

UNIVERSITY OF CALIFORNIA

Los Angeles

Fault-Tolerant Process Control:  
Handling Actuator and Sensor Malfunctions

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

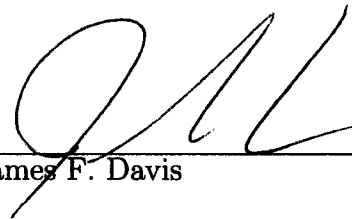
by

Adiwinata Gani

2007

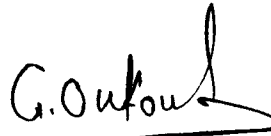
© Copyright by  
Adiwinata Gani  
2007

The dissertation of Adiwinata Gani is approved.



---

James F. Davis



---

Gerassimos Orkoulas



---

Robert T. M'Closkey



---

Panagiotis D. Christofides, Committee Chair

University of California, Los Angeles

2007

Buat Papa dan Mama

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background and Motivation for Fault-Tolerant Control . . . . .	1
1.2	Background and Motivation for Networked Control Systems . . . . .	3
1.3	Dissertation Objectives and Structure . . . . .	6
<b>2</b>	<b>Fault-Tolerant Control of Constrained Nonlinear Processes Using Communication Networks</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Preliminaries . . . . .	10
2.2.1	System Description . . . . .	10
2.2.2	Problem Statement and Solution Overview . . . . .	12
2.2.3	Motivating Example . . . . .	13
2.3	Fault-Tolerant Control System Design Methodology . . . . .	17
2.3.1	Constrained Feedback Controller Synthesis . . . . .	18
2.3.2	Characterization of Fault-Recovery Regions . . . . .	25
2.3.3	Supervisory Switching Logic Design . . . . .	27
2.3.4	Design of Communication Logic . . . . .	29

2.4	Simulation Studies . . . . .	31
2.4.1	Application to a Single Chemical Reactor . . . . .	33
2.4.2	Application to Two Chemical Reactors in Series . . . . .	42
2.5	Conclusions . . . . .	54
<b>3</b>	<b>Fault-Tolerant Control of Nonlinear Processes: Performance-Based Reconfiguration and Robustness</b>	<b>56</b>
3.1	Introduction . . . . .	56
3.2	Preliminaries . . . . .	58
3.2.1	Motivating Example . . . . .	59
3.2.2	Bounded Lyapunov-Based Control . . . . .	60
3.2.3	Model Predictive Control . . . . .	62
3.3	Fault-Tolerant Control: Performance-Based Reconfiguration . . . . .	63
3.3.1	Lyapunov-Based Predictive Control . . . . .	64
3.3.2	Performance-Based Reconfiguration . . . . .	67
3.3.3	Application to Chemical Process Example . . . . .	70
3.4	Fault-Tolerant Control: Robustness Considerations . . . . .	75
3.4.1	Robust Hybrid Predictive Controller . . . . .	78
3.4.2	Robust Fault-Tolerant Control . . . . .	80
3.4.3	Application to Chemical Process Example with Uncertainty and Disturbance . . . . .	81
3.5	Conclusions . . . . .	84
<b>4</b>	<b>Integrated Fault-Detection and Fault-Tolerant Control of Process Systems</b>	<b>87</b>

4.1	Introduction . . . . .	87
4.2	Preliminaries . . . . .	90
4.2.1	Process Description . . . . .	90
4.2.2	Motivating Example . . . . .	91
4.2.3	Bounded Lyapunov-Based Control . . . . .	92
4.3	Integrated Fault-Detection and Fault-Tolerant Control: State Feed- back Case . . . . .	94
4.3.1	State Feedback Fault-Tolerant Control . . . . .	94
4.3.2	Simulation Results . . . . .	100
4.4	Integrated Fault-Detection and Fault-Tolerant Control: Output Feed- back Case . . . . .	104
4.4.1	Output Feedback Control . . . . .	107
4.4.2	Integrating Fault-Detection and Fault-Tolerant Output Feed- back Control . . . . .	111
4.4.3	Simulation Results . . . . .	119
4.5	Conclusions . . . . .	136
<b>5</b>	<b>Fault-Tolerant Control of a Polyethylene Reactor</b>	<b>138</b>
5.1	Introduction . . . . .	138
5.2	Process Description and Modeling . . . . .	140
5.3	Fault-Tolerant Control . . . . .	144
5.3.1	Constrained Feedback Controller Synthesis . . . . .	146
5.3.2	Characterization of Constrained Stability Regions . . . . .	148
5.3.3	Fault-Detection Filter Design . . . . .	149

5.3.4	Fault-Tolerant Switching Logic . . . . .	151
5.4	Simulation Results . . . . .	152
5.5	Conclusions . . . . .	163
<b>6</b>	<b>Fault-Tolerant Control of Nonlinear Process Systems Subject to Sensor Faults</b>	<b>165</b>
6.1	Introduction . . . . .	165
6.2	Preliminaries . . . . .	168
6.2.1	A Chemical Reactor Example . . . . .	171
6.3	Stabilization Subject to Sensor Failures . . . . .	172
6.3.1	Reconfiguration Law . . . . .	172
6.3.2	Application to Chemical Reactor . . . . .	176
6.4	Stabilization Subject to Sensor Data Losses . . . . .	178
6.4.1	Modeling Sensor Data Loss . . . . .	178
6.4.2	Analyzing Closed-Loop Stability . . . . .	180
6.4.3	Control of a Chemical Reactor Subject to Sensor Data Loss . . . . .	188
6.5	Fault-Tolerant Control Subject to Sensor Data Losses . . . . .	190
6.5.1	Reconfiguration law . . . . .	190
6.5.2	Fault-Tolerant Control of a Chemical Reactor . . . . .	193
6.5.3	Fault-Tolerant Control of a Polyethylene Reactor Subject to Sensor Data Loss . . . . .	194
6.6	Conclusions . . . . .	202
<b>7</b>	<b>Handling Sensor Malfunctions in Control of Particulate Processes</b>	<b>205</b>
7.1	Introduction . . . . .	205



7.2	Handling Sensor Malfunctions: Continuous Crystallizer . . . . .	208
7.2.1	Population Balance Model of a Continuous Crystallizer . . . . .	209
7.2.2	Bounded Lyapunov-Based Control . . . . .	211
7.2.3	Modeling Sensor Data Loss . . . . .	214
7.2.4	Simulation Results . . . . .	216
7.3	Handling Sensor Malfunctions: Batch Crystallizer . . . . .	220
7.3.1	Population Balance Model of a Protein Batch Crystallizer . . . . .	227
7.3.2	State Estimator Design . . . . .	230
7.3.3	Predictive Controller Formulation and Closed-Loop Results . . . . .	231
7.4	Conclusions . . . . .	234
<b>8</b>	<b>Analysis and Control of Mode Transitions in Biological Networks</b>	<b>237</b>
8.1	Introduction . . . . .	237
8.2	A Switched System Representation of Biological Networks . . . . .	242
8.3	Methodology for Analysis of Mode Transitions . . . . .	243
8.4	Application to Eukaryotic Cell Cycle Regulation . . . . .	249
8.5	Application to the Bacteriophage $\lambda$ -Switch System . . . . .	258
8.6	Conclusions . . . . .	278
<b>9</b>	<b>Conclusions</b>	<b>280</b>
	<b>Bibliography</b>	<b>283</b>

# List of Figures

1.1	Generic setup of point-to-point connection systems. . . . .	3
1.2	Generic setup of distributed control systems. . . . .	4
1.3	Block diagram [101] of a centralized networked control system for a single-unit plant. . . . .	5
1.4	Block diagram of a hierarchical distributed networked control architecture for a multi-unit plant. . . . .	6
2.1	Process flow diagram of two CSTR units in series. . . . .	17
2.2	Summary of the fault-tolerant control strategy, for a two-unit plant, using communication networks. . . . .	32
2.3	Fault-tolerant control structure for a single unit operation, integrating supervisory and feedback control over a communication network. . . .	35
2.4	Stability regions of the three control configurations (I, II, III) considered for the chemical reactor example of Equation 2.13. . . . .	38
2.5	Evolution of the closed-loop state profiles under repeated control system failures and subsequent switching by the supervisor from configuration 1 (solid lines) to configuration 2 (dashed lines) to configuration 3 (dotted lines). . . . .	40

2.6	Manipulated input profiles for each control configuration as the supervisor switches from configuration 1 to configuration 2 at $t = 2$ hr and from configuration 2 to configuration 3 at $t = 15$ hr. . . . .	41
2.7	A phase plot showing the closed-loop state trajectory leaving the intersection zone (I, II, & III) during the delay period (dashed-dotted trajectory) rendering configuration 3 destabilizing (dotted trajectory).	42
2.8	Evolution of the closed-loop state profiles when configuration 1 (solid lines) fails at $t = 10$ hr and an overall delay of $\tau_{max} = 8.0$ min elapses before the backup configuration is activated. Activation of configuration 2 preserves closed-loop stability (dashed lines) while activation of configuration 3 leads to instability (dotted lines). . . . .	43
2.9	Manipulated input profiles when configuration 1 fails at $t = 10$ hr and an overall delay of $\tau_{max} = 8.0$ min elapses before the backup configuration is activated. . . . .	44
2.10	Evolution of the closed-loop state and manipulated input profiles for CSTR 1 under a well-functioning control system (solid lines) and when the control actuator fail at $t = 5$ min (dashed lines). . . . .	46
2.11	Evolution of the closed-loop state and manipulated input profiles for CSTR 2 under a well-functioning control system. . . . .	47
2.12	Evolution of the closed-loop state and manipulated input profiles for CSTR 2 when the controller of the fall-back configuration ( $Q_2, C_{A03}$ ) is activated immediately after the failure (solid lines), and the open-loop state and input profiles resulting when the fall-back configuration is not activated after the failure (dashed lines). . . . .	48

2.13	Fault-recovery region of the fall-back control configuration $(Q_2, C_{A03})$ for CSTR 2, with constraints $ Q_2  \leq 2.8 \times 10^6 \text{ KJ/hr}$ and $ C_{A03} - C_{A03}^s  \leq 0.4 \text{ kmol/m}^3$ when failure occurs at $T_f = 5 \text{ min}$ . Activation of the fall-back configuration after a 3 min delay preserves closed-loop stability (top plot), while activation after 4.1 min delay fails to ensure fault-tolerance (bottom plot). . . . .	50
2.14	Evolution of the closed-loop state and input profiles when the failure occurs at $T_f = 5 \text{ min}$ and the fall-back configuration $(Q_2, C_{A03})$ , with constraints $ Q_2  \leq 2.8 \times 10^6 \text{ KJ/hr}$ and $ C_{A03} - C_{A03}^s  \leq 0.4 \text{ kmol/m}^3$ is activated after a total delay of 3 min (solid lines) and after a total delay of 4.1 min (dashed lines). . . . .	51
2.15	Fault-recovery region of the fall-back control configuration $(Q_2, C_{A03})$ for CSTR 2, with constraints $ Q_2  \leq 1.4 \times 10^7 \text{ KJ/hr}$ and $ C_{A03} - C_{A03}^s  \leq 2.0 \text{ kmol/m}^3$ when failure occurs at $T_f = 5 \text{ min}$ . Activation of the fall-back configuration after a delay of either 3 min or 4.1 min ensures fault-tolerance. . . . .	52
2.16	Evolution of the closed-loop state and manipulated input profiles when the failure occurs at $T_f = 5 \text{ min}$ and the fall-back configuration $(Q_2, C_{A03})$ , with constraints $ Q_2  \leq 1.4 \times 10^7 \text{ KJ/hr}$ and $ C_{A03} - C_{A03}^s  \leq 2.0 \text{ kmol/m}^3$ is activated after a total delay of 3 min (solid lines) and after a total delay of 4.1 min. (dashed lines). . . . .	53
3.1	Evolution of closed-loop state profiles subject to failure in control configuration 1 (solid line) under the switching rule of Theorem 3.1 (dotted line) and under arbitrary switching (dashed line). . . . .	72

