(a) and Figure 11.10(b), it is clear that through feedback, the predictive controller ‘corrects’ for plant-model mismatch and produces a different manipulated input trajectory to enforce constraint satisfaction and achieve the desired objective. Simulation results, shown in Figure 11.7, also demonstrate that if the off-line optimized trajectory of jacket temperature, based on the mismatched model, is implemented for the entire duration of the batch run without using the measurements to update the control trajectory, the state constraints are violated at the end of the batch run underscoring the importance of feedback to achieve robustness with respect to modeling errors.

Note that predictive control without constraint on fines can result in a product with a large number of fines (see Figure 11.10(a)) which is undesirable. The implementation of the predictive controller with the constraint of Eq.11.10, designed to reduce the fines in the product, results in a product with much less fines while still maximizing the volume-averaged crystal size (see Figure 11.10(c) and Table 11.3).

Compared to predictive control, the crystal size distribution evolves very differently

under CTC and CSC. Under CTC, where a constant reactor temperature is maintained throughout the batch run, the control strategy results in a constant solubility during the batch run, and the depletion in the solute concentration is reflected in the decrease of supersaturation, which eventually hits its lower constraint (the dotted line in Figure 11.9(a)). In contrast, under CSC, which tries to maintain a constant value of the supersaturation during the entire batch run, the reactor temperature is lowered during the entire batch run (via the continued lowering of the jacket temperature) to keep up with the falling concentration levels (see the dash-dotted line in Figures 11.8(a) and 11.9(a)). Under CSC, therefore, the growth rate stays constant during the batch run, since it is only dependent on the supersaturation. The nucleation rate, which also depends directly on the solute concentration itself, is lowered gradually because of the depletion of the solute. Unlike the CSD evolution under predictive control, in the case of CTC and CSC, a constant number of new nuclei are continuously formed during the whole batch run until the depletion of the solute and this leads to a relatively low volume-averaged crystal size compared to the case of crystallizer operation under predictive control.

Table 11.3 summarizes the simulation results obtained under the five different control strategies: (1) Predictive control with a matched model, (2) Predictive control with a mismatched model, (3) Predictive control with constraint on fines, (4) Open-loop operation under CTC, and (5) Open-loop operation under CSC. In Table 11.3, six characteristic parameters of the product at the end of the batch run are compared, including the value of the volume-averaged crystal size (μ_4/μ_3), total volume (μ_3), r_{10} , r_{50} , r_{90} and the span. r_{10} , r_{50} and r_{90} are the 10%, 50% and 90% volume fractions of the CSD, respectively, denoting the percentage of crystals smaller than that size. The span, defined as $(r_{90} - r_{10})/r_{50}$, is an important characteristic of the CSD, and is widely used in the pharmaceutical industry. A high span value indicates a wide distribution in size and a high polydispersity, which is undesirable.

Comparing the results of the five control strategies listed in Table 11.3, it is clear that the predictive controller increases the volume-averaged crystal size by 30% compared to CTC, and 8.5% compared to CSC. It is also observed that, in the case of process-model mismatch (a 10% error of the exponent relating growth rate to supersaturation), the predictive control strategy is able to increase the volume-averaged crystal size by 20% compared to CTC. Although the predictive controller with a mismatched model results in a similar volume-averaged crystal size compared to CSC, it leads to a much higher volume of the product, 103% compared to CTC and 16% compared to CSC. Note also that the CSD under predictive control has the largest r_{10} , r_{50} and r_{90} . Figure 11.9(b) shows that the CSDs under the proposed predictive controller with perfect model (solid line), plant-model mismatch (dashed line) and constraint on fines (solid line marked with circles) have lower polydispersity. A lower span and much larger volume-averaged crystal size are achieved under predictive control with constraint on fines compared to the one under CTC and CSC, as shown in Figure 11.10 and Table 11.3. In summary, the implementation of the proposed predictive controller increases the volume-averaged crystal size, satisfies state and input constraints, and is found to be robust with respect to plant-model mismatch.

Table 11.3: Comparison between the simulation results under five different control strategies.

Control Strategy	μ_4/μ_3	μ_3	r_{10}	r_{50}	r_{90}	Span
	μm	$\times 10^3$	μm	μm	μm	
Predictive control (matched)	218	5.589	143	239	253	0.463
Predictive control (mismatched)	201	6.459	167	208	226	0.284
Predictive control with constraint on fines	208	6.374	166	219	235	0.315
Open-loop CTC	168	3.186	123	175	201	0.445
Open-loop CSC	201	5.583	143	211	244	0.478

11.6 Conclusions

In this chapter, we considered the problem of modeling, simulation and control of a batch protein crystallization process that is used to produce the tetragonal HEW lysozyme crystals. First, a population balance model, using empirically obtained rate expressions, was developed to simulate the batch crystallization process. Then, model reduction techniques were used to derive a reduced-order moments model for the purpose of controller design. Online measurements of the solute concentration and reactor temperature were assumed to be available, and a Luenberger-type observer was developed to estimate the moments of the crystal size. A predictive controller, that uses these state estimates, was designed to achieve the objective of maximizing the volume-averaged crystal size while respecting constraints on the manipulated input variables (which reflect physical limitations of control actuators) and on the process state variables (which reflect performance considerations). Simulation results showed that the proposed predictive controller was able to increase the volume-averaged crystal size by 30% and 8.5% compared to Constant Temperature Control (CTC) and Constant Supersaturation Control (CSC) strategies, respectively, while reducing the number of fine crystals produced. Furthermore, a comparison of the crystal size distributions (CSDs) indicated that the product achieved by the proposed predictive control strategy has larger total volume and lower polydispersity compared to the CTC and CSC strategies. Finally, the robustness of the proposed method with respect to plant-model mismatch was evaluated. In contrast to open-loop control strategies, which led to violation of constraints in the presence of plant-model mismatch, the proposed method was demonstrated to successfully achieve the task of maximizing the volume-averaged crystal size while satisfying the state and input constraints.

Chapter 12

A Method for PID Controller Tuning Using Nonlinear Control Techniques

12.1 Introduction

The majority (over 90%) of the regulatory loops in the process industries use conventional Proportional-Integral-Derivative (PID) controllers. Owing to the abundance of PID controllers in practice and the varied nature of processes that the PID controllers regulate, extensive research studies have been dedicated to the analysis of the closed-loop properties of PID controllers and to devising new and improved tuning guidelines for them, focusing on closed-loop stability, performance and robustness (see, for example, [246, 188, 224, 120, 162, 210, 11], the survey papers [12, 42] and books [154, 211, 138]). Most of the tuning rules are based on obtaining linear models of the system, either through running step tests or by linearizing a nonlinear model around the operating steady-state, and then computing values of the controller parameters that incorporate stability, performance and robustness objectives in the closed-loop system.

While the use of linear models for the PID controller tuning makes the tuning process easy, the underlying dynamics of many processes are often highly complex due, for example, to the inherent nonlinearity of the underlying chemical reaction or due to operating issues such as actuator constraints, time-delays and disturbances. Ignoring the inherent nonlinearity of the process when setting the values of the controller parameters may result in the controller's inability to stabilize the closed-loop system and may call for extensive re-tuning of the controller parameters.

The shortcomings of classical controllers in dealing with complex process dynamics, together with the abundance of such complexities in modern-day processes, have been an important driving force behind the significant and growing body of research work within the area of nonlinear process control over the past two decades, leading to the development of several practically-implementable nonlinear control strategies that can deal effectively with a wide range of process control problems such as nonlinearities, constraints, uncertainties, and time-delays (see, for example, [7, 66, 69, 127, 4] and the books [207, 80, 112, 101, 38]). While process control practice has the potential to benefit from these advances through the direct implementation of the developed nonlinear controllers, an equally important direction in which process control practice stands to gain from these developments lies in investigating how nonlinear control techniques can be utilized for the improved tuning of classical PID controllers. This is an appealing goal because it allows control engineers to potentially take advantage of the improved stability and performance properties provided by nonlinear control without actually forsaking the ubiquitous conventional PID controllers or re-designing the control system hardware.

There has been some research effort towards incorporating nonlinear control tools in the design of PID controllers. For example, in [232] it is shown that controllers resulting from nonlinear model-based control theory can be put in a form that looks like the PI or PID controllers for first and second-order systems. Other examples include [20]

where adaptive PID controllers are designed using a backstepping procedure, and [29] where a self-tuning PID controller is derived using Lyapunov techniques. In these works, however, even though the resulting controller has the same structure as that of a PID controller, the controller parameters (gain: K_c , integral time-constant: τ_I , and derivative time-constant: τ_D) are not constant but functions of the error or process states. While such analysis provides useful analogies between nonlinear controllers and PID controllers, implementation of these control designs would require changing the control hardware in a way that allows the tuning parameter values to be continuously changed while the process is in operation.

Motivated by the above considerations, we propose in this chapter a two-level, optimization-based method for the derivation of tuning guidelines for PID controllers that take nonlinear process behavior explicitly into account. The central idea behind the proposed method is the selection of the tuning parameters in a way that has the PID controller emulate, as closely as possible, the control action and closed-loop response obtained under a given nonlinear controller, for a broad set of initial conditions and set-point changes. To this end, classical tuning guidelines (typically derived on the basis of linear approximations, running open or closed-loop tests) are initially used in the first level to obtain reasonable bounds on the range of stabilizing tuning parameters over which the search for the parameters best matching the PID and nonlinear controllers is to be conducted. In addition to stability, performance and robustness considerations for the linearized closed-loop system can be introduced in the first level to further narrow down the parameter search range. The bounds obtained from the first level are then incorporated as constraints on the optimization problem solved at the second level to yield a set of tuning parameter values that enforce closed-loop behavior under the PID controller that closely matches the closed-loop behavior under the nonlinear controller. Implications of the proposed method, as a transparent and meaningful link between the classical and nonlinear control domains, as well as possible extensions of the tuning guidelines and other implementation issues are discussed. Finally, the

implementation of the proposed tuning method is demonstrated through a nonlinear chemical reactor example.

12.2 Two-level PID Tuning Method

In this chapter, we consider continuous-time single-input single-output (SISO) nonlinear systems, with the following state-space description

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\ y &= h(x)\end{aligned}\tag{12.1}$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables and x' denotes the transpose of x , $y \in \mathbb{R}$ is the controlled output, $u \in \mathbb{R}$ is the manipulated input, $f(\cdot)$ is a sufficiently smooth nonlinear vector function with $f(0) = 0$, $g(\cdot)$ is a sufficiently smooth nonlinear vector function and $h(\cdot)$ is a sufficiently smooth scalar nonlinear function with $h(0) = 0$. Throughout the paper, the notation $L_f h$ denotes the standard Lie derivative of a scalar function $h(\cdot)$ with respect to the vector function $f(\cdot)$, i.e., $L_f h(x) = (\partial h / \partial x) f(x)$.

The basic idea behind the proposed approach is the design (but not implementation) of a nonlinear controller that achieves the desired closed-loop response, and then the tuning of the PID controller parameters so as to best “emulate” the control action and the closed-loop process response under the nonlinear controller, subject to constraints derived from classical PID controller tuning rules.

First, we use classical PID tuning guidelines to come up with reasonable ranges for the PID controller tuning parameters. Then we compute off-line, through simulation, the control action of the nonlinear controller over the time that it takes to practically achieve the set-point change, and the corresponding process response. The PID controller tuning parameters are then computed by solving (off-line) an optimization problem that minimizes some measure of the difference between the control action and the closed-loop process response under the nonlinear controller on one hand, and

the control action and closed-loop response obtained under the PID controller on the other, for a given initial condition and set-point change. The optimization is carried out subject to constraints that ensure that the PID controller tuning parameters are within acceptable ranges of the values derived from the classical tuning rules. These ideas are described algorithmically below (see also Fig.12.1):

1. Construct a nonlinear process model and derive a linear model around the operating steady-state (either through linearization or by running step tests).
2. On the basis of the linear model, use classical tuning guidelines to determine bounds on the stabilizing range of the tuning parameters, K_c , τ_I and τ_D .
3. Using the nonlinear process model and the desired closed-loop response, design a nonlinear controller.
4. For a set-point change, compute off-line, through simulations, the input trajectory ($u_{nl}(t)$) “prescribed” by the nonlinear controller over the time interval, $[0, t_{final}]$, required to achieve the set-point change. Compute the corresponding closed-loop output profile under the nonlinear controller, $y_{nl}(t)$.
5. Compute the PID controller tuning parameters, K_c , τ_I and τ_D , as the solution to the following optimization problem

$$\min_{K_c, \tau_I, \tau_D} J = \int_0^{t_{final}} [(y_{nl}(t) - y_{PID}(t))^2 + (u_{nl}(t) - u_{PID}(t))^2] dt \quad (12.2)$$

$$\begin{aligned}
s.t. \ u_{PID}(t) &= K_c \left(e + \frac{\int_0^t e(t') dt'}{\tau_I} + \tau_D \frac{de}{dt} \right) \\
e(t) &= y_{sp} - y_{PID} \\
\dot{x}(t) &= f(x(t)) + g(x(t))u_{PID}(t) \\
y_{PID} &= h(x) \\
\alpha_1 K_c^c &\leq K_c \leq \alpha_4 K_c^c \\
\alpha_2 \tau_I^c &\leq \tau_I \leq \alpha_5 \tau_I^c \\
\alpha_3 \tau_D^c &\leq \tau_D \leq \alpha_6 \tau_D^c \\
(K_c, \tau_I, \tau_D) &= \arg \min(J)
\end{aligned} \tag{12.3}$$

where y_{sp} is the desired set-point, y_{nl} , u_{nl} are the closed-loop process response and control action, respectively, under the nonlinear controller, K_c^c , τ_I^c and τ_D^c are parameter values obtained using (first level) tuning rules based on linear models, and $0 \leq \alpha_i < 1$, $i = 1, 2, 3$ and $1 < \alpha_i < \infty$, $i = 4, 5, 6$ are design parameters.