3.2	Evolution of closed-loop (a) temperature and (b) concentration subject to failure in control configuration 1 (solid lines) under the switching rule of Theorem 3.1 (dotted lines) and under arbitrary switching (dashed lines). . . . .	73
3.3	Manipulated input profiles under (a) control configuration 1 (solid line), (b) control configuration 2 (under the switching rule of Theorem 3.1 (dotted line)), and (c) control configuration 3 (under arbitrary switching (dashed line)). . . . .	74
3.4	Evolution of closed-loop state profiles subject to failure in control configuration 1 (solid line) and switching to configuration 2 (dotted line) and, according to the switching rule of Theorem 3.1, to configuration 3 (dashed line). . . . .	75
3.5	Evolution of closed-loop (a) temperature and (b) concentration subject to failure in control configuration 1 (solid line) and switching to configuration 2 (dotted lines) and, according to the switching rule of Theorem 3.1, to configuration 3 (dashed lines). . . . .	76
3.6	Manipulated input profiles under (a) control configuration 1 (solid line), (b) control configuration 2 (dotted lines) and (c) according to the switching rule of Theorem 3.1 to control configuration 3 (dashed lines). . . . .	77
3.7	Evolution of closed-loop state profiles under the switching rule of Section 3.4.2 subject to failure in control system 1. . . . .	84
3.8	Evolution of closed-loop (a) temperature and (b) concentration under the switching rule of Section 3.4.2 subject to failure in control system 1.	85

3.9	Manipulated input profiles under (a) control configuration 1 and (b) control configuration 2 under the switching rule of Section 3.4.2 subject to failure in control system 1. . . . .	86
4.1	Integrated fault-detection and fault-tolerant control design: state feedback case. . . . .	97
4.2	Evolution of the closed-loop state profiles under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid line) and under arbitrary switching (dashed line). . . . .	102
4.3	Evolution of the closed-loop (a) temperature and (b) concentration under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines). . . . .	103
4.4	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 Equation 4.7 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines). . . . .	105
4.5	Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines). . . . .	106
4.6	Integrated fault-detection and fault-tolerant control design under output feedback. . . . .	112

4.7	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. . . . .	121
4.8	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. . . . .	122
4.9	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 Equation 4.21 subject to failures in control systems 1 and 2. . . . .	124
4.10	Evolution of the closed-loop state trajectory under the switching rule of Equation 4.21 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). . . . .	125
4.11	Evolution of the residual for (a) the first control configuration and (b) the second control configuration. . . . .	126

4.12	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 Equation 4.21 subject to failures in control systems 1 and 2 in the absence of a fault-detection filter. . . . .	127
4.13	Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Equation 4.21 subject to failures in control systems 1 and 2 in the presence (solid lines) and absence (dashed lines) of a fault-detection filter. . . . .	128
4.14	Flow diagram showing two CSTRs operating in series. . . . .	131
4.15	Two reactors in series scenario one: (a) temperature profile of reactor two with reconfiguration (solid line) and without reconfiguration (dotted line), (b) $Q_1$ residual profile, (c) $Q_2$ residual profile (note fault detection at time $t = 40.79$ min). . . . .	134
4.16	Two reactors in series scenario two: (a) temperature profile of reactor two with reconfiguration (solid line) and without reconfiguration (dotted line), (b) $Q_1$ residual profile, (c) $Q_2$ residual profile (note fault detection at time $t = 41.33$ min). . . . .	135
5.1	Industrial gas phase polyethylene reactor system. . . . .	140
5.2	Evolution of the open-loop process states. . . . .	153
5.3	Closed-loop state profiles under the primary control configuration. . .	154



5.4 Manipulated input profile under primary control configuration. . . . . 155

5.5 Evolution of the closed-loop state profiles under primary control configuration (dashed lines) and no fall-back control configuration available to switch to (or fall-back control configuration is not activated) resulting in open-loop oscillatory behavior (solid lines) after primary control configuration fails at  $T_{fault} = 5\text{ hrs } 34\text{ mins}$ . . . . . 156

5.6 Evolution of the closed-loop state profiles under primary control configuration (dashed lines) which fails at  $T_{fault} = 5\text{ hrs } 34\text{ mins}$ . At this point, the process starts operating open-loop (dotted lines). At  $T_{detect} = 5\text{ hrs } 54\text{ mins}$ , the detection filter verifies that there is a fault on the primary control configuration and the control system switches to the fall-back control configuration (solid lines). . . . . 157

5.7 Evolution of the detection filter residual value under primary control configuration (dashed line). At  $T_{detect} = 5\text{ hrs } 54\text{ mins}$ , the detection filter residual value reaches the detection threshold of 0.5 which verifies that a fault on the primary control configuration occurs. A switch to the fall-back control configuration (solid line) resets the detection filter residual back to zero. . . . . 159

5.8 Evolution of the closed-loop state profiles in the case of measurement noise under primary control configuration (dashed lines) which fails at  $T_{fault} = 5\text{ hrs } 34\text{ mins}$ . At this point, the process starts operating open-loop (dotted lines). At  $T_{detect} = 5\text{ hrs } 54\text{ mins}$ , the detection filter verifies that there is a fault on the primary control configuration and the control system switches to the fall-back control configuration (solid lines). . . . . 160

5.9	Evolution of the detection filter residual value in the case of measurement noise. A detection threshold of 0.5 triggers false alarm even before real fault on primary control configuration at $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ . A new detection threshold of 0.7 is picked and implemented. At $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$ , the detection filter residual value reaches the detection threshold of 0.7 which verifies that a fault on the primary control configuration occurs. A switch to the fall-back control configuration resets the detection filter residual back to normal. . . .	161
5.10	Evolution of the open-loop (dotted lines) and closed-loop (solid lines) state profiles under the primary control configuration in the presence of parametric model uncertainty and disturbances. . . . .	162
6.1	Evolution of the state profile under configuration 2 (dashed line) followed by loss of measurements (dotted line) and upon recovery reactivating configuration 2 (dash-dotted line), closed-loop stability is not preserved; however, switching to configuration 1 (solid line) preserves closed-loop stability. . . . .	177
6.2	Closed-loop system in the (a) absence, and (b) presence of sensor data losses. . . . .	179
6.3	Evolution of the state trajectory under control configuration 2 in the presence of sensor data loss (defined over a finite interval) at a rate of 0.4 (dashed line), sensor data loss (defined over an infinite interval) at a rate of 0.05 (dash-dotted line) and sensor data loss (defined over a finite interval) at a rate of 0.1 (solid line). . . . .	189

6.4	Manipulated input profile under control configuration 2 in the presence of sensor data loss (defined over a finite interval) at a rate of 0.4 (dashed line), sensor data loss (defined over an infinite interval) at a rate of 0.05 (dash-dotted line) and sensor data loss (defined over a finite interval) at a rate of 0.1 (solid line). . . . .	189
6.5	Evolution of the state trajectory: At $t = 13.5$ minutes the data loss rate goes up to 0.35 under configuration 2 (solid line). Keeping with configuration 2 (dotted line) or switching to configuration 3 (dashed line) does not preserve stability, while switching to configuration 1 (dash-dotted line) preserves stability. . . . .	194
6.6	Manipulate input profiles: At $t = 13.5$ minutes the data loss rate goes up to 0.35 under configuration 2 (solid line), switching to configuration 3 does not preserve stability (dashed line), while switching to configuration 1 (dash-dotted line) preserves stability. . . . .	195
6.7	Evolution of the closed-loop state profiles under primary control configuration under continuous measurements (solid lines) and sensor data loss rate of 0.75 (dotted lines). . . . .	198
6.8	Evolution of the manipulated input profiles under primary control configuration under continuous measurements. . . . .	199
6.9	Evolution of the manipulated input profiles under primary control configuration with sensor data loss rate of 0.75. . . . .	200
6.10	Evolution of the closed-loop state profiles under the primary configuration with the data loss rate increasing from 0.75 to 0.80 at 0.97 hours. . . . .	201

6.11	Evolution of the closed-loop state profiles under the reconfiguration law of Equation 6.11 with the data loss rate increasing from 0.75 to 0.80 at 0.97 hours. . . . .	203
6.12	Evolution of the closed-loop input profiles under the reconfiguration law of Equation 6.11 with the data loss rate increasing from 0.75 to 0.80 at 0.97 hours. . . . .	204
7.1	Closed-loop system in the (a) absence, and (b) presence of sensor data losses. . . . .	215
7.2	Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for $u_{max} = 2$ and no sensor data losses. . . . .	221
7.3	Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for $u_{max} = 2$ and 90% probability of sensor data losses. . . . .	222
7.4	Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for $u_{max} = 2$ and 95% probability of sensor data losses. . . . .	223
7.5	Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for $u_{max} = 4$ and no sensor data losses. . . . .	224
7.6	Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for $u_{max} = 4$ and 70% probability of sensor data losses. . . . .	225

7.7	Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for $u_{max} = 4$ and 75% probability of sensor data losses. . . . .	226
7.8	(a) Jacket temperature and (b) supersaturation profiles under output feedback control; sampling time of 5 minutes (solid lines), sampling time of 10 minutes without constraint modification (dashed lines) and sampling time of 10 minutes under the predictive controller with tightened constraints (dash-dotted lines). . . . .	235
8.1	Phase-plane portraits of the system of Equation 8.4, for different values of $k'_2$ , $k''_2$ , and $k_{wee}$ , showing: (a) Stable steady-state with most MPF inactive, (b) Stable steady-state with most MPF active, (c) Unstable steady-state surrounded by a limit cycle, and (d) Bi-stability: two stable steady-states separated by an unstable saddle point. . . . .	251
8.2	(a) A plot showing the overlap of the limit cycle of the oscillatory mode with the domains of attraction for the M-arrested steady-state (entire area above dashed curve) and for the G2-arrested steady-state (entire area below the dashed curve), (b) A plot showing that switching from the oscillatory to the bi-stable mode moves the system to different steady-states depending on where switching takes place. In both cases, the oscillatory mode is fixed at $k'_2 = 0.01$ , $k''_2 = 10$ , $k_{wee} = 2.0$ . . . . .	253

8.3	The time evolution plots of (a) active MPF, and (b) total cyclin upon switching from the oscillatory to the bi-stable mode at two representative switching times. At $t = 333.5$ min, the state trajectory lies on segment A (see Figure 8.2(a)) and therefore switching lands the state in the M-arrested steady-state (dash-dotted line), while at $t = 334$ min, switching lands the state in the G2-arrested steady-state (dotted line). In both cases, the oscillatory mode is fixed at $k'_2 = 0.01$ , $k''_2 = 10$ , $k_{wee} = 2.0$ . . . . .	255
8.4	A plot showing that switching from the oscillatory mode (of the following parameter values: $k'_2 = 0.01$ , $k''_2 = 10$ , $k_{wee} = 2.5$ ) to the bi-stable mode at same time as in Figure 8.2(b) ( $t = 333.5$ min) moves the system to G2-arrest steady-state (instead of M-arrest steady-state) because switching does not occur on segment B. Note that the portion of the limit cycle overlapping the domain of attraction of the M-arrested steady-state (segment B) is larger than the one in Figure 8.2(a) (segment A). . . . .	256
8.5	The time evolution plots of (a) active MPF, and (b) total cyclin upon switching from the oscillatory to the bi-stable mode at $t = 333.5$ min. In both cases, the oscillatory mode is fixed at $k'_2 = 0.01$ , $k''_2 = 10$ , $k_{wee} = 2.5$ . . . . .	257
8.6	A schematic representation of the molecular mechanism responsible for the lysogenic to lytic mode transition in the bacteriophage $\lambda$ . . . . .	259

8.7 A phase plot for the moderate *CI* degradation mode showing that an initial condition within the lysogenic domain of attraction (entire area below the dotted curve) will converge to the lysogenic steady-state (dashed trajectory) and that an initial condition within the lytic domain of attraction (entire area above the dotted curve) will converge to the lytic steady-state (solid trajectory). Here, the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 263

8.8 A phase plot showing the system of Equation 8.5 being initialized using  $\gamma_x = 0.05$  (dashed trajectory) and undergoing: (a) a decrease in the degradation rate of *CI* protein (to  $\gamma_x = 0.004$ ) at  $t = 20$ , leading the state to converge to the lysogenic steady-state, and (b) an increase in the degradation rate of *CI* protein (to  $\gamma_x = 0.1$ ) at  $t = 20$ , leading the state to converge to the lytic steady-state. In both cases, the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 264

8.9 The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.004$  mode at  $t = 20$  and converges (solid lines) to the lysogenic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 265

8.10 The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.1$  mode at  $t = 20$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 266