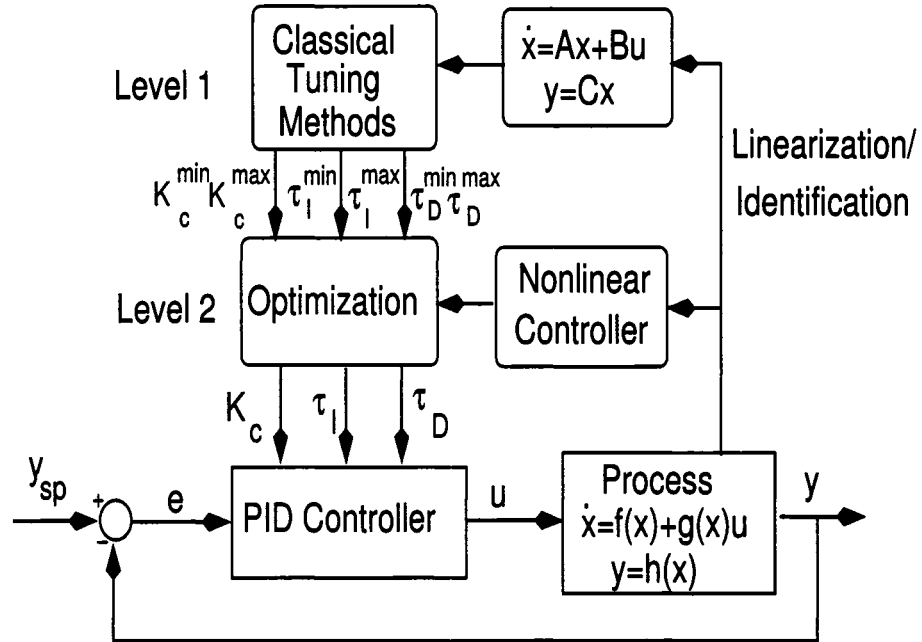


Figure 12.1: Schematic representation of the implementation of the proposed two-level, optimization-based tuning method.

Remark 12.1: The optimization problem of Eqs.12.2-12.3 computes values for K_c , τ_I ,

τ_D such that the closed-loop control action and process response under the PID controller are similar to those under the nonlinear controller, while being within acceptable ranges of the values obtained from the classical tuning methods. The method, therefore, allows fine-tuning the closed-loop performance under the PID controller to mimic that of the nonlinear controller. Note that, in principle, the PID controller parameters could have been tuned to mimic any desired arbitrarily-chosen closed-loop behavior. Mimicking the behavior of the nonlinear controller, however, is a meaningful objective since the nonlinear controller utilizes the nonlinear system dynamics in generating the prescribed input and output response, and therefore provides a “target” closed-loop system behavior that is realizable. Note also that the performance index of Eq.12.2 can be generalized to include a weight factor (in front of the input penalty term) in order to quantify the relative importance of the two main terms: the first one that aims at minimizing the closed-loop output mismatch associated with the nonlinear vs. the PID controller, and the second input penalty term. Furthermore, the inclusion of a weight factor would ensure consistency of units for the two terms in the performance functional.

Remark 12.2: It should be noted that the controller actually implemented in the closed-loop system is the PID controller with fixed parameter values and that it may not always be possible for the PID controller to exactly match the closed-loop behavior under the nonlinear controller. The purpose behind introducing the first-level tuning is twofold: (1) to ensure that important objectives (such as closed-loop stability) are not sacrificed in the (possibly unsuccessful) quest for a nonlinear controller like behavior (which is accomplished through the optimization), and (2) to provide a rational way of constructing a range of the tuning parameter values over which the optimization is performed. In relation to the first objective, we note that the essence of the second-level optimization is to try to find the best tuning parameter values that make the PID controller emulate the behavior of the nonlinear controller. However, this objective should not come at the expense of more overriding objectives such as closed-loop

stability. In particular, if the optimization problem were to be carried out without imposing any constraints on the parameter values, the solution may indeed lead to a closer match, but no longer guarantee that the PID controller enforces closed-loop stability when implemented. This is one reason why the proposed method includes the first level whose purpose, in part, is to make use of existing methods for PID controller tuning in order to first determine the range of tuning parameters for which closed-loop stability of the linearized process model under the PID controller is guaranteed. For such a range of values, the PID controller is expected to enforce (local) closed-loop stability when implemented on the nonlinear process.

Remark 12.3: Having obtained the stabilizing parameter range and incorporated it as a constraint on the optimization, the search for the “optimal” gain then can take place only over this range. However, the stabilizing range may be too large to search over and the designer may wish to limit this range further by incorporating additional performance and robustness considerations. The use of first-level methods (which are based on the use of linear models) provides a rational, though not necessarily unique, way of constructing an appropriate subrange to work with. For example, if certain robustness margins can be obtained (and quantified explicitly as ranges on the tuning parameters) through the use of existing methods based on the linearized model, these margins can be incorporated as constraints that further limit the search range. This of course does not (nor is intended to) guarantee that the PID controller will exhibit such robustness or performance when implemented on the nonlinear process since the margins are based on the linearized model. However, at a minimum, this is a meaningful way to go about constructing or, more precisely, narrowing down the range over which the optimization is done (by requesting that the performance of the linearized model be consistent with what existing methods based on linear models yield). Ultimately, it is the “closeness” of the PID controller to the nonlinear controller resulting from the second-level optimization (not the first-level methods) that is essentially responsible for the performance properties exhibited by the PID controller when implemented on

the nonlinear process. Since different (first-level) tuning methods lead to different performance properties and yield different parameter values, the designer can examine the values obtained from different methods to form a reasonable idea about what an acceptable range might be around these nominal values and then construct such a range (through choosing appropriate α_i values) and implement it as constraints on the optimization (see the simulation study for an example). If the parameter values obtained (after performing the optimization) do not yield satisfactory performance (tested through simulations), then the parameter range could be expanded further (but still within the stabilizing range determined initially) in an iterative procedure.

Remark 12.4: Note that the tuning methodology is not proposed as an *alternative* to existing tuning guidelines, but one that serves to *improve* upon existing ones by imbuing them with desirable features provided by nonlinear control tools. To this end, the α_i 's in Eq.12.3 are introduced into the optimization as design parameters that allow the designer flexibility in tightening or relaxing the range of parameter values over which the optimization is carried out. If a given method yields satisfactory performance, and it is desired that the tuning parameters not be changed appreciably, this can be enforced by using values of the design parameters, α_i , close to 1. The tuning parameters resulting from the solution to the optimization problem in this case, while changed to mimic the nonlinear control action, will be close to the ones obtained from the classical tuning method considered in the first level. If, on the other hand, it is decided that some further improvement is warranted, then, at a minimum, the α_i values should be chosen to reflect the range (or a subset of the range) within which the parameter values can be changed by the optimization without losing stability. If the designer wishes to constrain the search over a smaller range (using, for example, certain performance or robustness margins obtained from the first-level methods for the linearized closed-loop system), then the α_i values can be modified to reflect the new range. In general, the choice of α_i varies depending on the particular process under consideration and on the particular tuning methods that are being considered in the first level. For example,

when considering a combination of different approaches, one could choose the α_i values so as to construct a range within which some desirable or acceptable tradeoff between the performance of those approaches can be achieved for the linearized closed-loop system. Different approaches will yield different sets of critical parameters below which or above which certain performance characteristics could be unsatisfactory. Examining the overlap of the different ranges can guide the designer's choice of the α_i values in a way that creates a parameter range that accommodates an acceptable degree of tradeoff between the objectives of the different approaches.

Remark 12.5: Regarding the performance properties of the proposed method in relation to those of the first-level methods, we first note that the first-level tuning guidelines are derived on the basis of the linearized process model, and therefore the robustness and performance properties obtained when using these methods to tune the PID controller are not guaranteed to carry over when the PID controller is implemented on the nonlinear system. So even if re-tuning of the first-level methods may bring about further performance improvement in the linear case (by possibly sacrificing stability and robustness margins), this does not imply that a similar improvement should be expected in the nonlinear setting. In general, there is no systematic way in which such methods can be re-tuned (if at all possible) to improve the performance in the nonlinear case. By contrast, the proposed method aims to improve the performance of the PID controller in the nonlinear setting by explicitly accounting for process nonlinearity through the optimization (an objective not shared by the first-level approaches). Whether this necessarily means that the resulting parameters will always yield performance that is “better” than what a given tuning method might yield is difficult to judge, but, more importantly, is not a point that the method is intended to address. The point is that the proposed approach is a meaningful way of tuning PID controllers that can yield good performance when implemented on the nonlinear system. From this perspective, the first-level methods are not intended as a point of comparison (even though the simulation study shows that it is possible to improve upon them using the proposed

method mainly because of the added feature of accounting for the nonlinearity through the optimization) but rather as a rational starting point for the construction of the search ranges as discussed in Remark 12.2.

Remark 12.6: The optimization problem of Eqs.12.2-12.3 is solved off-line as part of the design procedure to compute the optimal values of the tuning parameters. Also, the above optimization problem can be carried out over a range of initial conditions and set-point changes that are locally representative of the process operation to obtain PID tuning parameters that allow the PID controller to approximate, in an average (with respect to initial conditions and set-point changes) sense, the closed-loop response under the nonlinear controller. If the process is required to operate at an equilibrium point that is very far from the operating point that the parameters are tuned for, then it is best to perform the optimization again around the new, desired operating point to yield new tuning parameter values (as is done in classical tuning also).

Remark 12.7: The proposed tuning method is different from other methods that propose a nonlinear PID controller (see, for example, [232, 20]), which require changing the hardware of the PID controller because, in these designs, K_c , τ_I and τ_D are not constant, but instead are functions of the error or process states, and therefore need to be continuously changed while the process is in operation. The tuning framework proposed in this chapter, while using nonlinear control techniques, provides tuning guidelines for *existing* PID controllers, by providing values for the controller parameters that stay fixed during implementation, and does not require redesigning or replacing the PID controller hardware.

Remark 12.8: The basic idea behind the proposed PID controller tuning methodology, i.e., that of tuning the PID controller to emulate some other well-designed controller that handles complex dynamics effectively, can be used to develop conceptually similar tuning methods for PID control of processes with other sources of complexities (besides nonlinearity) such as uncertainty, time-delays, and manipulated input con-

straints. The logic behind such extensions is based on the following intuitive parallel: just as a nonlinear controller is a meaningful guide to be emulated by a PID controller being implemented on a nonlinear process, a controller that handles constraints, uncertainty and/or time-delays effectively can also be a meaningful guide to be emulated by a PID controller that is being implemented on a process with these characteristics. In principle, the extensions can be realized by adequately accounting for the complex characteristics of these processes within both levels of the tuning method. For example, for systems with uncertainty, classical tuning methods that provide sufficient robustness margins can be used to come up with the first-level parameter values. Then a robust nonlinear controller (for example, [66]) can be designed and the closed-loop profiles, obtained under the robust nonlinear controller for a sufficient number of realizations of the uncertainty (which may be simulated, for instance, using random number generators) may be computed. Finally, the parameter values obtained from the first level tuning method may be improved upon by solving an optimization problem that minimizes the error over the profiles in the representative set. In a conceptually similar fashion, for systems with constraints, an anti-windup scheme could be used initially to obtain the first level parameter values. A nonlinear controller design that handles input constraints can then be chosen for the second-level optimization. The PID controller tuning guidelines can then serve to “carry over” the constraint handling properties of this nonlinear controller and improve upon the first level tuning methods in two ways: (1) through the objective function, by requiring the control action and closed-loop process response under PID control to mimic that under the constrained nonlinear controller, and (2) through the incorporation of the input constraints directly into the optimization problem. It should be noted however that, while such extensions are intuitively appealing, a detailed assessment and characterization of their potential requires further investigation that is beyond the scope of the current work.

Remark 12.9: The nature of the operating conditions should be considered when deciding whether to tune the parameter values of a PI or PID controller to mimic

the response of the nonlinear controller. When the closed-loop system is required to operate around an unstable steady state, under certain conditions, the PI controller may not be able to stabilize the closed-loop system ([168]). In these cases, a PID controller must be chosen. Also, adequate stability constraints may be imposed when solving the optimization problem. For instance, constraints that ensure the satisfaction of the Routh stability test (based on linearization of the process around the desired operating condition) may be incorporated as part of the optimization problem.

Remark 12.10: Note that the derivative part of the PID controller is often implemented using a filter. This feature can be easily incorporated in the optimization problem by explicitly accounting for the filter dynamics. Constraints on the filter time-constant, τ_f , obtained empirically through knowledge of the nature of noise in the process, can be imposed to ensure that the filtering action restricts the process noise from being transmitted to the control action.

Remark 12.11: To allow for simple computations, approximations can be introduced in solving the optimization problem of Eqs.12.2-12.3. For instance, in the computation of the control action, the error, $e(t)$, may be approximated by simply taking the difference between the set-point, y_{sp} , and the process output under the nonlinear controller, $y_{nl}(t)$, leading to a simpler optimization problem that can be solved easily using numerical solvers such as Microsoft Excel (for a given choice of the decision variables, the objective function can be computed algebraically and does not involve integrating the process dynamics). The justification behind this being that if the resulting value of $u_{PID}(t)$ is “close enough” to $u_{nl}(t)$, then this approximation holds (see the simulation example for a demonstration). If the solution of the optimization problem does not yield a sufficiently small value for the objective function (indicating that u_{PID} , and hence y_{PID} , is significantly different from u_{nl} , and y_{nl}), this approximation may not be valid anymore. In this case, one could revert to using $e(t) = y_{sp} - y_{PID}(t)$ in the optimization problem, where y_{PID} is the closed-loop process response under the PID

controller. Note also that, in some cases (particularly for low-dimensional systems with real analytic vector fields), the value of the performance index may be calculated explicitly as an algebraic function of the controller parameters (leading to a static finite-dimensional optimization problem) by solving Zubov's partial differential equation using techniques similar to those presented in [109]. Those techniques can also be used in designing an optimally-tuned nonlinear controller that serves as a meaningful target to be emulated by the PID controller.

Remark 12.12: The proposed method involves solving a nonlinear dynamic optimization problem and is therefore expected to exhibit some of the implementation issues and properties associated with nonlinear optimization problems (even though certain features of the problem formulation distinguish it from more general nonlinear dynamic optimization problems as explained below). Regarding the optimization complexity issue, for example, we note that possible complexity of the proposed optimization is mainly a function of the model complexity (e.g., nonlinearity, model order, etc.). The increase in computational demand expected in the case of higher-order and highly nonlinear systems is primarily due to the need to solve a higher-order system of nonlinear differential equations. However, with current computational capabilities, this does not pose unduely significant limitations on the practical implementation prospects of the proposed method especially when compared with the computational complexity encountered in typical nonlinear optimization problems (e.g., nonlinear MPC) where not only does a nonlinear program have to be solved repeatedly online, but also the number of decision variables increases significantly with the optimization horizon. By its fundamentally different formulation, the proposed optimization problem involves only three decision variables regardless of the time horizon over which the cost functional is computed and regardless of the model order. The input and output profiles associated with the nonlinear controller are computed (using existing nonlinear control and computations methods) prior to performing the optimization. During the optimization, the optimizer needs to solve the nonlinear closed-loop state equations under the

PID controller in order to search for the best tuning parameters. Other features of the proposed optimization problem (such as the fact that the optimization is carried out off-line, is not done repeatedly and involves a fixed small number of decision variables) make it simpler than what currently available computational tools are capable of handling. Furthermore, the approximations discussed in Remark 12.9 provide possible means that can help manage potential complexities even further. The main objective of the proposed method is to provide a transparent and insightful link between classical and nonlinear control tools. This appealing objective, together with the availability of enhanced computational capabilities, motivate real interest in the practical implementation of the aforementioned ideas for optimal tuning of PID controllers using nonlinear control guidelines. Finally, we note that since the method involves a form of nonlinear optimization, it is expected that, in general, multiple optimal solutions (sets of parameters) may exist. However, as with general nonlinear optimization problems, a rigorous characterization of the existence and uniqueness of solutions is difficult to obtain (and is beyond the scope of the paper).

Remark 12.13: The characteristics of the closed-loop system, analyzed on the basis of the linearized model, can also be used as guidelines in setting up the optimization problem. For example, the gain and phase margins can be computed for the parameter values prescribed by the proposed method and changed if not found to be adequate. Note that the value of the gain margin depends on the value of tuning parameters which, in turn, depend on response of the nonlinear controller that the PID controller is required to mimic. A highly aggressive response prescribed by the nonlinear controller will likely lead to aggressively tuned parameter values and a possibly low gain margin. The value of the gain margin can be altered by changing the requested response of the nonlinear controller, making it less or more aggressive as the need may be.

Remark 12.14: Finally, we note that the proposed method *does not* turn the PID controller into a nonlinear controller. The tuning method can only serve to improve

upon the process response of the PID controller for operating conditions for which PID control action can be used to stabilize the process. If the process is highly nonlinear, or a complex process response is desired, it may be possible that the PID controller structure is not adequate and, in this case, the appropriate nonlinear controller should be implemented in the closed-loop to achieve the desired closed-loop properties.

12.3 Application to a Chemical Reactor Example

We consider a continuous stirred tank reactor where an irreversible, first-order reaction of the form $A \xrightarrow{k} B$ takes place. The inlet stream consists of pure species A at flow rate F , concentration C_{A0} and temperature T_{A0} . Under standard modeling assumptions, the mathematical model for the process takes the form

$$\begin{aligned}\dot{C}_A &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT_R}} C_A \\ \dot{T}_R &= \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{\frac{-E}{RT_R}} C_A + \frac{UA}{\rho c_p V}(T_j - T)\end{aligned}\tag{12.4}$$

where C_A denotes the concentration of the species A , T_R denotes the temperature of the reactor, T_j is the temperature of the fluid in the surrounding jacket, U is the heat-transfer coefficient, A is the jacket area, V is the volume of the reactor, k_0 , E , ΔH are the pre-exponential constant, the activation energy, and the enthalpy of the reaction respectively, and c_p and ρ , are the heat capacity and fluid density in the reactor respectively. The values of all process parameters are given in Table 12.1. At the nominal operating condition of $T_j^{nom} = 493.87 \text{ K}$, the reactor is operating at the unique, stable steady-state $(C_A^s, T_R^s) = (0.52 \text{ Kmol/m}^3, 398.97 \text{ K})$. The control objective is to implement set-point changes in the reactor temperature using the jacket fluid temperature, T_j , as the manipulated input, using a P, PI or PID controller.

To proceed with our controller tuning method, we initially design an input/output linearizing nonlinear controller. Note that the linearizing controller design is used in

Table 12.1:

V	$=$	0.1	m^3
E/R	$=$	8000	K
C_{A0}	$=$	1.0	$Kmol/m^3$
T_{A0}	$=$	400.0	K
ΔH	$=$	2.0×10^5	$KJ/Kmol$
k_0	$=$	7.85×10^6	s^{-1}
c_p	$=$	1.0	$KJ/Kg.K$
ρ	$=$	1000.0	Kg/m^3
UA	$=$	1.667×10^3	$KJ/s.K$
F	$=$	0.001667	m^3/s
C_A^s	$=$	0.52	$Kmol/m^3$
T_R^s	$=$	398.97	K
T_j^{nom}	$=$	493.87	K

the simulation example only for the purpose of illustration, and any other nonlinear controller design deemed fit for the problem at hand can be used as part of the proposed controller tuning method.

Defining $x = [C_A - C_A^s, T_R - T_R^s]'$ and $u = T_j - T_j^{nom}$, the process of Eq.12.4 can be recast in the form of Eq.12.1 where the explicit form of $f(\cdot)$ and $g(\cdot)$ are omitted for brevity. Consider the control law given by:

$$u = \frac{\nu - y(t) - \gamma L_f h(x)}{\gamma L_g h(x)} \quad (12.5)$$

where $L_f h(x)$ and $L_g h(x)$ are the Lie derivatives of the function $h(x)$ with respect to the vector functions $f(x)$ and $g(x)$, respectively, γ , a positive real number, is a design parameter and ν is the set-point. Taking the derivative of the output in Eq.12.1 with respect to time, we get

$$\dot{y} = L_f h(x) + L_g h(x)u \quad (12.6)$$

Substituting the linearizing control law of Eq.12.5, we get

$$\dot{y} = \frac{\nu - y}{\gamma} \quad (12.7)$$

Under the control law of Eq.12.5, the controlled output y evolves linearly, to achieve

the prescribed value of ν , with the design parameter γ being the time-constant of the closed-loop response.

It is well known that when a first-order closed-loop response, with a given time-constant, is requested for a linear first-order process, the method of direct synthesis yields a PI controller. Note that the relative order of the controller output, T_R , with respect to the manipulated input, T_j , in the example of Eq.12.4 is also one. Even though the nonlinear linearizing controller is a static controller, and the PI controller is dynamic, both controllers are capable of generating closed-loop behaviors that are of the same kind (a linear first order response with a prescribed closed-loop time constant). This motivates using a PI controller and tuning the controller parameters to achieve the prescribed first-order response.

For the purpose of tuning the PI controller, the nonlinear process response generated under the nonlinear controller, using a value of $\gamma = 0.25$, was used in the optimization problem. An appropriate range for the tuning parameters was derived from the K_c and τ_I suggested by the IMC-based and Ziegler-Nichols tuning rules (where the parameters $K_{cu} = 6.36$ and $P_u = 5.0$ are obtained using the method of relay auto tuning [10]). In particular, the constraints on the values of the parameters were chosen as follows: for a given parameter, the largest and the smallest values prescribed by the available tuning methods (in this case the IMC-based and Ziegler Nichols) were chosen and the upper bound on the parameters was chosen as twice the maximum value, and the lower bound was chosen as half the minimum value; $0.9 \leq K_c \leq 5.7$ and $0.2 \leq \tau_I \leq 8.2$. The values of the parameters, computed using the IMC method, Ziegler-Nichols and the two-level PI tuning method are reported in Table 12.2.

The solid lines in Fig.12.2(a)-(b) show the closed-loop response of the output and the manipulated input under the nonlinear control of Eq.12.5. Note that the value of γ was chosen as 0.25 to yield a smooth, fast transition to the desired set-point. The optimization problem was solved approximately, using the closed-loop process response

Table 12.2:

Tuning method	K_c	τ_I
IMC	1.81	0.403
Ziegler-Nichols	2.86	4.16
Proposed method	5.56	0.297

under the nonlinear controller to compute $e(t)$, and the objective function only included penalties on the difference between the control actions under the PI controller and the nonlinear controller (see Remark 12.7). The dashed-line shows the response of the PI controller tuned using the proposed optimization-based method. The result shows that the response under the PI controller is close to that under the nonlinear controller and demonstrates the feasibility of using a PI controller to generate a closed-loop response that mimics the response of the nonlinear controller.

In Fig.12.3(a), we present the closed-loop responses when the controller parameters computed using the IMC-based tuning rules and Ziegler-Nichols are implemented. As can be seen, the transition to the new set-point under the PID controller tuned using the proposed method (dashed lines) is fastest when compared to a classical PI controller tuned using IMC tuning rules (solid line) and Ziegler-Nichols tuning rules (dotted line). The corresponding manipulated input profiles are shown in Fig.12.3(b).

We now demonstrate the application of the proposed method to the same system, but with C_A as the controlled variable and T_j as the manipulated variable. As in the previous case, we initially design an input/output linearizing nonlinear controller to yield a second-order linear input-output response in the closed-loop system of the form:

$$\tau_d^2 \ddot{y} + \frac{2\xi}{\tau_d} \dot{y} + y = \nu \quad (12.8)$$

where τ_d and ξ are design parameters and were chosen as $\tau_d = 0.2$ and $\xi = 1.05$ (implying that the closed-loop system is a slightly over-damped second-order system).