8.11	A phase plot showing the system of Equation 8.5 being initialized using $\gamma_y = 0.008$ (dashed trajectory) and undergoing: (a) a decrease in the degradation rate of <i>Cro</i> protein (to $\gamma_y = 0.0005$ ) at $t = 20$ , leading the state to converge to the lytic steady-state, and (b) an increase in the degradation rate of <i>Cro</i> protein (to $\gamma_y = 0.06$ ) at $t = 20$ , leading the state to converge to the lysogenic steady-state. In both cases, the <i>CI</i> degradation rate is fixed at $\gamma_x = 0.05$ . . . . .	268
8.12	The time evolution plots of the <i>CI</i> (left) and <i>Cro</i> (right) protein concentrations when the system initialized at $(x(0), y(0)) = (35, 18)$ undergoes a transition from the $\gamma_y = 0.008$ mode (dashed lines) to the $\gamma_y = 0.0005$ mode at $t = 20$ and converges (solid lines) to the lytic steady-state. The <i>CI</i> degradation rate is fixed at $\gamma_x = 0.05$ . . . . .	269
8.13	The time evolution plots of the <i>CI</i> (left) and <i>Cro</i> (right) protein concentrations when the system initialized at $(x(0), y(0)) = (35, 18)$ undergoes a transition from the $\gamma_y = 0.008$ mode (dashed lines) to the $\gamma_y = 0.06$ mode at $t = 20$ and converges (solid lines) to the lysogenic steady-state. The <i>CI</i> degradation rate is fixed at $\gamma_x = 0.05$ . . . . .	270
8.14	A phase plot showing the system undergoing a transition from the $\gamma_x = 0.05$ mode (dashed trajectory) to the $\gamma_x = 0.004$ at $t = 70$ and converging (solid trajectory) to the lytic steady-state. The <i>Cro</i> degradation rate is fixed at $\gamma_y = 0.008$ . . . . .	273



- 8.15 The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.004$  mode at  $t = 70$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 274
- 8.16 A phase plot showing the system of Equation 8.5 being initialized using  $\gamma_x = 0.05$  (dashed trajectory) and undergoing: (a) a decrease in the degradation rate of *CI* protein (to  $\gamma_x = 0.004$ ) at  $t = 40$  and (b) an increase in the degradation rate of *CI* protein (to  $\gamma_x = 0.1$ ) at  $t = 40$ , both leading the state to converge to the lysogenic steady-state. In both cases, the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 275
- 8.17 The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system initialized at  $(x(0), y(0)) = (35, 2)$  undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.004$  mode at  $t = 40$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 276
- 8.18 The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system initialized at  $(x(0), y(0)) = (35, 2)$  undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.1$  mode at  $t = 40$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . . . . . 277

# List of Tables

2.1	Process parameters and steady-state values for the chemical reactors of Equation 2.2. . . . .	15
2.2	Process parameters and steady-state values for the chemical reactor of Equation 2.13. . . . .	34
4.1	Process parameters and steady-state values for the chemical reactors of Equation 4.23. . . . .	130
5.1	Process variables. . . . .	142
5.2	Parameter values and units. . . . .	143
7.1	Process parameters of the continuous crystallizer. . . . .	210
7.2	Dimensionless parameter values of the continuous crystallizer. . . . .	218
7.3	Summary of $r^*$ values for different $u_{max}$ for the continuous crystallizer example. . . . .	221
7.4	Parameter values for the batch crystallizer model of Equations 7.14-7.18.228	
7.5	Parameter values for the Luenberger-type observer of Equation 7.22. .	230
8.1	Parameter values for the cell cycle model in Equation 8.4 [133]. . . .	250

8.2	Steady-state values $(u_s, v_s)$ for the cell cycle model for different values of $k'_2, k''_2$ and $k_{wee}$ . . . . .	252
8.3	Parameter values for the bacteriophage $\lambda$ model in Equation 8.5 [76].	261
8.4	Steady-state values $(x_s, y_s)$ for the lysogenic, lytic, and unstable steady-states for different values of $\gamma_x$ and $\gamma_y$ . . . . .	261
8.5	Lyapunov functions used in estimating the invariant set $\Omega_{lysogenic}$ for the lysogenic state and the invariant set $\Omega_{lytic}$ for the lytic state. . . .	262

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Professor Panagiotis D. Christofides, for his encouragement and support in the course of this project.

I am grateful to Dr. Nael H. El-Farra and Dr. Prashant Mhaskar for their advises and thoughts that have greatly inspired and influenced the outcome of this project.

I would like to thank Professor James F. Davis, Professor Gerassimos Orkoulas, and Professor Robert T. M'Closkey for their invaluable inputs and comments toward this project.

Finally, but most importantly, I am indebted to my parents and my wife, Suse Halim, for their support, encouragement, and dedication.

## VITA

May 1, 1979	Born, Jakarta, Indonesia
September 2002	Chemical Engineering Departmental Scholar University of California, Los Angeles, CA
December 2002	Bachelor of Science, Chemical Engineering University of California, Los Angeles, CA
December 2003	Master of Science, Chemical Engineering University of California, Los Angeles, CA
June 2005	Outstanding M.S. Student, Chemical Engineering University of California, Los Angeles, CA
June 2005	American Institute of Chemical Engineers Student Chapter's Teaching Assistant of the Year University of California, Los Angeles, CA
2006–2007	Chancellor's Dissertation Year Fellowship University of California, Los Angeles, CA
2003–2007	Ph.D. student, Chemical Engineering University of California, Los Angeles, CA

## PUBLICATIONS AND PRESENTATIONS

1. El-Farra, N. H., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Nonlinear Process Systems: Integrating Optimal Actuator/Sensor Placement, Feedback and Supervisory Control," American Institute of Chemical Engineers Annual Meeting, paper 440b, San Francisco, California, 2003.
2. El-Farra, N. H., A. Gani and P. D. Christofides, "Analysis of Biological Regulatory Networks using Hybrid Systems Theory," American Institute of Chemical Engineers Annual Meeting, paper 463h, San Francisco, California, 2003.

3. El-Farra, N. H., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Process Systems: Integrating Supervisory and Feedback Control Over Networks," *Proceedings of 7th International Symposium on Advanced Control of Chemical Processes*, 784–789, Hong Kong, P. R. China, 2004.
4. El-Farra, N. H., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Multi-Unit Process Systems Using Communication Networks," *Proceedings of 7th IFAC Symposium on Dynamics and Control of Process Systems*, paper 188, Cambridge, Massachusetts, 2004.
5. El-Farra, N. H., A. Gani and P. D. Christofides, "Analysis of Switching Biological Networks using Hybrid Systems Theory," *Proceedings of the American Control Conference*, Special Evening Session on "Systems Engineering of Systems Biology," Boston, Massachusetts, 2004. (Best Presentation in Session Award)
6. El-Farra, N. H., A. Gani and P. D. Christofides, "A Switched Dynamical Systems Approach for the Analysis and Control of Mode Transitions in Bacteriophage Lambda," American Institute of Chemical Engineers Annual Meeting, paper 442b, Austin, Texas, 2004.
7. El-Farra, N. H., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Process Systems Using Communication Networks," *American Institute of Chemical Engineers Journal*, 51(6): 1665–1682, 2005.
8. El-Farra, N. H., A. Gani and P. D. Christofides, "Analysis of Mode Transitions in Biological Networks," *American Institute of Chemical Engineers Journal*, 51(8): 2220–2234, 2005.
9. El-Farra, N. H., A. Gani and P. D. Christofides, "A Switched Systems Approach for the Analysis and Control of Mode Transitions in Biological Networks," *Pro-*

- ceedings of the American Control Conference*, 3247–3252, Portland, Oregon, 2005.
10. Gani, A., C. McFall, P. Mhaskar, P. D. Christofides and J. F. Davis, “Fault-Tolerant Process Control: Handling Asynchronous Sensor Behavior,” American Institute of Chemical Engineers Annual Meeting, paper 494e, San Francisco, California, 2006.
  11. Gani, A., P. Mhaskar, and P. D. Christofides, “Fault-Tolerant Control of Non-linear Process Systems: Handling Sensor Malfunctions,” American Institute of Chemical Engineers Annual Meeting, paper 374f, Cincinnati, Ohio, 2005.
  12. Gani, A., P. Mhaskar, and P. D. Christofides, “Fault-Tolerant Control of a Polyethylene Reactor,” American Institute of Chemical Engineers Annual Meeting, paper 402b, Cincinnati, Ohio, 2005.
  13. Gani, A., P. Mhaskar, and P. D. Christofides, “Predictive Control of Particulate Processes with Actuator/Sensor Faults,” American Institute of Chemical Engineers Annual Meeting, paper 498a, Cincinnati, Ohio, 2005.
  14. Gani, A., P. Mhaskar and P. D. Christofides, “Fault-Tolerant Control of a Polyethylene Reactor,” *Proceedings of the American Control Conference*, 6026–6032, Minneapolis, Minnesota, 2006.
  15. Gani, A., P. Mhaskar and P. D. Christofides, “Fault-Tolerant Control of a Polyethylene Reactor,” *Journal of Process Control*, 17(5): 439–451, 2007.
  16. Gani, A., P. Mhaskar and P. D. Christofides, “Handling Sensor Malfunctions in Control of Particulate Processes,” *Chemical Engineering Science*, 62: in press, 2007.

17. Gani, A., P. Mhaskar and P. D. Christofides, "Control of a Polyethylene Reactor: Handling Sensor Faults," *Proceedings of the American Control Conference*, 7 pages, New York City, New York, 2007.
18. McFall, C., A. Gani, P. Mhaskar, P. D. Christofides and J. F. Davis, "Fault-Tolerant Process Control: Nonlinear FDI and Reconfiguration," American Institute of Chemical Engineers Annual Meeting, paper 553g, Cincinnati, Ohio, 2005.
19. McFall, C., P. Mhaskar, A. Gani, P. D. Christofides and J. F. Davis, "Fault-Tolerant Output Feedback Control of Multivariable Nonlinear Processes," American Institute of Chemical Engineers Annual Meeting, paper 654c, San Francisco, California, 2006.
20. Mhaskar, P., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Nonlinear Processes: Performance-based Reconfiguration and Robustness," *International Journal of Robust and Nonlinear Control*, 16(3): 91–111, 2006.
21. Mhaskar P., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Nonlinear Process Systems: Performance-based Reconfiguration and Robustness," American Institute of Chemical Engineers Annual Meeting, paper 242g, Cincinnati, Ohio, 2005.
22. Mhaskar, P., A. Gani and P. D. Christofides, "Fault-Tolerant Control of Nonlinear Processes: Performance-based Reconfiguration and Robustness," *Proceedings of the American Control Conference*, 6020–6025, Minneapolis, Minnesota, 2006.
23. Mhaskar, P., A. Gani, N. H. El-Farra, P. D. Christofides and J. F. Davis, "Integrated Fault-Detection and Fault-Tolerant Control of Process Systems," *Pro-*



*ceedings of 16th International Federation of Automatic Control World Congress*, paper 4742, Prague, Czech Republic, 2005.

24. Mhaskar, P., A. Gani, N. H. El-Farra and P. D. Christofides, “Integrating Fault-Detection and Isolation and Fault-Tolerant Control for Process Systems,” American Institute of Chemical Engineers Annual Meeting, paper 404b, Austin, Texas, 2004.
25. Mhaskar, P., A. Gani, N. H. El-Farra, C. McFall, P. D. Christofides and J. F. Davis, “Integrated Fault-Detection and Fault-Tolerant Control of Process Systems,” *American Institute of Chemical Engineers Journal*, 52(6): 2129–2148, 2006.
26. Mhaskar, P., A. Gani, C. McFall, P. D. Christofides and J. F. Davis, “Fault-Tolerant Control of Nonlinear Systems Subject To Sensor Data Losses,” *Proceedings of 45th IEEE Conference on Decision and Control*, 3498-3505, San Diego, California, 2006.
27. Mhaskar, P., A. Gani, C. McFall, P. D. Christofides and J. F. Davis, “Fault-Tolerant Control of Nonlinear Process Systems Subject to Sensor Faults,” *American Institute of Chemical Engineers Journal*, 53(3): 654–668, 2007.
28. Mhaskar, P., C. McFall, A. Gani, P. D. Christofides and J. F. Davis, “Fault-Tolerant Control of Nonlinear Systems: Fault-Detection and Isolation and Controller Reconfiguration,” *Proceedings of the American Control Conference*, 5115–5122, Minneapolis, Minnesota, 2006. (Best Presentation in Session Award)
29. Muñoz de la Peña, D., A. Gani, P. Mhaskar and P. D. Christofides, “Output Feedback Control of Nonlinear Systems Subject to Constraints and Asynchro-

nous Measurements,” American Institute of Chemical Engineers Annual Meeting, paper 146f, San Francisco, California, 2006.