The following tuning methods were used for the first level: (1) IMC-based tuning

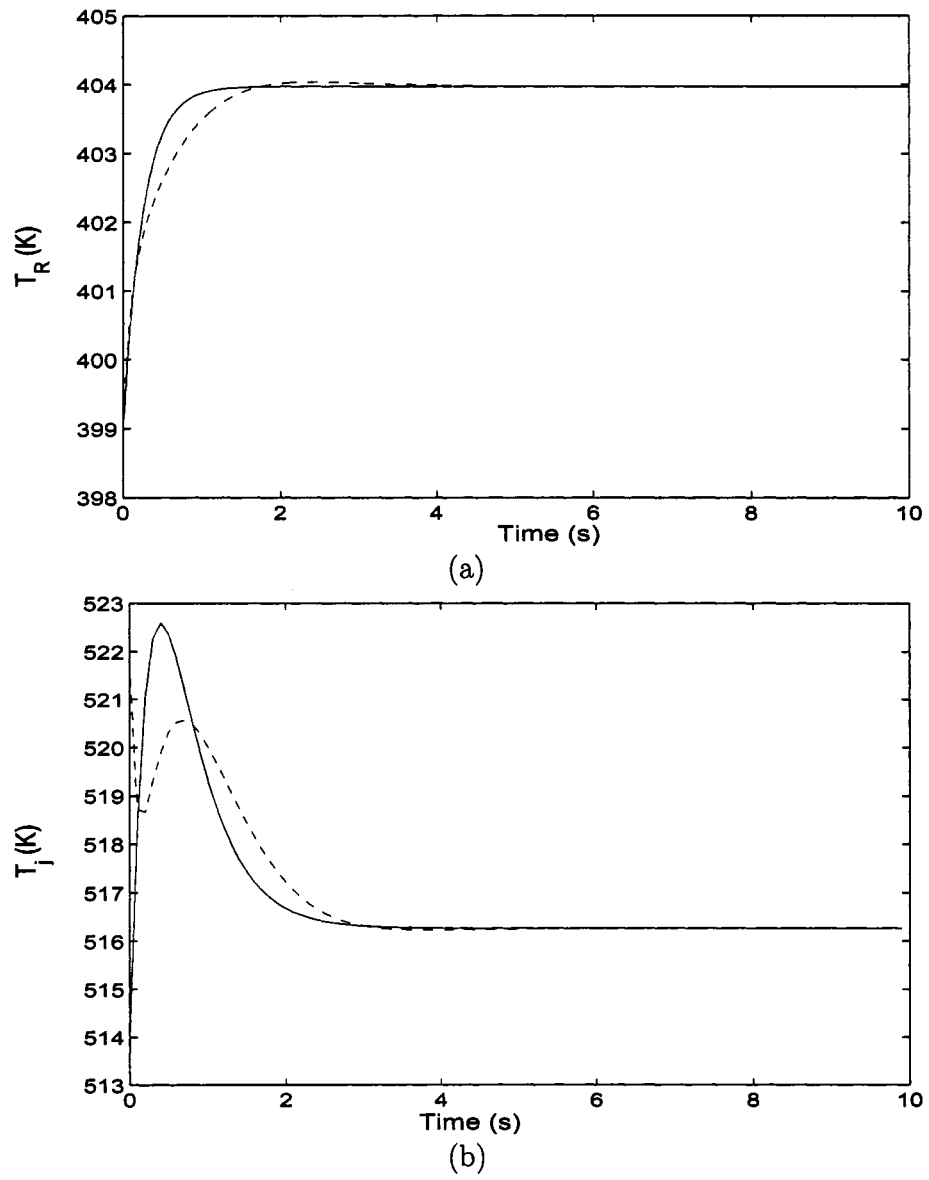


Figure 12.2: Closed-loop output (a) and manipulated input (b) profile under a linearizing controller (solid line) and a PI controller tuned using the proposed method (dashed line).

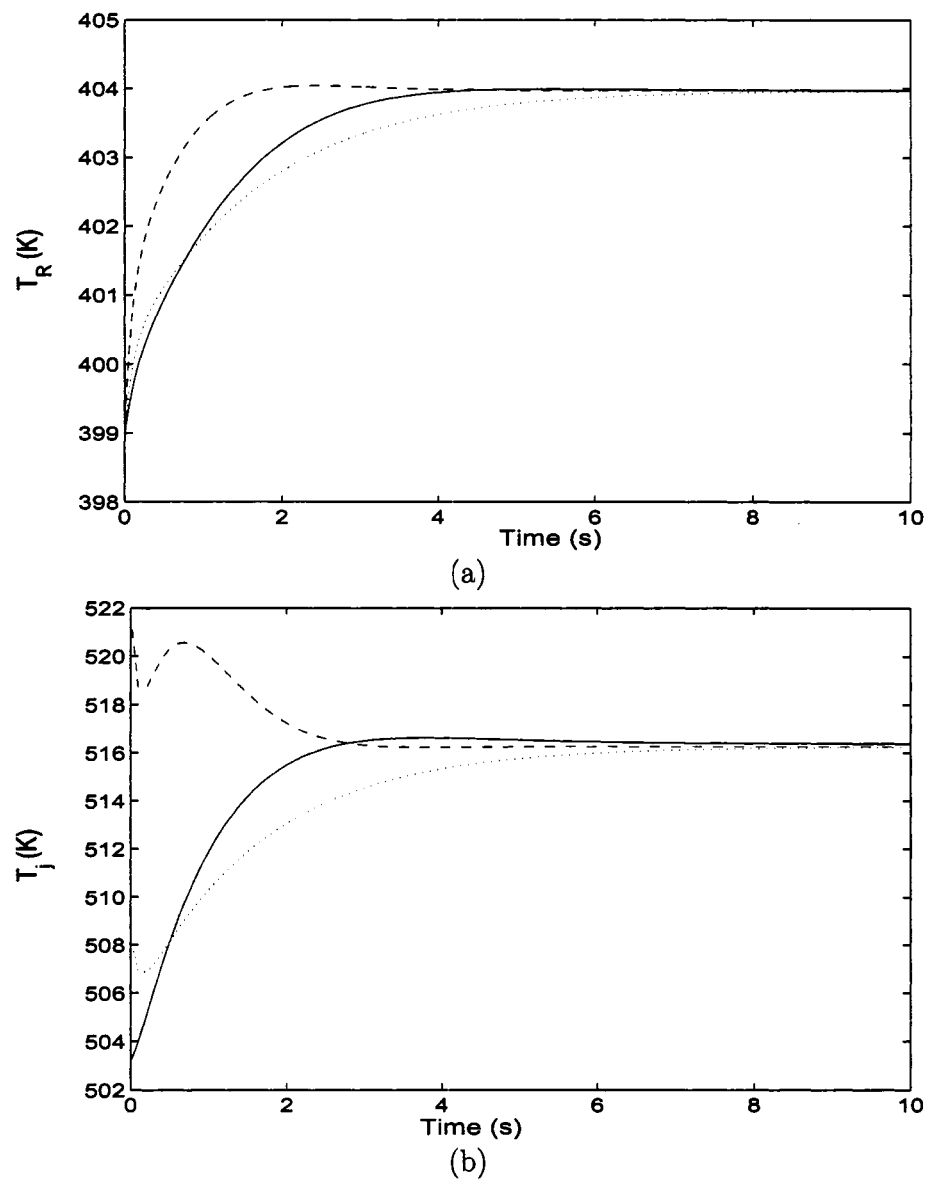


Figure 12.3: Closed-loop output (a) and manipulated input (b) profile using IMC tuning rules for PI controller (solid line), using Ziegler-Nichols tuning rules (dotted line) and the proposed method (dashed line).

rule, where a step test is run to approximate the system by a first-order + time-delay process, hence referred to as IMC-I, (2) IMC-based tuning rule, where the process is linearized around the operating steady-state to obtain a second-order linear model, hence referred to as IMC-II, and (3) Ziegler-Nichols tuning rules, where the parameters $K_{cu} = -34543$ and $P_u = 0.223$ are obtained using the method of relay auto tuning [10] (the tuning parameter values are reported in Table 12.3). Based on the parameter ranges suggested by the first level tuning methods, the following constraints were used in the optimization problem set up to compute K_c , τ_I and τ_D : $-2072.0 \leq K_c \leq -678$, $0.149 \leq \tau_I \leq 1$ and $0.0114 \leq \tau_D \leq 0.052$. The derivative part of the controller was implemented using a first order filter with time-constant $\tau_f = 0.1$.

Table 12.3:

Tuning method	K_c	τ_I	τ_D
IMC-I	-678.8	0.95	0.0524
IMC-II	-1208.9	1.00	0.114
Ziegler-Nichols	-2072.0	0.149	0.028
Proposed method	-951.21	0.978	0.114

The solid lines in Figs.12.4(a)-(b) show the closed-loop response of the output and the manipulated input under the linearizing control design. The dashed-line shows the response of the PID controller tuned using the proposed method, which is close to the response of the nonlinear controller. As is clear from Fig.12.4, the resulting PID controller yields a response that is close enough to that of the nonlinear controller.

In Fig.12.5(a), we present the closed-loop responses when the controller parameters computed using the classical tuning rules are implemented. The values suggested by Ziegler-Nichols tuning lead to closed-loop instability. In the simulation, a smaller value for $K_c = -518.4$ and a larger $\tau_I = .14$ were used. As can be seen, the transition to the new set-point using the proposed tuning method (dashed lines in Fig.12.5) compares favorably to that obtained when using the IMC-based tuning rules-I and II (dotted

and solid lines, respectively) and the Ziegler-Nichols tuning rules (dash-dotted line). The corresponding manipulated input profiles are shown in Fig.12.5(b).

12.4 Conclusions

In this chapter, we proposed a two-level, optimization-based method for deriving tuning guidelines for PID controllers that take explicitly into account the presence of nonlinear behavior. The central idea was the selection of the PID controller tuning parameters so as to best “emulate” the control action and closed-loop response obtained under a given nonlinear controller, for a broad set of initial conditions and set-point changes. The first level involved using classical tuning guidelines to obtain reasonable bounds on the stabilizing range of tuning parameters. These bounds were in turn incorporated as constraints on the optimization problem solved at the higher level to yield tuning parameter values that improved upon the values obtained from the first level to better emulate the closed-loop behavior under the nonlinear controller. Implications of the method for providing a meaningful link between classical and nonlinear control tools were highlighted, and various implementation issues and possible extensions were discussed. The proposed tuning method was demonstrated through a nonlinear chemical reactor example. For the cases studied, it was found that the proposed method achieved a closed-loop response sufficiently close to the one obtained under the nonlinear controller, and compared favorably with the responses obtained using classical tuning techniques.

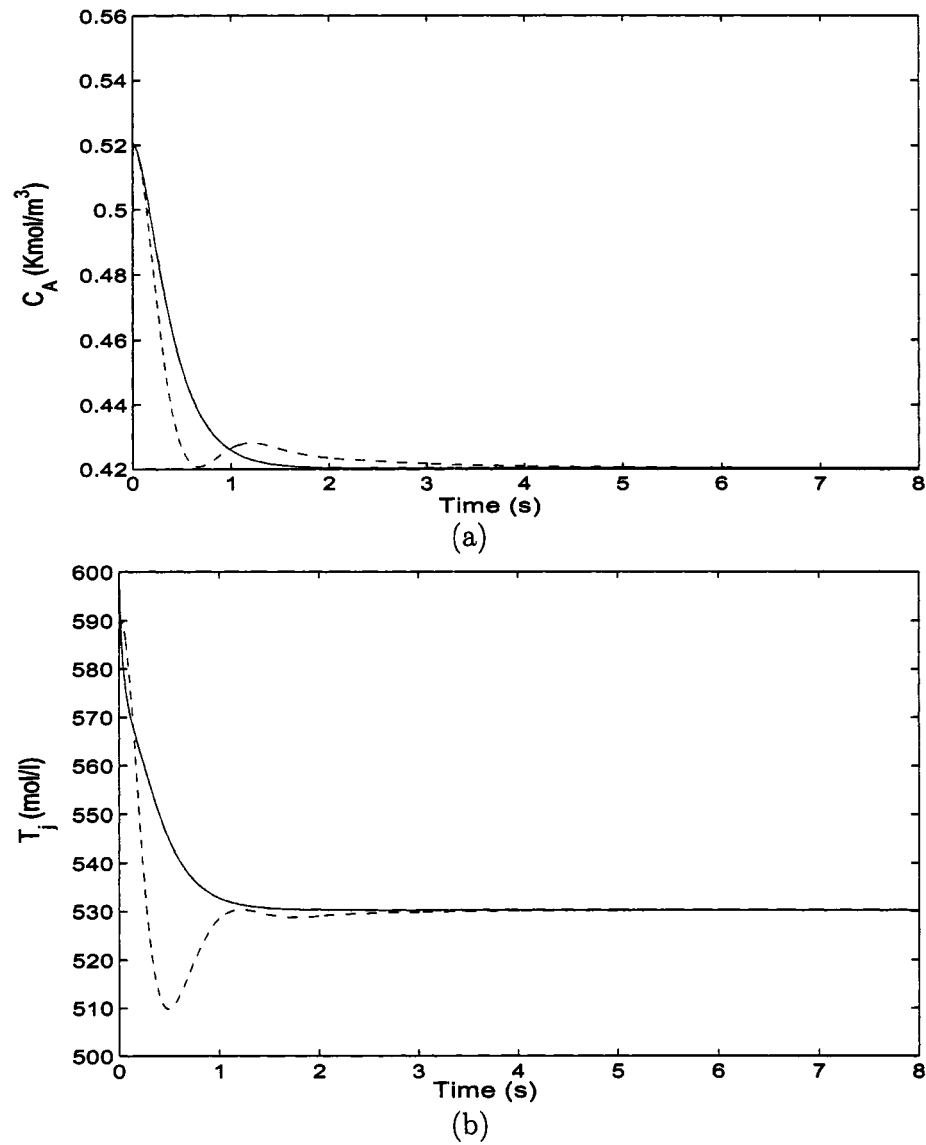


Figure 12.4: Closed-loop output (a) and manipulated input (b) profile under a linearizing controller (solid line) and a PID controller tuned using the proposed method (dashed line).