30. Ohran, B., A. Gani, C. McFall, P. Mhaskar, P. D. Christofides and J. F. Davis, “Uniting Data and Model-based Fault-Detection Filters for Fault-Tolerant Control of Process Systems,” American Institute of Chemical Engineers Annual Meeting, paper 125e, San Francisco, California, 2006.

## ABSTRACT OF THE DISSERTATION

Fault-Tolerant Process Control:  
Handling Actuator and Sensor Malfunctions

by

Adiwinata Gani

Doctor of Philosophy in Chemical Engineering  
University of California, Los Angeles, 2007  
Professor Panagiotis D. Christofides, Chair

Increasingly faced with the requirements of safety, reliability, and profitability, chemical process operation is relying extensively on highly automated process control systems. Automation, however, tends to increase vulnerability of the process to faults (for example, 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 process. Problems due to faults may include physical damage to the process 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 U.S. alone.

The above considerations provide a strong motivation for the development of meth-

ods and strategies for the design of advanced fault-tolerant control systems 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. To this end, the present doctoral dissertation focuses on the design of advanced fault-tolerant control systems for chemical processes which explicitly deal with actuator/controller failures and sensor data losses. Specifically, the dissertation proposes a methodology for the design of fault-tolerant control systems for nonlinear processes with actuator constraints and uncertainty in the presence of actuator and sensor faults, incorporating performance and robustness considerations. The proposed methodology employs a hybrid systems framework and is predicated upon the idea of integrating fault-detection, local feedback control, and supervisory control over networks. The efficacy and implementation of the proposed methodology are demonstrated through a single unit chemical reactor, a cascading multi-unit chemical reactor, a polyethylene reactor, batch and continuous crystallizers.

# Chapter 1

## Introduction

### 1.1 Background and Motivation for Fault-Tolerant Control

Safety and reliability are primary goals in the operation of industrial chemical plants. An important need currently exists for enhancing the safety and reliability of chemical plants in ways that reduce their vulnerability to serious failures. Increasingly faced with the requirements of operational flexibility under tight performance specifications and other economic drivers, plant operation is relying extensively on highly automated process control systems. Automation, however, tends to increase vulnerability of the plant to faults, such as defects/malfunctions in process equipment, sensors and actuators, failures in the controllers or in the control loops, which, if not appropriately handled in the control system design, can potentially cause a host of undesired economic, environmental, and safety problems that seriously degrade the operating efficiency of the plant. These considerations provide a strong motivation for the development of systematic methods and strategies for the design of fault-tolerant control systems and have motivated many research studies in this area (see, for example, [172, 179, 9] and [136, 18, 105] for references).

Given the complex dynamics of chemical processes (due, for example, to the presence of nonlinearities and constraints) and the geographically distributed, interconnected nature of plant units, as well as the large number of distributed sensors and actuators typically involved, the success of any fault-tolerant control strategy 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 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, and (3) the efficient exchange of information and communication between the different plant units through a high-level supervisor that coordinates the overall plant response in failure situations and minimizes the effects of failure propagation.

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 for chemical processes (for example, [88, 86, 46, 48, 163]) and advances in the analysis and control of hybrid process systems leading to the development of a systematic framework for the integration of feedback and supervisory control [47, 49]. A hybrid systems framework provides a natural setting for the analysis and design of fault-tolerant control systems since the occurrence of failure and subsequent switching to fall-back control configurations induce discrete transitions superimposed on the underlying continuous dynamics. Hybrid control techniques have been useful in dealing with a wide range of problems that cannot be addressed using classical control approaches, including fault-tolerant control of spatially-distributed systems (for example, [50, 53]), control of processes with switched dynamics (for example, [49, 13]), and the design of hybrid predictive con-

trol structures that overcome some of the limitations of classical predictive control algorithms (for example, [54]). In addition to control studies, research work on hybrid systems spans a diverse set of problems ranging from the modeling (for example, [177, 12]) and simulation (for example, [12, 68]) to the optimization (for example, [73, 72]) and stability analysis (for example, [80, 39]) of several classes of hybrid systems.

## 1.2 Background and Motivation for Networked Control Systems

A major trend in modern industrial and commercial systems is to integrate computing, communication, and control into different levels of plant operations and information process. The traditional communication architecture for control system, which has been successfully implemented in industry for decades, is a point-to-point connection system (Figure 1.1), that is, a wire connects the central control computer with each sensor or actuator point. However, a traditional centralized point-to-point control system is no longer suitable to meet new requirements, such as modularity, decentralization of control, integrated diagnostics, system agility, quick and easy maintenance, and low cost.

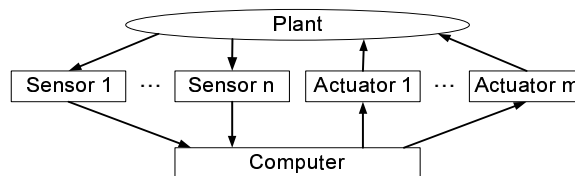


Figure 1.1: Generic setup of point-to-point connection systems.

Current industry application is a distributed control system (DCS), Figure 1.2, where several interacting computers connected to a serial network sharing the same

workload. However, control modules in a DCS are loosely connected because most of the real-time control tasks (sensing, calculation, and actuation) are carried out within the individual process stations themselves. Only on/off signals, monitoring information, alarm information, and the like are transmitted on the serial network.

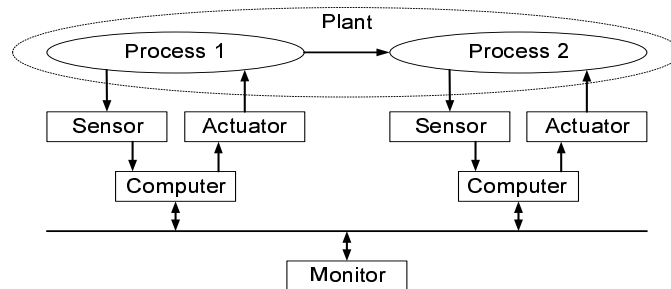


Figure 1.2: Generic setup of distributed control systems.

Recent innovations in actuator/sensor and communication technologies are increasingly enabling the integration of communication and control domains [181]. For example, the use of communication networks as media to interconnect the different components in an industrial control system is rapidly increasing and expected to replace the more costly point-to-point connection schemes currently employed in distributed control systems.

Figure 1.3 shows the basic networked control architecture within a single-unit plant with few actuators and sensors (centralized structure) and Figure 1.4 shows the basic networked control architecture for a larger plant with several interconnected processing units and larger number of actuators and sensors (distributed hierarchical structure). Currently, networked control systems is an active area of research within control engineering (for example, see [169, 124, 159, 135, 176, 130, 139, 143, 155] for some recent results and references in this area). In addition to the advantages of reduced system wiring (reduced installation, maintenance time and costs) in this architecture, the increased flexibility and ease of maintenance of a system using a



network to transfer information is an appealing goal. In the context of fault-tolerant control in particular, systems designed in this manner allow for easy modification of the control strategy by rerouting signals, having redundant systems that can be activated automatically when component failure occurs, and in general they allow having a high-level supervisor control over the entire plant. The appealing features of communication networks motivate investigating ways for integrating them in the design of fault-tolerant control systems to ensure a timely and coordinated response of the plant in ways that minimize the effects of failure propagation between plant units. This entails devising strategies to deal with some of the fundamental issues introduced by the network, including issues of bandwidth limitations, quantization effects, network scheduling, communication delays time-varying transmission period, unreliable transmission paths, and single-packet versus multiple-packet transmission, which continue to be topics of active research (see [185, 169, 130, 101] for further discussion on these issues).

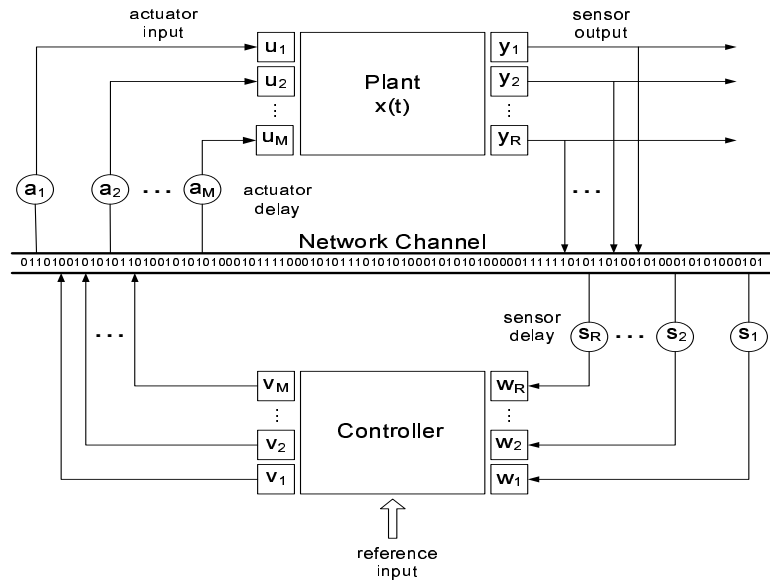


Figure 1.3: Block diagram [101] of a centralized networked control system for a single-unit plant.

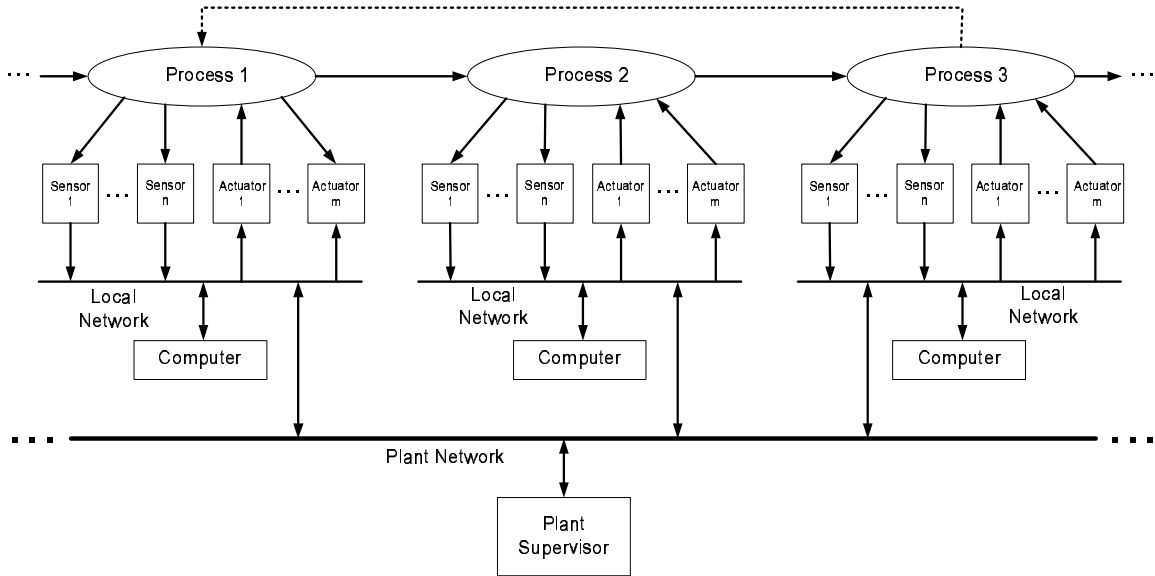


Figure 1.4: Block diagram of a hierarchical distributed networked control architecture for a multi-unit plant.

### 1.3 Dissertation Objectives and Structure

Motivated by the above considerations, we develop a fault-tolerant control methodology together with fault-detection and communication components for handling actuator and sensor malfunctions in the process systems, specifically in chemical processes. The rest of the dissertation is structured as follows.

In Chapter 2, we develop a fault-tolerant control system design methodology, for plants with multiple (distributed) interconnected processing units, that accounts explicitly for the inherent complexities in supervisory control and communication tasks resulting from the distributed interconnected nature of plant units. The proposed approach provides explicit guidelines for managing the interplays between the coupled tasks of feedback control, fault-tolerance, and communication.

In Chapter 3, we focus on incorporating performance and robustness considera-

tions for fault-tolerant control problems. Performance considerations are incorporated via the design of a Lyapunov-based predictive controller that enforces closed-loop stability from an explicitly characterized set of initial conditions. Robustness considerations are incorporated via the design of a robust hybrid predictive controller for each candidate control configuration that guarantees stability subject to uncertainty and constraints.

In Chapter 4, we consider the integration of fault-detection, feedback, and supervisory control. The fault-detection filter computes the expected closed-loop behavior in the absence of faults and the deviations of the process states from the expected closed-loop behavior are used to detect faults. This approach is demonstrated both for the state-feedback and the output-feedback problem.