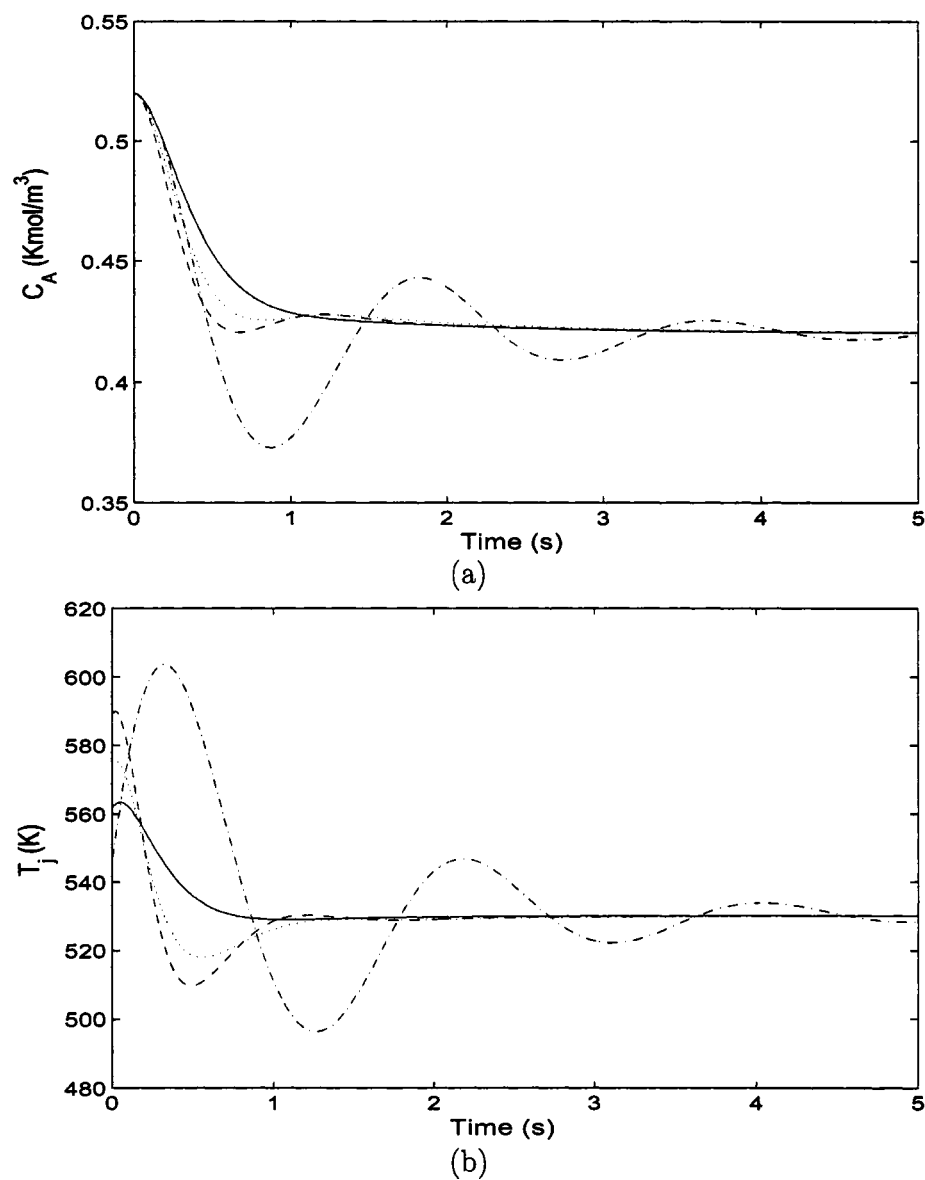


Figure 12.5: Closed-loop output (a) and manipulated input (b) profile using IMC tuning rules I (dotted line), IMC tuning rules II (solid line), Ziegler-Nichols tuning rules (dash-dotted line) and the proposed method (dashed line).

Chapter 13

Conclusions

Modern-day chemical plants involve a complex arrangement of processing units connected, in series and/or in parallel, and highly integrated with respect to material and energy flows through recycle streams. Moreover, chemical processes need to satisfy high quality product specifications and increasingly stringent safety and environmental regulations. The more stringent operating conditions, induced by performance, safety or environmental considerations, have placed new constraints on the operating flexibility of the process and dictates the need for continuous monitoring of the operation of a chemical plant and effective external intervention (automatic control) to guarantee the satisfaction of the process objectives. Central to the design of effective process control and monitoring systems are the issues of strong nonlinear dynamics, plant-model mismatch, external disturbances and actuator constraints. In addition, increased automation tends to increase the vulnerability of the plant to faults (e.g., defects/malfunctions in process equipment, sensors and actuators, failures in the controllers or in the control loops) potentially causing a host of economic, environmental, and safety problems that can seriously degrade the operating efficiency of the plant.

Motivated by the above considerations, this doctoral thesis developed methods and strategies for the design of advanced process control systems that account for the complex process dynamics, including issues such as constraints and uncertainty, and ensure

an efficient and timely response to enhance fault recovery, prevent faults from propagating or developing into total failures, and reduce the risk of safety hazards. To this end, we first developed a hybrid predictive control framework that seamlessly united two different control approaches in a way that reconciled their respective stability and optimality properties. Within this framework, we were able to address several practical issues including constraints, unavailability of measurements and uncertainty for nonlinear systems via explicitly switching between the predictive and Lyapunov-based control approaches. Then, we integrated the two approaches at a design level to develop Lyapunov-based predictive control approaches that allow for an explicit computation of the stability region in the presence of state and control constraints and use the developed Lyapunov-based predictive controllers for the task of ensuring the satisfaction of schedules for chemical processes that switch between various modes of operations. The developed approaches provide an easily implementable framework that seamlessly unite the tasks of performance satisfaction and stability guarantees for both operation within individual modes, and also in the context of operation of chemical processes required to follow a switching schedule. Furthermore, the developed control approaches, and the stability region information that they provided were used in the development of fault-detection and fault-tolerant control strategies that account for the complex dynamics of chemical processes and allow for early detection and reconfiguration to preserve process stability in the event of occurrence of a fault. Finally, the insight gained in the design of predictive control strategies to appropriately express and account for the presence of state and control constraints was used toward developing predictive control strategies for the control of distributed and particulate processes that are operated in continuous and batch mode. For distributed processes described by linear parabolic PDEs, we were able to develop reduced-order predictive control formulations that upon satisfaction of certain conditions (which we derived) guarantee satisfaction of the constraints for the infinite dimensional system. For particulate processes operated in batch mode, we developed predictive control formulations that

allow for appropriately expressing and satisfying constraints on the process operation. Throughout the thesis, numerous examples were used to illustrate the application and demonstrate the advantages of the new control methods.

Some of the further research opportunities that the thesis suggests include the development of Lyapunov-based predictive control algorithms that handle issues such as unavailability of measurement and uncertainty, and their use within scheduled chemical processes where the schedule description involves uncertainty, and the development of nonlinear fault-detection and isolation filters, and the integration of these as well as data-based fault-detection filters within fault-detection and fault-tolerant control approaches, both for single-unit and multi-unit processes. Within this framework, additional issues such as inclusion of robustness and performance-based reconfiguration in fault-tolerant control and accounting for other kinds of faults, including sensor failures and sensor data losses can also be the subject of further research.

Bibliography

- [1] Bemporad A. and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.
- [2] W. M. Haddad A. Leonessa and V. Chellaboina. Nonlinear system stabilization via hierarchical switching control. *IEEE Transactions on Automatic Control*, 46:17–28, 2001.
- [3] F. Allgower and H. Chen. Nonlinear model predictive control schemes with guaranteed stability. In : R. Berber and C. Kravaris (Eds.), *NATO ASI on Nonlinear Model Based Process Control*, pages 465–494, Dordrecht: Kluwer, 1998.
- [4] C. Antoniadis and P. D. Christofides. Feedback control of nonlinear differential difference equation systems. *Chem. Eng. Sci.*, 54:5677–5709, 1999.
- [5] H. B. Aradhye, B. R. Bakshi, J. F. Davis, and S. C. Ahalt. Clustering in wavelet domain: A multiresolution art network for anomaly detection. *AIChE J.*, 50:2455–2466, 2004.
- [6] H. B. Aradhye, J. F. Davis, and B. R. Bakshi. Art-2 and multiscale art-2 for on-line process fault detection - validation via industrial case studies and monte carlo simulation. *Annual Reviews in Control*, 26:113–127, 2002.
- [7] M. Arcak, A. R. Teel, and P. Kokotovic. Robust nonlinear control of feedforward systems with unmodeled dynamics. *Automatica*, 37:265–272, 2001.
- [8] A. Armou and P. D. Christofides. Dynamic optimization of dissipative PDE systems using nonlinear order reduction. *Chem. Eng. Sci.*, 57:5083–5114, 2002.

- [9] Z. Artstein. Stabilization with relaxed control. *Nonlinear Analysis*, 7:1163–1173, 1983.
- [10] K. J. Astrom and T. Hagglund. Automatic tuning of simple regulators with specification on the gain and phase margins. *Automatica*, 20:645–651, 1984.
- [11] K. J. Astrom and T. Hagglund. Revisiting the Ziegler-Nichols step response method for PID control. *J. Proc. Contr.*, 14:635–650, 2004.
- [12] K. J. Astrom, T. Hagglund, C. C. Hang, and W. K. Ho. Automatic tuning and adaption for PID controllers- A survey. *Contr. Eng. Pract.*, 1:699–714, 1993.
- [13] B. Aufderheide, V. Prasad, and B. W. Bequette. A comparison of fundamental model-based and multiple model predictive control. In *Proceedings of 40th IEEE Conference on Decision and Control*, pages 4863–4868, Orlando, FL, 2001.
- [14] J. Baker and P. D. Christofides. Finite dimensional approximation and control of nonlinear parabolic PDE systems. *Int. J. Contr.* to appear, 1999.
- [15] M. J. Balas. Feedback control of linear diffusion processes. *Int. J. Contr.*, 29:523–533, 1979.
- [16] A. Banerjee and Y. Arkun. Model predictive control of plant transitions using a new identification technique for interpolating nonlinear models. *Journal of Process Control*, 8:441–457, 1998.
- [17] J. Bao, W. Z. Zhang, and P. L. Lee. Decentralized fault-tolerant control system design for unstable processes. *Chem. Eng. Sci.*, 58:5045–5054, 2003.
- [18] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35:407–427, 1999.
- [19] A. Bemporad and M. Morari. Robust model predictive control: A survey. In : A. Garulli, A. Tesi and A. Vicino (Eds.), *Robustness in Identification and Control, Lecture Notes in Control and Information Sciences* Vol. 245, pages 207–266, Berlin: Springer, 1999.

- [20] A. R. Benaskeur and A. Desbiens. Backstepping-based adaptive PID control. *IEEE Proc. Contr. Theory Appl.*, 149:54–59, 2002.
- [21] P. Bennema. Spiral growth and surface roughening: Developments since Burton, Cabrera and Frank. *J. Cryst. Growth*, 69:182–197, 1984.
- [22] W. B. Bequette. Nonlinear control of chemical processes: A review. *Ind. & Eng. Chem. Res.*, 30:1391–1413, 1991.
- [23] V. Bhamidi, S. Varanasi, and C. A. Schall. Measurement and modeling of protein crystal nucleation kinetics. *Crystal Growth & Design*, 2:395–400, 2002.
- [24] R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control-The Thinking Man's GPC*. Prentice-Hall, Englewood Cliffs, NJ, 1990.
- [25] M. S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automat. Contr.*, 43:475–482, 1998.
- [26] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Trans. Automat. Control.*, 43:31–45, 1998.
- [27] E. Cacioppo and M. L. Pusey. The solubility of the tetragonal form of hen egg white lysozyme from pH 4.0 to 5.4. *J. Cryst. Growth*, 114:286–292, 1991.
- [28] D. C. Carter, K. Lim, J. X. Ho, B. S. Wright, P. D. Twigg, T. Y. Miller, J. Chapman, K. Keeling, J. Ruble, P. G. Vekilov, B. R. Thomas, F. Rosenberger, and A. A. Chernov. Lower dimer impurity incorporation may result in higher perfection of HEWL crystals grown in microgravity: A case study. *J. Cryst. Growth*, 196:623–637, 1999.
- [29] R. Hwang Chang, W and J. Hsieh. A self tuning PID control for a class of nonlinear systems based on the Lyapunov approach. *J. Proc. Contr.*, 12:233–242, 2002.

- [30] N. E. Chayen and E. Saridakis. Is lysozyme really the ideal model protein? *J. Cryst. Growth*, 232:262–264, 2001.
- [31] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34:1205–1217, 1998.
- [32] Y. Chen, T. Edgar, and V. Manousiouthakis. On infinite-time nonlinear quadratic optimal control. *Syst. Contr. & Lett.*, 51:259–268, 2004.
- [33] L. Chisci, P. Falugi, and G. Zappa. Gain-scheduling MPC of nonlinear systems. *Intr. J. Robust & Nonlin. Contr.*, 13:295–308, 2003.
- [34] T. Chiu and P.D. Christofides. Nonlinear control of particulate processes. *AIChE J.*, 45:1279–1297, 1999.
- [35] T. Chiu and P.D. Christofides. Robust control of particulate processes using uncertain population balances, submitted. *AIChE J.*, 1999.
- [36] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29:121–129, 1996.
- [37] P. D. Christofides. *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*, accepted for publication, 232 pages. Birkhäuser, Boston, 1999.
- [38] P. D. Christofides. *Nonlinear Process Control: Time-scale multiplicity, uncertainty and time-delays*, to be submitted for publication. SIAM, New York, 1999.
- [39] P. D. Christofides. Robust output feedback control of nonlinear singularly perturbed systems. *Automatica*, 36:45–52, 2000.
- [40] P. D. Christofides. *Model-Based Control of Particulate Processes*. Kluwer Academic Publishers, 209 pages, Netherlands, 2002.
- [41] P. D. Christofides and P. Daoutidis. Finite-dimensional control of parabolic PDE systems using approximate inertial manifolds. *J. Math. Anal. Appl.*, 216:398–420, 1997.

- [42] P. Cominos and N. Munro. PID controllers: recent tuning methods and design to specifications. *IEEE Proc. Contr. Theory Appl.*, 149:17–25, 2002.
- [43] R. F. Curtain. Finite-dimensional compensator design for parabolic distributed systems with point sensors and boundary input. *IEEE Trans. Automat. Contr.*, 27:98–104, 1982.
- [44] R. F. Curtain and A. J. Pritchard. *Functional Analysis in Modern Applied Mathematics*, volume 132 of *Mathematics in Science and Engineering*. Academic Press, New York, New York, 1977.
- [45] R. F. Curtain and A. J. Pritchard. *Infinite dimensional linear systems theory*, volume 9 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, Germany, 1978.
- [46] P. A. Darcy and J. M. Wiencek. Identifying nucleation temperatures for lysozyme via differential scanning calorimetry. *J. Cryst. Growth*, 196:243–249, 1999.
- [47] J. F. Davis, M. L. Piovoso, K. Kosanovich, and B. Bakshi. Process data analysis and interpretation. *Advances in Chemical Engineering*, 25:1–103, 1999.
- [48] N. M. C. De oliveira and L. T. Biegler. Constraint handling and stability properties of model-predictive control. *AIChE J.*, 40:1138–1155, 1994.
- [49] R. A. Decarlo, M. S. Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88:1069–1082, 2000.
- [50] R. A. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88:1069–1082, 2000.
- [51] R. C. DeMattei and R. S. Feigelson. Controlling nucleation in protein solutions. *J. of Cryst. Gr.*, 122:21–30, 1992.

- [52] M. A. Demetriou and N. Kazantzis. A new actuator activation policy for performance enhancement of controlled diffusion processes. *Automatica*, 40:415–421, 2004.
- [53] C. DePersis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Trans. Automat. Contr.*, 46:853–865, 2001.
- [54] J. Dimitratos, G. Elicabe, and C. Georgakis. Control of emulsion polymerization reactors. *AIChE J.*, 40:1993–2021, 1994.
- [55] V. Dua, N. A. Bozinis, and E. N. Pistikopoulos. A multiparametric programming approach for mixed-integer quadratic engineering problems. *Comp. & Chem. Engng*, 26:715–733, 2002.
- [56] S. Dubljevic and N. Kazantzis. A new lyapunov design approach for nonlinear systems based on zubov’s method. *Automatica*, 38:1999–2007, 2002.
- [57] P. Dufour, Y. Tour, D. Blanc, and P. Laurent. On nonlinear distributed parameter model predictive control strategy: on-line calculation time reduction and application to an experimental drying process. *Computers & Chemical Engineering*, 39:715–725, 2003.
- [58] R. Dunia and S. J. Qin. Subspace approach to multidimensional fault identification and reconstruction. *AIChE J.*, 44:1813–1831, 1998.
- [59] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE J.*, 42:2797–2812, 1996.
- [60] S. D. Durbin, W. E. Carlson, and M. T. Saros. In-situ studies of protein crystal-growth by atomic-force microscopy. *J. Phys. D: Appl. Phys.*, 26:B128–B132, 1993.
- [61] S. D. Durbin and G. Feher. Crystal growth studies of lysozyme as a model for protein crystallization. *J. Cryst. Growth*, 76:583–592, 1986.

- [62] N. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid predictive control of nonlinear systems: Method and applications to chemical processes. *Int. J. Rob. & Non. Contr.*, 14:199–225, 2004.
- [63] N. H. El-Farra, A. Armou, and P.D. Christofides. Analysis and control of parabolic PDE systems with input constraints. *Automatica*, 39:715–725, 2003.
- [64] N. H. El-Farra, T. Chiu, and P. D. Christofides. Analysis and control of particulate processes with input constraints. *AIChE J.*, 47:1849–1865, 2001.
- [65] N. H. El-Farra and P. D. Christofides. Robust inverse optimal control of nonlinear systems, submitted. *Syst. & Contr. Lett.*; Presented also at 1999 American Control Conference, pp. 124-128, San Diego, CA, 2000.
- [66] N. H. El-Farra and P. D. Christofides. Integrating robustness, optimality and constraints in control of nonlinear processes. *Chem. Eng. Sci.*, 56:1841–1868, 2001.
- [67] N. H. El-Farra and P. D. Christofides. Switching and feedback laws for control of constrained switched nonlinear systems. In *Lecture Notes in Computer Science*, volume 2289, pages 164–178, C. J. Tomlin and M. R. Greenstreet (Eds.), Springer-Verlag, Berlin, 2002.
- [68] N. H. El-Farra and P. D. Christofides. Bounded robust control of constrained multivariable nonlinear processes. *Chem. Eng. Sci.*, to appear, 2003.
- [69] N. H. El-Farra and P. D. Christofides. Bounded robust control of constrained multivariable nonlinear processes. *Chem. Eng. Sci.*, 58:3025–3047, 2003.
- [70] N. H. El-Farra and P. D. Christofides. Coordinating feedback and switching for control of hybrid nonlinear processes. *AIChE J.*, 49:2079–2098, 2003.
- [71] N. H. El-Farra, A. Gani, and P. D. Christofides. Fault-tolerant control of process systems using communication networks. *AIChE J.*, 51:1665–1682, 2005.

- [72] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid control of uncertain process systems. In *Proceedings of 8th International Symposium on Process Systems and Engineering*, Kuming, P.R. China, 2003.
- [73] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid predictive control of nonlinear systems: Method and applications to chemical processes. *International Journal of Robust & Nonlinear Control*, to appear, 2004.
- [74] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid predictive control of nonlinear systems: Method and applications to chemical processes. *Int. J. Rob. & Non. Contr.*, 4:199–225, 2004.
- [75] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Uniting bounded control and MPC for stabilization of constrained linear systems. *Automatica*, 40:101–110, 2004.
- [76] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss. State and output feedback nonlinear model predictive control: An overview. *Eur. J. Contr.*, 9:190–206, 2003.
- [77] E. L. Forsythe, A. Nadarajah, and M. L. Pusey. Growth of (101) faces of tetragonal lysozyme crystals: measured growth-rate trends. *Acta Crystallographica Section D - Biological Crystallography*, 55:1005–1011, 1999.
- [78] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – a survey and some new results. *Automatica*, 26:459–474, 1990.
- [79] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *J. Proc. Contr.*, 7:403–424, 1997.
- [80] R. A. Freeman and P. V. Kokotovic. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Birkhauser, Boston, 1996.
- [81] S. K. Friedlander. *Smoke, Dust, and Haze: Fundamentals of Aerosol Behavior*. Wiley, New York, 1977.

- [82] O. Galkin and P. G. Vekilov. Direct determination of the nucleation rates of protein crystals. *Journal of Physical Chemistry B*, 103:10965 – 10971, 1999.
- [83] C. E. Garcia and A. M. Morshedi. Quadratic-programming solution of dynamic matrix control (QDMC). *Chem. Eng. Commun.*, 46:73–87, 1986.
- [84] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25:335–348, 1989.
- [85] V. Garcia-Onorio and B. E. Ydstie. Distributed, asynchronous and hybrid simulation of process networks using recording controllers. *Int. J. Rob. & Non. Contr.*, 14:227–248, 2004.
- [86] F. Gelbard and J. H. Seinfeld. Numerical solution of the dynamic equation for particulate processes. *J. Comp. Phys.*, 28:357–375, 1978.
- [87] H. Genceli and M. Nikolaou. Robust stability analysis of constrained $l(1)$ -norm model predictive control. *AIChE J.*, 39:1954–1965, 1993.
- [88] G. Grimm, M. Messina, A. Teel, and S. Tuna. Nominally robust model predictive control with state constraints. In *Proceedings of 42nd IEEE Conference on Decision and Control*, to appear, Maui, HI, 2003.
- [89] J. W. Grizzle and P. V. Kokotovic. Feedback linearization of sampled-data systems. *IEEE. Trans. Automat. Contr.*, 33:857–859, 1988.
- [90] I. E. Grossmann, S. A. van den Heever, and I. Harjunoski. Discrete optimization methods and their role in the integration of planning and scheduling. In *Proceedings of 6th International Conference on Chemical Process Control*, pages 124–152, Tucson, AZ, 2001.
- [91] M. A. Henson and D. E. Seborg. *Nonlinear Process Control*. Prentice-Hall, Englewood Cliffs, NJ, 1997.
- [92] J. Hespanha, D. Liberzon, A. S. Morse, B. D. O. Anderson, T. S. Brinsmead, and F. De Bruyne. Multiple model adaptive control. part 2: Switching. *International Journal Of Robust & Nonlinear Control*, 11:479–496, 2001.