In Chapter 5, we implement fault-detection and fault-tolerant control strategies to industrial gas phase polyethylene reactor modeled by seven nonlinear ordinary differential equations (ODEs). The effectiveness of the fault-tolerant control strategy and the applicability of the stability region concept toward a complex system are investigated as well as the presence of measurement noise and robustness issues.

In Chapter 6, we consider the problem of fault-tolerant control subject to input constraints and sensor faults (complete failure or intermittent unavailability of measurements). For each control configuration, the stability region and the maximum allowable data loss rate which preserves closed-loop stability is computed. Fault-tolerance to sensor faults can be achieved via controller reconfiguration that accounts for the nonlinearity of the system, the presence of constraints, and the maximum allowable data loss rate.

In Chapter 7, we focus on feedback control of particulate processes in the presence of sensor data losses. Two typical particulate processes, a continuous crystallizer and

a batch protein crystallizer, modeled by population balance models, are considered.

In Chapter 8, we extend the applicability of Lyapunov-based tools, hybrid systems theory, and concept of stability regions (discuss in Chapter 2) to biological networks. We present a methodology for the analysis and control of mode transitions in biological networks. The proposed method can provide both qualitative and quantitative insights into the description, analysis and manipulation of biological networks. From a practical point of view, these techniques could potentially reduce the degree of trial-and-error experimentation. More importantly, computational and theoretical approaches can lead to testable predictions regarding the current understanding of biological networks, which can serve as the basis for revising existing hypotheses.

Finally, in Chapter 9, we conclude this dissertation.

## Chapter 2

# Fault-Tolerant Control of Constrained Nonlinear Processes Using Communication Networks

### 2.1 Introduction

We develop in this chapter a fault-tolerant control system design methodology, for plants with multiple (distributed) interconnected processing units, that accounts explicitly for the inherent complexities in supervisory control and communication tasks resulting from the distributed interconnected nature of plant units. The approach brings together tools from Lyapunov-based control and hybrid systems theory and is based on a hierarchical distributed architecture that integrates lower-level feedback control of the individual units with upper-level logic-based supervisory control over communication networks. The local control systems consist each of a family of feedback control configurations together with a local supervisor that communicates with actuators and sensors, via a local communication network, to orchestrate the transition between control configurations, on the basis of their fault-recovery regions,

in the event of failures. The local supervisors communicate, through a plant-wide communication network, with a plant supervisor responsible for monitoring the different units and coordinating their responses in a way that minimizes the propagation of failure effects. The communication logic is designed to ensure efficient transmission of information between units while also respecting the inherent limitations in network resources by minimizing unnecessary network usage and accounting explicitly for the effects of possible delays due to fault-detection, control computations, network communication, and actuator activation. The proposed approach provides explicit guidelines for managing the interplays between the coupled tasks of feedback control, fault-tolerance and communication. The efficacy of the proposed approach is demonstrated through chemical process examples.

## 2.2 Preliminaries

### 2.2.1 System Description

We consider a plant composed of  $l$  connected processing units, each of which is modeled by a continuous-time multivariable nonlinear system with constraints on the manipulated inputs, and represented by the following state-space description:

$$\begin{aligned}
 \dot{x}_1 &= f_1^{k_1}(x_1) + G_1^{k_1}(x_1)u_1^{k_1} \\
 \dot{x}_2 &= f_2^{k_2}(x_2) + G_2^{k_2}(x_2)u_2^{k_2} + W_{2,1}^{k_2}(x_2)x_1 \\
 &\vdots \\
 \dot{x}_l &= f_l^{k_l}(x_l) + G_l^{k_l}(x_l)u_l^{k_l} + \sum_{p=1}^{l-1} W_{l,p}^{k_l}(x_l)x_p \\
 \|u_i^{k_i}\| &\leq u_{i,max}^{k_i} \\
 k_i(t) &\in \mathcal{K}_i := \{1, \dots, N_i\}, \quad N_i < \infty, \quad i = 1, \dots, l
 \end{aligned} \tag{2.1}$$

where  $x_i := [x_i^{(1)} \ x_i^{(2)} \ \dots \ x_i^{(n_i)}]^T \in \mathbb{R}^{n_i}$  denotes the vector of process state variables associated with the  $i$ -th processing unit,  $u_i^{k_i} := [u_{i,1}^{k_i} \ u_{i,2}^{k_i} \ \dots \ u_{i,m_i}^{k_i}]^T \in \mathbb{R}^{m_i}$  denotes the vector of constrained manipulated inputs associated with the  $k_i$ -th control configuration in the  $i$ -th processing unit,  $u_{i,max}^{k_i}$  is a positive real number that captures the maximum size of the vector of manipulated inputs dictated by the constraints,  $\|\cdot\|$  denotes the Euclidean norm of a vector, and  $N_i$  is the number of different control configurations that can be used to control the  $i$ -th processing unit. The index,  $k_i(t)$ , which takes values in the finite set  $\mathcal{K}_i$ , represents a discrete state that indexes the right-hand side of the set of differential equations in Equation 2.1. For each value that  $k_i$  assumes in  $\mathcal{K}_i$ , the  $i$ -th processing unit is controlled via a different set of manipulated inputs which define a given control configuration. For each unit, switching between the available  $N_i$  control configurations is controlled by a local supervisor that monitors the operation of the unit and orchestrates, accordingly, the transition between the different control configurations in the event of control system failures. This in turn determines the temporal evolution of the discrete state,  $k_i(t)$ , which takes the form of a piecewise constant function of time. The local supervisor ensures that only one control configuration is active at any given time, and allows only a finite number of switches over any finite interval of time.

Without loss of generality, it is assumed that  $x_i = 0$  is an equilibrium point of the uncontrolled  $i$ -th processing unit (i.e., with  $u_i^{k_i} = 0$ ) and that the vector functions,  $f_i^{k_i}(\cdot)$ , and the matrix functions,  $G_i^{k_i}(\cdot)$  and  $W_{j,p}^{k_j}(\cdot)$ , are sufficiently smooth on their domains of definition, for all  $k_i \in \mathcal{K}_i$ ,  $i = 1, \dots, l$ ,  $j = 2, \dots, l$ ,  $p = 1, \dots, l-1$ . For the  $j$ -th processing unit, the term,  $W_{j,p}^{k_j}(x_j)x_p$ , represents the connection that this unit has with the  $p$ -th unit upstream. Note from the summation notation in Equation 2.1 that each processing unit can in general be connected to all the units upstream

from it. Our nominal control objective (i.e., in the absence of control system failures) is to design, for each processing unit, a stabilizing feedback controller that enforces asymptotic stability of the origin of the closed-loop system in the presence of control actuator constraints. To simplify the presentation of our results, we will focus only on the state feedback problem where measurements of all process states are available for all times.

### **2.2.2 Problem Statement and Solution Overview**

Consider the plant of Equation 2.1 where, for each processing unit, a stabilizing feedback control system has been designed and implemented. Given some catastrophic fault – that has been detected and isolated – in the actuators of one of the control systems, our objective is to develop a plant-wide fault-tolerant control strategy that: (1) preserves closed-loop stability of the failing unit, if possible, and (2) minimizes the negative impact of this failure on the closed-loop stability of the remaining processing units downstream. To accomplish both of these objectives, we construct a hierarchical control structure that integrates lower-level feedback control of the individual units with upper-level logic-based supervisory control over communication networks. The local control system for each unit consists of a family of control configurations for each of which a stabilizing feedback controller is designed and the stability region is explicitly characterized. The actuators and sensors of each configuration are connected, via a local communication network, to a local supervisor that orchestrates switching between the constituent configurations, on the basis of the stability regions, in the event of failures. The local supervisors communicate, through a plant-wide communication network, with a plant supervisor responsible for monitoring the different units and coordinating their responses in a way that minimizes the propagation of



failure effects. The basic problem under investigation is how to coordinate the tasks of feedback, control system reconfiguration and communication, both at the local (processing unit) and plant-wide levels in a way that ensures timely recovery in the event of failure and preserves closed-loop stability.

**Remark 2.1** In the design of any fault-tolerant control system, an important task that precedes the control system reconfiguration is the task of fault-detection and isolation (FDI). There is an extensive body of literature on this topic including, for example, the design of fault-detection and isolation schemes based on fundamental process models (for example, [61, 37]) and statistical/pattern recognition and fault diagnosis techniques (for example, [170, 74, 35, 132, 6, 168, 166, 167]). In this chapter, we focus mainly on the interplay between the communication network and the control system reconfiguration task. To this end, we assume that the FDI tasks take place at a time scale that is very fast compared to the time constant of the overall process dynamics and the time needed for the control system reconfiguration, and thus can be treated separately from the control system reconfiguration (we note that the time needed for FDI is accounted for in the control system reconfiguration through a time-delay; see the next section and the simulation studies for details). In the context of process control applications, this sequential and decoupled treatment of FDI and control system reconfiguration is further justified by the overall slow dynamics of chemical plants. Integration of fault-detection and fault-tolerant control will be covered in Chapter 4.

### 2.2.3 Motivating Example

In this section, we introduce a simple benchmark example [52] that will be revisited later to illustrate the design and implementation aspects of the fault-tolerant control design methodology to be proposed in the next section. While the discussion will center around this example, we note that the proposed framework can be applied to more

complex plants involving more complex arrangements of processing units as shown in Equation 2.1. To this end, consider two well-mixed, non-isothermal continuous stirred tank reactors (CSTRs) in series, 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. The feed to CSTR 1 consists of pure  $A$  at flow rate  $F_0$ , molar concentration  $C_{A0}$  and temperature  $T_0$ , and the feed to CSTR 2 consists of the output of CSTR 1 and an additional fresh stream feeding pure  $A$  at flow rate  $F_3$ , molar concentration  $C_{A03}$  and temperature  $T_{03}$ . Due to the non-isothermal nature of the reactions, a jacket is used to remove/provide heat to both reactors. Under standard modeling assumptions, a mathematical model of the plant can be derived from material and energy balances and takes the following form:

$$\begin{aligned}
\frac{dT_1}{dt} &= \frac{F_0}{V_1}(T_0 - T_1) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_{A1}, T_1) + \frac{Q_1}{\rho c_p V_1} \\
\frac{dC_{A1}}{dt} &= \frac{F_0}{V_1}(C_{A0} - C_{A1}) - \sum_{i=1}^3 R_i(C_{A1}, T_1) \\
\frac{dT_2}{dt} &= \frac{F_1}{V_2}(T_1 - T_2) + \frac{F_3}{V_2}(T_{03} - T_2) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_{A2}, T_2) + \frac{Q_2}{\rho c_p V_2} \\
\frac{dC_{A2}}{dt} &= \frac{F_1}{V_2}(C_{A1} - C_{A2}) + \frac{F_3}{V_2}(C_{A03} - C_{A2}) - \sum_{i=1}^3 R_i(C_{A2}, T_2)
\end{aligned} \tag{2.2}$$

where  $R_i(C_{Aj}, T_j) = k_{i0} \exp\left(\frac{-E_i}{RT_j}\right) C_{Aj}$ , for  $j = 1, 2$ .  $T$ ,  $C_A$ ,  $Q$ , and  $V$  denote the temperature of the reactor, the concentration of species  $A$ , the rate of heat input/removal from the reactor, and the volume of reactor, respectively, with subscript 1 denoting CSTR 1 and subscript 2 denoting CSTR 2.  $\Delta H_i$ ,  $k_i$ ,  $E_i$ ,  $i = 1, 2, 3$ , denote the enthalpies, pre-exponential constants and activation energies of the three reactions, respectively,  $c_p$  and  $\rho$  denote the heat capacity and density of fluid in the reactor. Using typical values for the process parameters (see Table 2.1), CSTR 1, with  $Q_1 = 0$ , has three steady-states: two locally asymptotically stable and one unstable

at  $(T_1^s, C_{A1}^s) = (388.57 \text{ K}, 3.59 \text{ kmol/m}^3)$ . The unstable steady-state of CSTR 1 corresponds to three steady-states for CSTR 2 (with  $Q_2 = 0$ ), one of which is unstable at  $(T_2^s, C_{A2}^s) = (429.24 \text{ K}, 2.55 \text{ kmol/m}^3)$ .

Table 2.1: Process parameters and steady-state values for the chemical reactors of Equation 2.2.

$F_0$	=	4.998	$m^3/hr$
$F_1$	=	4.998	$m^3/hr$
$F_3$	=	30.0	$m^3/hr$
$V_1$	=	1.0	$m^3$
$V_2$	=	3.0	$m^3$
$R$	=	8.314	$KJ/kmol \cdot K$
$T_0$	=	300.0	$K$
$T_{03}$	=	300.0	$K$
$C_{A0}$	=	4.0	$kmol/m^3$
$C_{A03}^s$	=	3.0	$kmol/m^3$
$\Delta H_1$	=	$-5.0 \times 10^4$	$KJ/kmol$
$\Delta H_2$	=	$-5.2 \times 10^4$	$KJ/kmol$
$\Delta H_3$	=	$-5.4 \times 10^4$	$KJ/kmol$
$k_{10}$	=	$3.0 \times 10^6$	$hr^{-1}$
$k_{20}$	=	$3.0 \times 10^5$	$hr^{-1}$
$k_{30}$	=	$3.0 \times 10^5$	$hr^{-1}$
$E_1$	=	$5.0 \times 10^4$	$KJ/kmol$
$E_2$	=	$7.53 \times 10^4$	$KJ/kmol$
$E_3$	=	$7.53 \times 10^4$	$KJ/kmol$
$\rho$	=	1000.0	$kg/m^3$
$c_p$	=	0.231	$KJ/kg \cdot K$
$T_1^s$	=	388.57	$K$
$C_{A1}^s$	=	3.59	$kmol/m^3$
$T_2^s$	=	429.24	$K$
$C_{A2}^s$	=	2.55	$kmol/m^3$

The control objective is to stabilize both reactors at the (open-loop) unstable steady-states. Operation at these points is typically sought to avoid high temperatures, while simultaneously achieving reasonable conversion. To accomplish the control objective under normal conditions (with no failures), we choose as manipulated

inputs the rates of heat input,  $u_1^1 = Q_1$ , subject to the constraint  $|Q_1| \leq u_{max}^{Q_1} = 2.7 \times 10^6 \text{ KJ/hr}$  and  $u_1^2 = Q_2$ , subject to the constraint  $|Q_2| \leq u_{max}^{Q_2} = 2.8 \times 10^6 \text{ KJ/hr}$ .

As shown in Figure 2.1, each unit has a local control system with its sensors and actuators connected through a communication network. The local control systems in turn communicate with the plant supervisor (and with each other) through a plant-wide communication network. Note that in designing each control system, only measurements of the local process variables are used (for example, the controller for the second unit uses only measurements of  $T_2$  and  $C_{A2}$ ). This decentralized architecture is intended to minimize unnecessary communication costs incurred by continuously sending measurement data from the first to the second unit over the network. We note that while this issue may not be a pressing one for the small plant considered here (where a centralized structure can in fact be easily designed), real plants nonetheless involve a far more complex arrangement of units with thousands of actuators and sensors, which makes the complexity of a centralized structure as well as the cost of using the network to share measurements between units quite significant. For this reason, we choose the distributed structure in Figure 2.1 in order to highlight some of the manifestations of the inherent interplays between the control and communication tasks.

The fault-tolerant control problem under consideration involves a total failure in both control systems after some time of startup, with the failure in the first unit being permanent. Our objective will be to preserve closed-loop stability of CSTR 2 by switching to an alternative control configuration involving, as manipulated variables, the rate of heat input,  $u_2^1 = Q_2$ , subject to the same constraint, and the inlet reactant concentration,  $u_2^2 = C_{A03} - C_{A03}^s$ , subject to the constraint  $|C_{A03} - C_{A03}^s| \leq u_{max}^{C_{A03}} = 0.4 \text{ kmol/m}^3$  where  $C_{A03}^s = 3.0 \text{ kmol/m}^3$ . The main question, which we address in

the next section, is how to devise the switching and network communication logics in a way that ensures fault-tolerance in the second unit and, simultaneously, accounts for the inherent limitations in network resources and possible delays in fault-detection, communication and actuator activation.

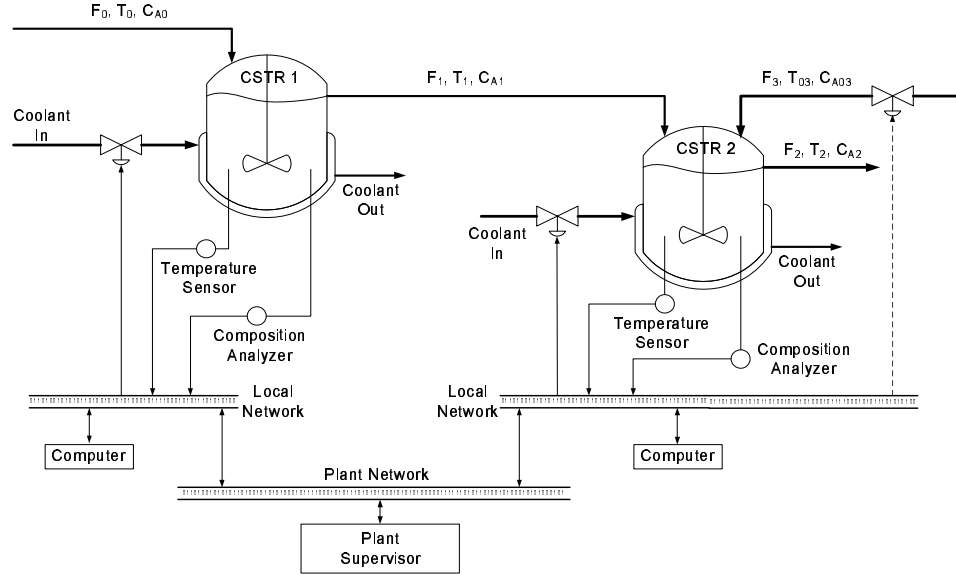


Figure 2.1: Process flow diagram of two CSTR units in series.

## 2.3 Fault-Tolerant Control System Design Methodology

In this section, we outline the main steps involved in the fault-tolerant control system design procedure. These include: (1) the synthesis of a stabilizing feedback controller for each of the available fall-back control configurations, (2) the explicit characterization of the stability region for each configuration which characterize the operating conditions for which fault-recovery can be guaranteed, (3) the design of a switching law that orchestrates the re-configuration of the failing control system in a way that safeguards closed-loop stability in the event of failures, and (4) the design of the network communication logic in a way that minimizes the propagation of failure effects

between plant units while also accounting for bandwidth constraints and delays. A major feature of the design methodology is the inherent coupling between the aforementioned tasks whereby each task affects how the rest are carried out. Below is a more detailed description of each step and a discussion on how the tradeoffs between the different steps are managed.

### 2.3.1 Constrained Feedback Controller Synthesis

Referring to the system of Equation 2.1, consider first the case when no failures take place anywhere in the plant. Under such conditions, our objective is to design, for each processing unit, a “nominal” feedback controller that enforces asymptotic closed-loop stability and provides an explicit characterization of the stability region under actuator constraints. One way to do this is to use Lyapunov-based control techniques. Specifically, consider the nonlinear system describing the  $i$ -th processing unit under the  $k_i$ -th control configuration, for which a control Lyapunov function,  $V_i^{k_i}$ , is available. Using this function, one can construct the following bounded nonlinear control law (see [103, 46]):

$$u_i^{k_i} = -r(x_i, u_{i,max}^{k_i})\beta^T(x_i) \quad (2.3)$$

where

$$r(x_i, u_{i,max}^{k_i}) = \frac{\alpha^*(x_i) + \sqrt{(\alpha^*(x_i))^2 + (u_{i,max}^{k_i}\|\beta^T(x_i)\|)^4}}{\|\beta^T(x_i)\|^2 \left[1 + \sqrt{1 + (u_{i,max}^{k_i}\|\beta^T(x_i)\|)^2}\right]} \quad (2.4)$$

$\alpha^*(x_i) = \alpha(x_i) + \rho_i^{k_i}\|x_i\|^2$ ,  $\rho_i^{k_i} > 0$  is a real number,  $\alpha(x_i) = L_{f_i^{k_i}}V_i^{k_i}(x_i)$ ,  $\beta^T(x_i) = (L_{G_i^{k_i}}V_i^{k_i})^T(x_i)$ , the notation  $L_{f_i^{k_i}}V_i^{k_i}$  is used to denote the Lie derivative of the scalar function,  $V_i^{k_i}$ , with respect to the vector field,  $f_i^{k_i}$ , and  $L_{G_i^{k_i}}V_i^{k_i}$  is a row vector whose constituent components are the Lie derivatives of  $V_i^{k_i}$  along the column vectors of the matrix  $G_i^{k_i}$ . Note that the control law of Equations 2.3-2.4 requires measure-

ments of the local process state variables,  $x_i$ , only and not measurements from other plant units upstream. This fully decentralized design is motivated by the desire to minimize unnecessary communication costs which would be incurred when sharing measurement data between the different units over the communication network. By disregarding the interconnections between the units in the controller design, however, closed-loop stability for a given unit rests on the stability properties of the upstream units. In particular, using a combination of Lyapunov and small-gain theorem type arguments, one can show that, starting from any invariant subset (for example, a level-set of  $V_i^{k_i}$ ) of the region described by:

$$\Phi_i(u_{i,max}^{k_i}) := \{x_i \in \mathbb{R}^{n_i} : \alpha(x_i) + \rho_i^{k_i} \|x_i\|^2 \leq u_{i,max}^{k_i} \|\beta^T(x_i)\|\}, \quad (2.5)$$

the control law of Equations 2.3-2.4 asymptotically stabilizes the  $i$ -th unit, under the  $k_i$ -th control configuration, at the origin provided that the closed-loop states of the upstream units,  $x_1, x_2, \dots, x_{i-1}$ , converge asymptotically to the origin. In this case, and because of the way the various units are connected (see Equation 2.1), the closed-loop states of the upstream units can be viewed as bounded vanishing perturbations that affect the  $i$ -th unit and, therefore, a control law that asymptotically stabilizes the unperturbed  $i$ -th unit (i.e., disregarding the upstream states) also stabilizes the closed-loop system when the perturbations (connections) are added.

Having designed the nominal feedback control systems, we now proceed to consider the effect of control actuator failure on the feedback controller design for each unit. To this end, let us consider a total failure in the actuators of the  $k_i$ -th control configuration in the  $i$ -th control system. This failure, if not addressed properly, can lead to closed-loop instabilities both within the  $i$ -th processing unit itself (where the failure has occurred) and within all the remaining units downstream. Minimizing the effects of failure propagation throughout the plant can be achieved in one of two

ways. The first involves reconfiguring the local control system of the  $i$ -th unit - once the failure is detected and isolated - by appropriately switching from the malfunctioning control configuration to some well-functioning fall-back configuration (recall that each processing unit has a family of control configurations). If this is feasible and can be done sufficiently fast, then the inherent fault-tolerance of the local control system is sufficient to preserve closed-loop stability not only for the  $i$ -th unit with the failing control system but also for the other units downstream without having to reconfigure their control systems. However, if local fault-recovery is not possible (this can happen, for example, in cases when the failure occurs at times that the state lies outside the stability regions of all the available fall-back control configurations; see the next subsection for details), then it becomes necessary to communicate the failure information to the control systems downstream and reconfigure them in order to preserve their closed-loop stability.

The main issue here is how to design the feedback control law for a given fall-back configuration in the units downstream in a way that respects the actuators' constraints and guarantees closed-loop stability despite the failure in the control system of some upstream unit. The choice of the feedback law depends on our choice of the communication policy. To explain this interdependence, we first note that a total failure in the control system of the  $i$ -th unit will cause its state,  $x_i$ , to move away from the origin (possibly settling at some other steady-state). Therefore, unless the nominal feedback controllers for the downstream units,  $i + 1, i + 2, \dots, l$ , are re-designed to account for this incoming "disturbance", the evolution of their states,  $x_{i+1}, x_{i+2}, \dots, x_l$ , will be adversely affected driving them away from the origin. To account for the disturbance caused by the upstream control system failure, one option is to send available measurements of  $x_i$ , through the communication network, to the



affected units and redesign their controllers accordingly. From a communications cost point of view, however, this option may be costly since it requires continued usage of the network resources after the failure, which can adversely affect the performance of other units sharing the same communication medium due to bandwidth limitations and overall delays.