- [93] J. Hespanha and A. S. Morse. Switching between stabilizing controllers. *Automatica*, 38:1905–1917, 2004.
- [94] J. P. Hespanha and A. S. Morse. Stability of switched systems with average dwell time. In *Proceedings of 38th IEEE Conference on Decision and Control*, pages 2655–2660, Phoenix, AZ, 1999.
- [95] M. J. Hounslow. A discretized population balance for continuous systems at steady-state. *AIChE J.*, 36:106–116, 1990.
- [96] B. Hu, X. Xu, P. J. Antsaklis, and A. N. Michel. Robust stabilizing control law for a class of second-order switched systems. *Syst. Contr. Lett.*, 38:197–207, 1999.
- [97] T. Hu and Z. Lin. Composite quadratic Lyapunov functions for constrained control systems. In *Proceedings of 41st IEEE Conference on Decision & Control*, pages 3500–3505, Las Vegas, NV, 2002.
- [98] T. Hu, Z. Lin, and Li Qiu. An explicit description of null controllable regions of linear systems with saturating actuators. *Syst. & Contr. Lett.*, 47:65–78, 2002.
- [99] H. M. Hulburt and S. Katz. Some problems in particle technology: A statistical mechanical formulation. *Chem. Eng. Sci.*, 19:555–574, 1964.
- [100] C. D. Immanuel and F. J. Doyle III. Hierarchical multiobjective strategy for particle-size distribution control. *AIChE J.*, 49:2383–2399, 2003.
- [101] A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin-Heidelberg, second edition, 1989.
- [102] K. Ito and K. Kunisch. Receding horizon optimal control for infinite dimensional systems. *ESIAM: Control, Optimization and Calculus of Variations*, 8:741–760, 2002.
- [103] G. R. Jerauld, Y. Vasatis, and M. F. Doherty. Simple conditions for the appearance of sustained oscillations in continuous crystallizers. *Chem. Eng. Sci.*, 38:1675–1681, 1983.

- [104] A. G. Jones, A. Chianese, and J. W. Mullin. Effect of fines destruction on batch colling crystallization of potassium sulphate solutions. In S. J. Jancic and E. J. de Jong, editors, *Industrial crystallization 84 : Proceedings of the 9th Symposium on Industrial Crystallization*, pages 191–195, 1984.
- [105] R. A. Judge, R. S. Jacobs, T. Frazier, E. H. Snell, and M. L. Pusey. The effect of temperature and solution pH on the nucleation of tetragonal lysozyme crystals. *Biophysical Journal*, 77:1585–1593, 1999.
- [106] N. Kapoor and P. Daoutidis. Stabilization of systems with input constraints. *Int. J. Contr.*, 34:653–675, 1998.
- [107] N. Kazantzis. A functional equations approach to nonlinear discrete-time feedback stabilization through pole-placement. *Syst. & Contr. Lett.*, 43:361–369, 2001.
- [108] N. Kazantzis and C. Kravaris. Nonlinear observer design using Lyapunov’s auxiliary theorem. *Syst. & Contr. Lett.*, 34:241–247, 1999.
- [109] N. Kazantzis, C. Kravaris, C. Tseronis, and R. A. Wright. Optimal controller tuning for nonlinear processes. *Automatica*, 41:79–86, 2005.
- [110] N. Kazantzis, C. Kravaris, and R. A. Wright. Nonlinear observer design for process monitoring. *Ind. & Eng. Chem. Res.*, 39:408–419, 2000.
- [111] S. Keerthi and E. Gilbert. Optimal, infinite horizon feedback laws for a general class of constrained discrete time systems. *Journal of Optimization Theory and Applications*, 57:265–293, 1988.
- [112] H. K. Khalil. *Nonlinear Systems*. Macmillan Publishing Company, New York, second edition, 1996.
- [113] A. M. Kierzek and P. Zielenkiewicz. Models of protein crystal growth. *Biophysical Chemistry*, 91:1–20, 2001.

- [114] P. Kokotovic and M. Arcak. Constructive nonlinear control: a historical perspective. *Automatica*, 37:637–662, 2001.
- [115] S. L. D. Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Automat. Contr.*, 45:1053–1071, 2000.
- [116] J. V. Kresta, J. F. Macgregor, and T. E. Marlin. Multivariate statistical monitoring of process operating performance. *Can. J. Chem. Eng.*, 69:35–47, 1991.
- [117] N. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design*. Wiley, New York, first edition, 1995.
- [118] S. Kumar and D. Ramkrishna. On the solution of population balance equations by discretization-II. a moving pivot technique. *Chem. Eng. Sci.*, 51:1333–1342, 1996.
- [119] K. Kurihara, S. Miyashita, G. Sazaki, T. Nakada, Y. Suzuki, and H. Komatsu. Interferometric study on the crystal growth of tetragonal lysozyme crystal. *J. Cryst. Growth*, 166:904–908, 1996.
- [120] J. T. Lee, W. Cho, and T. F. Edgar. Control system design based on a first order plus time delay model. *J. Proc. Contr.*, 7:65–73, 1997.
- [121] S. J. Lei, R. Shinnar, and S. Katz. The stability and dynamic behavior of a continuous crystallizer with a fines trap. *AIChE J.*, 17:1459–1470, 1971.
- [122] A. K. W. Leung, M. M. V. Park, and D. W. Borhani. An improved method for protein crystal density measurements. *J. Appl. Cryst.*, 32:1006–1009, 1999.
- [123] M. Li, A. Nadarajah, and M. L. Pusey. Modeling the growth rates of tetragonal lysozyme crystals. *J. Cryst. Growth*, 156:121–132, 1995.
- [124] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19:59–70, 1999.

- [125] D. Liberzon, E. D. Sontag, and Y. Wang. Universal construction of feedback laws achieving ISS and integral-ISS disturbance attenuation. *Syst. & Contr. Lett.*, 46:111–127, 2002.
- [126] D. Liberzon, E.D. Sontag, and Y Wang. Universal construction of feedback laws achieving ISS and integral-ISS disturbance attenuation. *Systems & Control Letters*, 46:111–127, 2002.
- [127] Y. Lin and E. D. Sontag. A universal formula for stabilization with bounded controls. *Systems & Control Letters*, 16:393–397, 1991.
- [128] J. Lygeros, D. N. Godbole, and S. S. Sastry. A game theoretic approach to hybrid system design. In *Lecture Notes in Computer Science*, volume 1066, pages 1–12, Alur, R. and T. Henzinger Eds., Berlin: Springer, 1996.
- [129] M. A. Dahleh M. W. McConley, B. D. Appleby and E. Freon. A computationally efficient lyapunov-based scheduling procedure for control of nonlinear systems with stability guarantees. *IEEE Transactions on Automatic Control*, 45:33–49, 2000.
- [130] D. L. Ma, D. K. Tafti, and R. D. Braatz. Optimal control and simulation of multidimensional crystallization processes. *Comp. & Chem. Eng.*, 26:1103–1116, 2002.
- [131] L. Magni, G. Nicolao, R. Scattolini, and F. Allgower. Robust model predictive control for nonlinear discrete-time systems. *Int. J. Rob. & Non. Contr.*, 13:229–246, 2003.
- [132] L. Magni, G. De Nicolao, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37:1351–1362, 2001.
- [133] N. A. Mahmoud and H. K. Khalil. Asymptotic regulation of minimum phase nonlinear systems using output feedback. *IEEE Trans. Automat. Contr.*, 41:1402–1412, 1996.

- [134] M. Malisoff and E.D. Sontag. Universal formulas for feedback stabilization with respect to Minkowski balls. *Systems & Control Letters*, 40:247–260, 2000.
- [135] V. Manousiouthakis and D. Chmielewski. On constrained infinite-time nonlinear optimal control. *Chem. Eng. Sci.*, 57:105–114, 2002.
- [136] N. V. Mantzaris and P. Daoutidis. Cell population balance modeling and control in continuous bioreactors. *Journal of Process Control*, 14:775–784, 2004.
- [137] N. V. Mantzaris, P. Daoutidis, and F. Srienc. Numerical solution of multi-variable cell population balance models, i. finite difference methods. *Computer & Chemical Engineering*, 25:1411–1440, 2001.
- [138] T. E. Marlin. *Process Control*. McGraw Hill, 2000.
- [139] M. Massoumnia, G. C. Verghese, and A. S. Willsky. Failure detection and identification. *IEEE Trans. Automat. Contr.*, 34:316–321, 1989.
- [140] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Trans. Automat. Contr.*, 35:814–824, 1990.
- [141] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.
- [142] D.Q. Mayne. Nonlinear model predictive control: An assessment. In : J. C. Kantor, C. E. Garcia and B. Carnahan (Eds.) *Proceedings of 5th International Conference on Chemical Process Control*, pages 217–231, CACHE, AIChE, 1997.
- [143] A. McPherson. A brief history of protein crystal growth. *J. Cryst. Growth*, 110:1–10, 1991.
- [144] N. Mehranbod, M. Soroush, and C. Panjapornpon. A method of sensor fault detection and identification. *J. Proc. Contr.*, 15:321–339, 2005.
- [145] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Hybrid predictive control of process systems. *AIChE J.*, 50:1242–1259, 2004.

- [146] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Stabilization of switched nonlinear systems using predictive control. In *Proceedings of American Control Conference*, pages 560–565, Boston, MA, 2004.
- [147] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Robust hybrid predictive control of nonlinear systems. *Automatica*, 41:209–217, 2005.
- [148] H. Michalska and D. Q. Mayne. Receding horizon control of non-linear systems. *IEEE Trans. Automat. Contr.*, 35:814–824, 1990.
- [149] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Automat. Contr.*, 38:1623–1633, 1993.
- [150] H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Trans. Automat. Contr.*, 40:995–1006, 1995.
- [151] S. M. Miller and J. B. Rawlings. Model identification and control strategies for batch cooling crystallizers. *AIChE J.*, 40:1312–1327, 1994.
- [152] L. A. Monaco and F. Rosenberger. Growth and etching kinetics of tetragonal lysozyme. *J. Cryst. Growth*, 129:465–484, 1993.
- [153] M. Morari and J. H. Lee. Model predictive control: Past, present and future. *Comp. & Chem. Eng.*, 23:667–682, 1999.
- [154] M. Morari and E. Zafriou. *Robust Process Control*. Prentice Hall, New Jersey, 1989.
- [155] J. W. Mullin and J. Nyvlt. Programmed cooling of batch crystallizers. *Chem. Eng. Sci.*, 26:369–377, 1971.
- [156] Y. Nagatoshi, G. Sazaki, Y. Suzuki, S. Miyashita, T. Matsui, T. Ujihara, K. Fujiwara, N. Usami, and K. Nakajima. Effects of high pressure on the growth kinetics of orthorhombic lysozyme crystals. *J. Cryst. Growth*, 254:188–195, 2003.
- [157] A. Negiz and A. Cinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE J.*, 43:2002–2020, 1997.

- [158] D. Nesic, A. R. Teel, and P. V. Kokotovic. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Syst. & Contr. Lett.*, 38:259–270, 1999.
- [159] I. Nimmo. Adequately address abnormal operations. *Chem. Eng. Prog.*, 91:36–45, 1995.
- [160] P. Nomikos and J. F. Macgregor. Monitoring batch processes using multiway principal component analysis. *AIChE J.*, 40:1361–1375, 1994.
- [161] N. M. C. Oliveira and L. T. Biegler. Constraint handling and stability properties of model-predictive control. *AIChE J.*, 40:1138–1150, 1994.
- [162] H. Panagopoulos, K. J. Astrom, and T. Hagglund. Design of PID controllers based on constrained optimization. *IEEE Proc. Contr. Theory Appl.*, 149:32–40, 2002.
- [163] R. J. Patton. Fault-tolerant control systems: The 1997 situation. In *Proceedings of the IFAC Symposium SAFEPROCESS 1997*, pages 1033–1054, Hull, United Kingdom, 1997.
- [164] P. Peleties and R. DeCarlo. Asymptotic stability of m -switched systems using lyapunov-like functions. In *Proceedings of American Control Conference*, pages 1679–1684, Boston, MA, 1991.
- [165] E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, and M. Morari. On-line optimization via off-line parametric optimization tools. *Computers & Chemical Engineering*, 26:175–185, 2002.
- [166] P. Pladis and C. Kiparissides. A comprehensive model for the calculation of molecular weight-long-chain branching distribution in free-radical polymerizations. *Chem. Eng. Sci.*, 53:3315–3333, 1998.
- [167] E. Polak and T. H. Yang. Moving horizon control of linear systems with input saturation and plant uncertainty - part 1: Robustness. *Intr. J. Contr.*, 58:613–638, 1993.

- [168] E. Poulin and A. Pomerleau. PID tuning for integrating and unstable processes. *IEEE Proc. Contr. Theory App.*, 143:429–435, 1996.
- [169] S. E. Pratsinis. Simultaneous nucleation, condensation, and coagulation in aerosol reactors. *J. Coll. Inter. Sci.*, 124:416–426, 1988.
- [170] J. A. Primbs, V. Nevistic, and J. C. Doyle. A receding horizon generalization of pointwise min-norm controllers. *IEEE Trans. Automat. Contr.*, 45:898–909, 2000.
- [171] M. L. Pusey and A. Nadarajah. A model for tetragonal lysozyme crystal nucleation and growth. *Crystal Growth & Design*, 2:475–483, 2002.
- [172] M. L. Pusey and R. Naumann. Growth kinetics of tetragonal lysozyme crystals. *J. Cryst. Growth*, 76:593–599, 1986.
- [173] M. L. Pusey, W. Witherow, and R. Naumann. Preliminary investigations into solutal flow about growing tetragonal lysozyme crystals. *J. Cryst. Growth*, 90:105–111, 1988.
- [174] S. J. Qin and T. J. Badgwell. An overview of industrial model predictive control technology. In : J. Kantor, C. Garcia and B. Carnahan (Eds.), *5th International Conference on Chemical Process Control*, pages 232–256, CACHE, AIChE, 1997.
- [175] D. Ramkrishna. The status of population balances. *Rev. Chem. Eng.*, 3:49–95, 1985.
- [176] D. Ramkrishna. *Population Balances – Theory and Applications to Particulate Systems in Engineering*. Academic Press, San Diego, 2000.
- [177] A. D. Randolph and M. A. Larson. *Theory of Particulate Processes*. Academic press, Second edition, San Diego, 1988.
- [178] C. V. Rao and J. B. Rawlings. Constrained process monitoring: Moving-horizon approach. *AIChE J.*, 48:97–109, 2002.

- [179] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20:38–52, 2000.
- [180] J. B. Rawlings, S. M. Miller, and W. R. Witkowski. Model identification and control of solution crystallization process – a review. *Ind. & Eng. Chem. Res.*, 32:1275–1296, 1993.
- [181] J. B. Rawlings, S. M. Miller, and W. R. Witkowski. Model identification and control of solution crystallization processes. *I & EC Res.*, 32:1275–1296, 1993.
- [182] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Trans. Automat. Contr.*, 38:1512–1516, 1993.
- [183] J. B. Rawlings and W. H. Ray. Stability of continuous emulsion polymerization reactors: a detailed model analysis. *Chem. Eng. Sci.*, 42:2767–2777, 1987.
- [184] J. B. Rawlings, C. W. Slink, and S. M. Miller. Control of crystallization processes. In Allan S. Myerson, editor, *Handbook of Industrial Crystallization. Second Edition*, pages 201–230. Butterworth-Heinemann, 2001.
- [185] W. H. Ray. *Advanced Process Control*. McGraw-Hill, New York, 1981.
- [186] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.
- [187] F. Riesz and B. Sz.-Nagy. *Functional analysis*. Dover Publications, New York, U.S.A., 1990.
- [188] D. E. Rivera, M. Morari, and S. Skogestad. Internal model control. 4. PID controller design. *Ind. Eng. Chem. Proc. Des. & Dev.*, 25:252–265, 1986.
- [189] S. Rohani and J. R. Bourne. Self-tuning control of crystal size distribution in a cooling batch crystallizer. *Chem. Eng. Sci.*, 12:3457–3466, 1990.
- [190] S. Rohani, N. S. Tavare, and J. Garside. Control of crystal size distribution in a batch cooling crystallizer. *Can. J. Chem. Eng.*, 68:260–267, 1990.

- [191] D. R. Rollins and J. F. Davis. An unbiased estimation technique when gross errors exist in process measurements. *AIChE J.*, 38:563–572, 1992.
- [192] D. R. Rollins and J. F. Davis. Unbiased estimation of gross errors when the covariance matrix is unknown. *AIChE J.*, 39:1335–1341, 1993.
- [193] F. Rosenberger. Protein crystallization. *J. Cryst. Growth*, 166:40–54, 1996.
- [194] F. Rosenberger, S. B. Howard, J. W. Sowers, and T. A. Nyce. Temperature dependence of protein solubility determination and application to crystallization in x-ray capillaries. *J. Cryst. Growth*, 129:1–12, 1993.
- [195] F. Rosenberger and E. J. Meehan. Control of nucleation and growth in protein crystal growth. *J. Cryst. Gr.*, 90:74–78, 1988.
- [196] F. Rosenberger, P. G. Vekilov, M. Muschol, and B. R. Thomas. Nucleation and crystallization of globular proteins – what we know and what is missing. *J. Cryst. Growth*, 168:1–27, 1996.
- [197] W. J. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36:1401–1425, 2000.
- [198] A. Saberi, A. A. Stoorvogel, P. Sannuti, and H. Niemann. Fundamental problems in fault detection and identification. *Int. J. Rob. & Non. Contr.*, 10:1209–1236, 2000.
- [199] G. Sazaki, K. Kurihara, T. Nakada, S. Miyashita, and H. Komatsu. A novel approach to the solubility measurement of protein crystals by two-beam interferometry. *J. Cryst. Growth*, 169:355–360, 1996.
- [200] C. A. Schall, J. S. Riley, E. Li, E. Arnold, and J. M. Wiencek. Application of temperature control strategies to the growth of hen egg-white lysozyme crystals. *J. Cryst. Gr.*, 165:299–307, 1996.
- [201] K. D. Schott and B. W. Bequette. MMAC: Using a bank of linear models to control a nonlinear system. In R. Berber and C. Kravaris (Eds.), *NATO ASI on*

Nonlinear Model Based Process Control, pages 33–57, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

- [202] P. O. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Auto. Contr.*, 44:648–654, 1999.
- [203] P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Trans. Automat. Contr.*, 43:1163–1169, 1998.
- [204] P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE J.*, 45:1649–1259, 1999.
- [205] D. Semino and W. H. Ray. Control of systems described by population balance equations-I. controllability analysis. *Chem. Eng. Sci.*, 50:1805–1824, 1995.
- [206] D. Semino and W. H. Ray. Control of systems described by population balance equations-II. emulsion polymerization with constrained control action. *Chem. Eng. Sci.*, 50:1825–1839, 1995.
- [207] R. Sepulchre, M. Jankovic, and P. Kokotovic. *Constructive Nonlinear Control*. Springer-Verlag, Berlin-Heidelberg, 1997.
- [208] J. S. Shamma and K. Tu. Output feedback control for systems with constraints and saturations: Scalar control case. *Syst. & Contr. Lett.*, 25:1–11, 1998.
- [209] D. Shi, N. H. El-Farra, M. Li, P. Mhaskar, and P. D. Christofides. Predictive control of particle size distribution in particulate processes. *Chem. Eng. Sci.*, accepted, 2005.
- [210] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *J. Proc. Contr.*, 13:291–309, 2003.
- [211] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. Wiley, 1996.
- [212] A. Smyshlyaev and M. Krstic. Explicit formulae for boundary control of parabolic PDEs. *Lecture Notes in Control & Information Sciences*, 301:231–249, 2004.

- [213] M. Soroush and N. Zambare. Nonlinear output feedback control of a class of polymerization reactors. *IEEE Trans. Contr. Syst. Tech.*, 8:310–320, 2000.
- [214] J.A. Stiver, X.D. Koutsoukos, and P.J. Antsaklis. An invariant-based approach to the design of hybrid control systems. *Intr. J. Robust & Nonlin. Contr.*, 11:453–478, 2001.
- [215] D. Sun and K. A. Hoo. Dynamic transition control structure for a class of SISO nonlinear systems. *IEEE Trans. on Contr. Syst. Tech.*, 7:622–629, 1999.
- [216] H. J. Sussmann and P. V. Kokotovic. The peaking phenomenon and the global stabilization of nonlinear systems. *IEEE Trans. Automat. Contr.*, 36:424–440, 1991.
- [217] M. Sznaier, J. Cloutier, R. Hull, D. Jacques, and C. Mracek. Receding horizon control Lyapunov function approach to suboptimal regulation of nonlinear systems. *Journal of Guidance Control & Dynamics*, 23:399–405, 2000.
- [218] M. Sznaier and M. J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proceedings of 26th IEEE Conference on Decision and Control*, pages 761–762, Los Angeles, CA, 1987.
- [219] M. Sznaier, R. Suarez, and J. Cloutier. Suboptimal control of constrained nonlinear systems via receding horizon constrained control Lyapunov functions. *Intr. J. Robust & Nonlin. Contr.*, 13:247–259, 2003.
- [220] E. Tatara and A. Cinar. An intelligent system for multivariate statistical process monitoring and diagnosis. *ISA Transactions*, 41:255–270, 2002.
- [221] A. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Syst. & Contr. Lett.*, 18:165–171, 1992.
- [222] A. Teel and L. Praly. Global stabilizability and observability imply semi-global stabilizability by output feedback. *Syst. & Contr. Lett.*, 22:313–325, 1994.

- [223] A. R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Syst. & Contr. Lett.*, 18:165–171, 1992.
- [224] B. D. Tyreus and W. L. Luyben. Tuning PI controllers for integrator dead time processes. *Ind. & Eng. Chem. Res.*, 31:2625–2628, 1992.
- [225] P. G. Vekilov and A. A. Chernov. The physics of protein crystallization. *Solid State Physics - Advances in Research and Applications*, 57:1–147, 2002.
- [226] P. G. Vekilov and F. Rosenberger. Dependence of lysozyme growth kinetics on step sources and impurities. *J. Cryst. Growth*, 158:540–551, 1996.
- [227] Z. Wan and M. V. Kothare. Efficient scheduled stabilizing model predictive control for constrained nonlinear systems. *Intr. J. Robust & Nonlin. Contr.*, 13:331–346, 2003.
- [228] K. B. Ward, W. M. Zuk, M. A. Perozzo, M. A. Walker, G. I. Birnbaum, W. Kung, A. Cavaliere, D. R. Uffen, and H. Scholaert. Dynamic telerobotic control of crystallization experiments. *J. Cryst. Growth*, 122:235–241, 1992.
- [229] J. S. Wey and P. H. Karpinski. Batch crystallization. In Allan S. Myerson, editor, *Handbook of Industrial Crystallization. Second Edition*, pages 231–248. Butterworth-Heinemann, 2001.
- [230] M. A. Wicks, P. Peleties, and R. A. DeCarlo. Switched controller synthesis for the quadratic stabilization of a pair of unstable linear systems. *European J. Contr.*, 4:140–147, 1998.
- [231] J. M. Wiencek. New strategies for protein crystal growth. *Annual Review of Biomedical Engineering*, 1:505–534, 1999.
- [232] R. A. Wright, C. Kravaris, and N. Kazantzis. Model based synthesis of nonlinear PI and PID controllers. *AIChE J.*, 47:1805–1818, 2001.
- [233] N. E. Wu. Coverage in fault-tolerant control. *Automatica*, 40:537–548, 2004.

- [234] W. Xie, S. Rohani, and A. Phoenix. Dynamic modeling and operation of a seeded batch cooling crystallizer. *Chem. Eng. Comm.*, 187:229–249, 2001.
- [235] X. P. Xu and P. J. Antsaklis. Results and perspectives on computational methods for optimal control of switched systems. *Lect. Notes Comp. Sci.*, 2623:540–555, 2003.
- [236] E. C. Yamalidou and J. Kantor. Modeling and optimal control of discrete-event chemical processes using petri nets. *Computers and Chemical Engineering*, 15:503–519, 1990.
- [237] G. H. Yang, J. L. Wang, and Y. C. Soh. Reliable H_∞ control design for linear systems. *Automatica*, 37:717–725, 2001.
- [238] E. B. Ydstie. Certainty equivalence adaptive control: Paradigms puzzles and switching. In *Proceedings of 5th International Conference on Chemical Process Control*, pages 9–23, Tahoe City, CA, 1997.
- [239] E. B. Ydstie. New vistas for process control: Integrating physics and communication networks. *AIChE J.*, 48:422–426, 2002.
- [240] L. Zaccarian, A. R. Teel, and D. Nesic. On finite gain L_P stability of nonlinear sampled-data systems. *Syst. & Contr. Lett.*, 49:201–212, 2003.
- [241] E. Zafiriou and A. L. Marchal. Stability of SISO quadratic dynamic matrix control with hard output constraints. *AIChE J.*, 37:1550–1560, 1991.
- [242] G. P. Zhang and S. Rohani. On-line optimal control of a seeded batch cooling crystallizer. *Chem. Eng. Sci.*, 58:1887–1896, 2003.
- [243] X. D. Zhang, T. Parisini, and M. M. Polycarpou. Adaptive fault-tolerant control of nonlinear uncertain systems: An information-based diagnostic approach. *IEEE Trans. Automat. Contr.*, 49:1259–1274, 2004.
- [244] A. Zheng and M. Morari. Stability of model predictive control with mixed constraints. *IEEE Trans. Automat. Contr.*, 40:1818–1823, 1995.

- [245] D. H. Zhou and P. M. Frank. Fault diagnostics and fault tolerant control. *IEEE Transactions on Aerospace and Electronic Systems*, 34:420–427, 1998.
- [246] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 64:759–768, 1942.