To reduce unnecessary network usage, we propose an alternative approach where the failure in the  $i$ -th processing unit is viewed as a bounded non-vanishing disturbance affecting units  $i + 1, i + 2, \dots, l$ , and use the available process models of these units to capture, or estimate, the size of this disturbance (by comparing, for example, the evolution of the process variables for the  $i$ -th unit under the failed and well-functioning control configurations through simulations). In this formulation, state measurements from the  $i$ -th unit need not be shared with the other units; instead, only bounds on the disturbance size are transmitted to the downstream units. This approach involves using the network only once at the failure time and not continuously thereafter. The disturbance information can then be used to design an appropriate robust controller for each downstream unit to attenuate the effect of the incoming disturbance and enforce robust closed-loop stability. To illustrate how this can be done, let us assume that the failure in the control system of unit  $i$  occurs at  $t = T_f$  and that the failure is detected immediately (the effect of possible delays in fault-detection and how to account for them are discussed below in the subsection on communication logic design). Consider some unit,  $j$ , downstream from the  $i$ -th unit, that is described by the following model

$$\dot{x}_j = f_j^{k_j}(x_j) + G_j^{k_j}(x_j)u_j^{k_j} + \delta_i \sum_{p=1}^{i-1} W_{j,p}^{k_j}(x_j)x_p + \sum_{p=i}^{j-1} W_{j,p}^{k_j}(x_j)\theta_p \quad (2.6)$$

for  $i = 1, \dots, l - 1, j = i + 1, \dots, l$ , where  $\delta_i = 0$  for  $i = 1$ , and  $\delta_i = 1$  for  $i = 2, \dots, l - 1$ . The third term on the right-hand side of Equation 2.6 describes

the input from all the units upstream of unit  $i$ . The  $\theta_p$ 's are time-varying, but bounded, functions of time that describe the evolution of the states of the  $i$ -th unit and all the units downstream from unit  $i$  but upstream from unit  $j$  (i.e.,  $\theta_p(t) = x_p(t)$ ,  $p = i, \dots, j-1$ ). The choice of using the notation  $\theta_p$ , instead of  $x_p$ , for units  $i, \dots, j-1$  is intended to distinguish the effect of these units (where the failure originates and propagates downstream) as non-vanishing disturbances to the  $j$ -th unit, compared with the units upstream from unit  $i$  which are unaffected by the failure. Note that for unit  $j = i + 1$ , which immediately follows the failing unit, the only source of disturbances that should be accounted for in its controller design is that coming from the  $i$ -th unit with the failing control system. However, for units that lie further downstream, i.e., for  $j = i + 2, \dots, l$ , the controller design needs to account for the additional disturbances resulting from the effect of the failure on the intermediate units separating units  $i$  and  $j$ .

For a system of the form of Equation 2.6, one possible choice of a stabilizing controller is the following bounded robust Lyapunov-based control law proposed in [48] which has the general form:

$$u_j^{k_j} = -r_j(x_j, u_{j,max}^{k_j}, \theta_b) \beta^T(x_j), \quad (2.7)$$

where

$$r_j(x_j, u_{j,max}^{k_j}, \theta_b) = \frac{\alpha_1(x_j) + \sqrt{(\alpha_2(x_j))^2 + (u_{j,max}^{k_j} \|\beta^T(x_j)\|)^4}}{(\|\beta^T(x_j)\|)^2 \left[ 1 + \sqrt{1 + (u_{j,max}^{k_j} \|\beta^T(x_j)\|)^2} \right]} \quad (2.8)$$

$$\alpha_1(x_j) = \alpha(x_j) + \left( \rho_j^{k_j} \|x_j\| + \sum_{p=i}^{j-1} \chi_j^{k_j} \theta_b^p(T_f) \|\omega_p^T(x_j)\| \right) \left( \frac{\|x_j\|}{\|x_j\| + \phi_j^{k_j}} \right) \quad (2.9)$$

$$\alpha_2(x_j) = \alpha(x_j) + \rho_j^{k_j} \|x_j\| + \sum_{p=i}^{j-1} \chi_j^{k_j} \theta_b^p(T_f) \|\omega_p^T(x_j)\| \quad (2.10)$$

$\theta_b^p(T_f) := \max_{t \geq T_f} \|x_p(t)\|$ ,  $p = i, \dots, j - 1$  are positive real numbers that capture the size of the disturbances, originating from the failure in the control system of the  $i$ -th unit and propagating downstream,  $\omega_p(x_j) = (L_{W_{j,p}^{k_j}} V_j^{k_j})(x_j)$  is a row vector whose constituent components are the Lie derivatives of  $V_j^{k_j}$  along the column vectors of the matrix  $W_{j,p}^{k_j}$ ,  $V_j^{k_j}$  is a robust control Lyapunov function for the  $j$ -th system under the  $k_j$ -th control configuration, and  $\rho_j^{k_j} > 0$ ,  $\chi_j^{k_j} > 1$ ,  $\phi_j^{k_j} > 0$  are tuning parameters. Estimates of the disturbance bounds,  $\theta_b^p$ , can be obtained by comparing, through simulations for example, the responses of the  $p$ -th unit under the pre- and post-failure configurations (see the simulation studies section for an example). It should be noted that since all the incoming disturbances to unit  $j$  take effect only after  $T_f$ , the controller of Equations 2.7-2.10 is implemented only for  $t \geq T_f$ . For  $t < T_f$ , the nominal controllers of Equations 2.3-2.4 are used.

**Remark 2.2** When compared with the nominal controller of Equations 2.3-2.4, we observe that the nonlinear gain function for the fall-back controller,  $r_j(\cdot)$  in Equations 2.7-2.10, depends not only on the size of actuator constraints,  $u_{j,max}^{k_j}$ , and the particular fall-back control configuration being used,  $k_j$ , but also on the size of the disturbances caused by the occurrence of failure,  $\theta_b^p$ . This gain re-shaping procedure is carried out in order to guarantee constraint satisfaction and enforce robust closed-loop stability, with an arbitrary degree of attenuation of the effect of the failure on the  $j$ -th unit downstream. Note that, owing to the assumption of a persistent failure in the  $i$ -th unit (i.e., a non-vanishing disturbance), asymptotic closed-loop stability cannot be achieved for any of the units downstream. Instead, practical stability can be enforced whereby the states of each unit are driven, in finite-time, to a neighborhood of the origin whose size can be made arbitrarily small by selecting the controller tuning parameters  $(\rho_j^{k_j}, \chi_j^{k_j}, \phi_j^{k_j})$  appropriately (see [50] for a detailed proof). These closed-loop properties are enforced within a well-defined state-space region that is explicitly characterized in the next subsection.

**Remark 2.3** Note that since the processing units upstream of unit  $i$  are not affected by its failing control system, the nominal controllers designed for these units (see Equations 2.3-2.4) will asymptotically stabilize their states,  $x_p, p = 1, \dots, i-1$ , at the origin regardless of the failure; hence these state can be viewed as bounded vanishing inputs to the  $j$ -th unit and thus need not be accounted for in the controller design. The terms describing the intermediate units,  $p = i + 1, \dots, j - 1$  cannot however be treated as vanishing inputs. The reason is that even if the control systems of these units are immediately and appropriately re-configured to suppress the effect of the failure, their controllers, as discussed above, will at best be able to drive the states of these units, in finite time, only near the origin without achieving asymptotic convergence.

**Remark 2.4** It should be noted that the fault-tolerant control system design methodology proposed in this section is not restricted to the use of the bounded controller designs given in Equations 2.3-2.4 (for the nominal case) and in Equations 2.7-2.10 (for the case with failure). Any other stabilizing controller design that accounts for the constraints, enforces the desired robustness properties under failure, and provides an explicit characterization of the stability region can be used, including recently-developed hybrid predictive control algorithms [55, 114, 54, 116] which embed the implementation of predictive controllers within the explicitly-characterized stability region of Lyapunov-based nonlinear bounded controllers.

**Remark 2.5** Control Lyapunov Function (CLF)-based stabilization of nonlinear systems has been studied extensively in the nonlinear control literature (for example, see [103, 62, 149]). 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 to construct CLFs. In this chapter, the bounded controllers in Equations 2.3-2.4 and Equations 2.7-2.10 are designed using unconstrained CLFs, which are also used to explicitly characterize the associated stability regions. 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 estimates using other methods such as the Zubov’s method [42] and a combination of several CLFs which can yield substantially less conservative estimates.

**Remark 2.6** The treatment of the failure in the control system of unit  $i$  as a bounded disturbance is rooted in the assumption that  $x_i$ , while moving away from the origin after failure, will eventually settle at some other (undesirable) steady-state (recall that this is how the disturbance bound is computed). In the case when the  $i$ -th processing unit has only a single steady-state in the post-failure configuration, however, the failure cannot be treated as a bounded disturbance since  $x_i$  will simply grow unbounded after the failure and not settle anywhere. In such a case, unless the control system of unit  $i$  is fixed in time, a shutdown of the plant will be unavoidable.

### 2.3.2 Characterization of Fault-Recovery Regions

Consider once again the  $j$ -th processing unit described by the model of Equation 2.6. In the previous section, we outlined how to design, for a given fall-back control configuration,  $k_j \in \mathcal{K}_j$ , a robust feedback controller that, when implemented, can preserve closed-loop stability for this unit in the event of control system failure in some upstream unit,  $i$ . Given that actuator constraints place fundamental limitations on the ability of the controller to steer the closed-loop dynamics at will, it is important for the control system designer to explicitly characterize these limitations by identifying, or estimating, the set of admissible states starting from where the controller of Equations 2.7-2.10 is guaranteed to robustly stabilize the closed-loop system for unit

$j$  (region of robust closed-loop stability). Since suppression of the upstream failure effects on unit  $j$  is formulated as a robust stabilization problem, we shall refer to the robust stability region associated with any of the fall-back configurations also as the fault-recovery region. As discussed in the next subsection, the characterization of this region plays a central role in devising the appropriate switching policy that reconfigures the control system and ensures fault-recovery.

For the class of robust control laws given in Equations 2.7-2.10, using a Lyapunov argument one can show that the set

$$\begin{aligned} \Pi_j^{k_j}(u_{j,max}^{k_j}, \theta_b(T_f)) &:= \{x_j \in \mathbb{R}^{n_j} : \alpha(x_j) + \rho_j^{k_j} \|x_j\| + \sum_{p=i}^{j-1} \chi_j^{k_j} \theta_b^p(T_f) \|\omega_p^T(x_j)\| \\ &\leq u_{j,max}^{k_j} \|\beta^T(x_j)\|\} \end{aligned} \quad (2.11)$$

describes a region in the state-space where the control action satisfies the constraints and the Lyapunov function decays monotonically along the trajectories of the closed-loop system (see [48] for the detailed mathematical analysis). Note that the size of this set depends both on the magnitude of the constraints and the size of the disturbance (which in turn depends on the failure time,  $T_f$ ). In particular, as the constraints become tighter and/or the disturbances greater, the set becomes smaller. Since  $\Pi_j^{k_j}$ , however, is in general, not an invariant set, there is no guarantee that a trajectory starting within  $\Pi_j^{k_j}$  will remain within it for all the times that the  $k_j$ -th control configuration is active, that is,  $\Pi_j^{k_j}$  by itself is not necessarily a stability region. One way to estimate the fault-recovery region associated with a given control configuration using Equation 2.11 is to construct an invariant subset – preferably the largest – within  $\Pi_j^{k_j}$ , which we denote by  $\Omega_j^{k_j}(u_{j,max}^{k_j}, \theta_b(T_f))$  (for example,  $\Omega_j^{k_j}$  can be chosen as a level-set of  $V_j^{k_j}$ ). For a given fall-back configuration,  $k_j$ , implementation of the controller of Equations 2.7-2.10 at any time that the state is within  $\Omega_j^{k_j}$  ensures that the closed-loop trajectory stays within the region defined by  $\Pi_j$  – and, hence  $V_j^{k_j}$

continues to decay monotonically outside of a small neighborhood around the origin – for all the times that the  $k_j$ -th configuration is active. The estimate provided by  $\Omega_j^{k_j}$  can be conservative but can also be improved using computer simulations. This approach was followed in the simulation examples in order to obtain appropriate estimates of the fault-recovery regions.

**Remark 2.7** Note that, unlike the nominal stability regions associated with the nominal controllers of Equations 2.3-2.4 and obtained from Equation 2.5, the fault-recovery region of any downstream unit,  $j$ , cannot be computed a priori (i.e., before plant startup) since this region, as can be seen from Equation 2.11, depends on the failure time which is unknown prior to startup. However, once the failure occurs, estimates of the disturbance bounds can be computed by the local supervisors of the upstream units,  $i, \dots, j-1$  (through on-line simulations of each unit’s response under the pre- and post-failure configurations) and then transmitted, through the communication network, to unit  $j$  which in turn uses these bounds to construct, on-line, both the controller and the fault-recovery region (see the subsection on communication logic for a discussion on how the resulting computational delays can be handled).

### 2.3.3 Supervisory Switching Logic Design

Having designed the robust feedback control law and characterized the fault-recovery region associated with each fall-back configuration, the third step in our design methodology is to derive the switching policy that the local supervisor of the downstream unit,  $j$ , needs to follow in reconfiguring the local control system (i.e., activating/deactivating the appropriate fall-back configurations) in the event of the upstream failure. In the general case, when more than one fall-back control configuration is available for the unit under consideration, the question is how to decide which of these configurations can and should be activated at the time of failure in order to preserve closed-loop stability. The key idea here is that, because of the

limitations imposed by constraints on the fault-recovery region of each configuration, the local supervisor can only activate the configuration whose fault-recovery region contains the closed-loop state at the time of the failure. Without loss of generality, let the active control configuration in the  $j$ -th unit, priori to the occurrence of failure in unit  $i$ , be  $k_j(T_f^-) = \mu$  for some  $\mu \in \mathcal{K}_j$ , where  $k_j(T_f^-) = \lim_{t \rightarrow T_f^-} k_j(t)$  and  $T_f$  is the time that the control system of unit  $i$  fails, then the switching rule given by

$$k_j(T_f^+) = \nu \text{ if } x_j(T_f) \in \Omega_j^\nu(u_{j,max}^\nu, \theta_b(T_f)) \quad (2.12)$$

for some  $\nu \in \mathcal{K}_j$ ,  $\nu \neq \mu$ , guarantees that the closed-loop system of the  $j$ -th unit is stable. The implementation of the above switching law requires monitoring, by the local supervisor, of the evolution of the closed-loop state trajectory with respect to the fault-recovery regions associated with the various control actuator configurations. Another way to look at the above switching logic is that it implicitly determines, for a fixed fall-back configuration, the times that the control system of the  $j$ -th unit can tolerate upstream failures by switching to this configuration. If failure occurs at times when  $x_j$  lies outside the fault-recovery region of all available configurations, this analysis suggests that either the constraints should be relaxed – to enlarge the fault-recovery region of the given configurations – or additional fall-back control loops must be introduced. The second option, however, is ultimately limited by the maximum allowable number of control loops that can be designed for the given processing unit. If neither option is feasible, a shutdown could be unavoidable. The proposition of constructing the switching logic on the basis of the stability regions was first proposed in [47] for the control of switched nonlinear systems.



### 2.3.4 Design of Communication Logic

Given the distributed interconnected nature of the plant units – and thus the potential for failure effects propagating from one unit to another – an essential element in the design of the fault-tolerant control system is the use of a communication medium that ensures fast and efficient transmission of information during failure events. As discussed earlier, communication networks offer such a medium that is both fast (relative to the typically slow dynamics of chemical processes) and inexpensive (relative to current point-to-point connection schemes which require extensive cabling and higher maintenance time and costs). The ability of the network to fulfill this role, however, requires that we devise the communication policy in a way that respects the inherent limitations in network resources, such as bandwidth constraints and overall delays, by minimizing unnecessary usage of the network.

In the section on feedback controller synthesis, we have already discussed how the bandwidth constraint issue can be handled by formulating the problem as a robust control problem, where the failure in the control system of the  $i$ -th processing unit and the consequent effects on units  $i + 1, \dots, j - 1$  are treated as a bounded non-vanishing disturbances that affect unit  $j$  downstream. The communication policy requires that the local supervisors of units  $i, \dots, j - 1$  perform the following tasks: (1) compute the disturbance bounds using the process model of each unit, and (2) send this information, together with other relevant information such as the failure type, the failure time and operating conditions, to the plant supervisor. The plant supervisor in turn forwards the information to the local supervisor of unit  $j$  utilizing the plant-wide communication network (see Figure 1.4). This policy avoids unnecessary overloading of the network (which could result when measurements from the upstream units are sent continuously to unit  $j$ ) while also guaranteeing fault-tolerance in the downstream

units. The idea of using knowledge of the plant dynamics to balance the tradeoff between bandwidth limitations (which favor reduced communication of measurements) and optimum control performance (which favors increased communication of measurements) is conceptually aligned with the notion of minimum attention control (for example, see [22, 124]). In our work, however, this idea is utilized in the context of fault-tolerant control.

The second consideration in devising the communication logic is the issue of time-delays which typically result from the time sharing of the communication medium as well as the computing time required for the physical signal coding and communication processing. The characteristics of these time-delays depend on the network protocols adopted as well as the hardware chosen. For our purposes here, we consider an overall fixed time-delay (which we denote by  $\tau_{max}^j$ ) that combines the contribution of several delays, including: (1) delays in fault-detection, (2) the time that the local supervisors of units  $i, \dots, j - 1$  take to compute the effective disturbance bounds (through simulations comparing the pre- and post-failure state evolutions in each unit), (3) the time that the local supervisors of units  $i, \dots, j - 1$  take to send the information to the plant supervisor, (4) the time that it takes the plant supervisor to forward the information to the local supervisor of unit  $j$ , (5) the time that it takes the local supervisor for unit  $j$  to compute the fault-recovery region for the given fall-back configurations using the information arriving from the upstream units and the time that it takes for the supervisor's decision to reach and activate the appropriate fall-back configuration, and (6) the inherent actuator/sensor dead-times.

Failure to take such delays into account can result in activating the wrong control configuration and subsequent instability. For example, even though the upstream failure may take place at  $t = T_f$ , the fall-back configuration in the control system of

unit  $j$  will not be switched in before  $t = T_f + \tau_{max}^j$ . If the delay is significant, then the switching rule in the previous section should be modified such that the local supervisor for unit  $j$  activates configuration,  $k_j = \nu$ , for which  $x_j(T_f + \tau_{max}^j) \in \Omega_j^\nu(u_{j,max}^\nu, \theta_b)$ . This modification is yet another manifestation of the inherent coupling between the switching and communication logics. The implementation of the modified switching rule that accounts for delays requires that the local supervisor of unit  $j$  be able to predict where the state trajectory will be at  $t = T_f + \tau_{max}^j$  (for example, through simulations using the process model) and check whether the state at this time is within the fault-recovery region of a given fall-back configuration. If not, then either an alternative fall-back configuration, for which the fault-recovery region contains the state at the end of the delay, should be activated or a shutdown maybe unavoidable. The availability of several fall-back control loops, however, is limited by process design considerations which dictate, for example, how many variables can be used for control. Figure 2.2 summarizes the overall fault-tolerant control strategy for a two-unit plant.

## 2.4 Simulation Studies

In this section, we present two simulation studies that demonstrate the application of the proposed fault-tolerant control system design methodology to two chemical processes. In the first application, a single chemical reactor example is considered to demonstrate the idea of re-configuring the local control system in the event of failures on the basis of the stability regions of the constituent control configurations, and how overall communication delays impact the re-configuration logic. In the second application, a cascade of two chemical reactors in series is considered to demonstrate how the issue of failure propagation between a multi-unit plant is handled within the proposed methodology, and how the various interplays between the feedback,

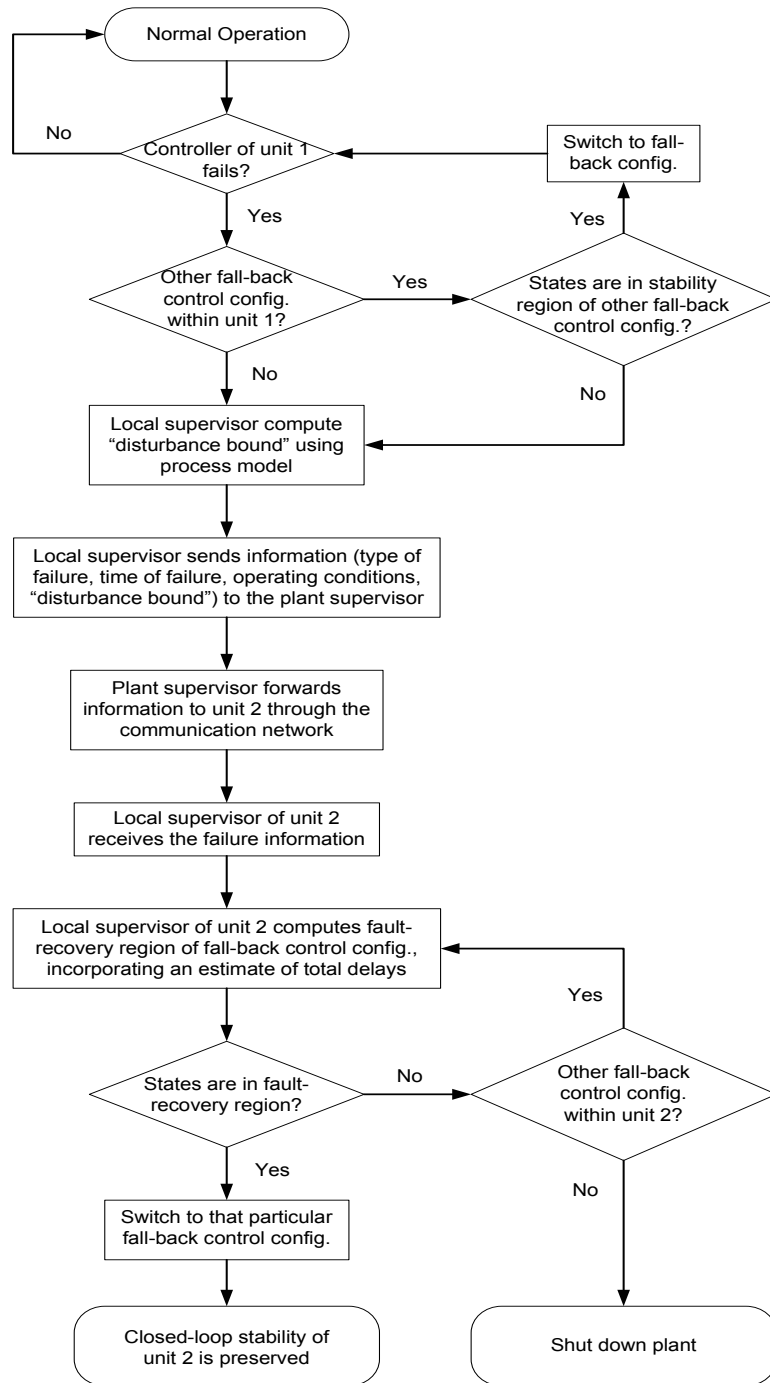


Figure 2.2: Summary of the fault-tolerant control strategy, for a two-unit plant, using communication networks.

supervisory control and communication tasks are handled in the multi-unit setting.

### 2.4.1 Application to a Single Chemical Reactor

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. The feed to the reactor consists of pure  $A$  at flow rate  $F$ , molar concentration  $C_{A0}$  and temperature  $T_{A0}$ . Due to the non-isothermal nature of the reactions, a jacket is used to remove/provide heat to the reactor. Under standard modeling assumptions, a mathematical model of the process can be derived from material and energy balances and takes the following form:

$$\begin{aligned}\frac{dT}{dt} &= \frac{F}{V}(T_{A0} - T) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_A, T) + \frac{Q}{\rho c_p V} \\ \frac{dC_A}{dt} &= \frac{F}{V}(C_{A0} - C_A) - \sum_{i=1}^3 R_i(C_A, T) \\ \frac{dC_B}{dt} &= -\frac{F}{V}C_B + R_1(C_A, T)\end{aligned}\tag{2.13}$$

where  $R_i(C_A, T) = k_{i0} \exp\left(\frac{-E_i}{RT}\right) C_A$ ,  $C_A$  and  $C_B$  denote the concentrations of the species  $A$  and  $B$ , respectively,  $T$  denotes the temperature of the reactor,  $Q$  denotes the rate of heat input to the reactor,  $V$  denotes the volume of the reactor,  $\Delta H_i$ ,  $k_i$ ,  $E_i$ ,  $i = 1, 2, 3$ , denote the enthalpies, pre-exponential constants and activation energies of the three reactions, respectively,  $c_p$  and  $\rho$  denote the heat capacity and density of fluid in the reactor. The values of the process parameters and the corresponding steady-state values are given in Table 2.2. It was verified that under these conditions, the process model of Equation 2.13 has three steady-states: two locally asymptotically stable and one unstable at  $(T^s, C_A^s, C_B^s) = (388 \text{ K}, 3.59 \text{ kmol/m}^3, 0.41 \text{ kmol/m}^3)$ .

The control objective is to stabilize the reactor at the (open-loop) unstable steady-

Table 2.2: Process parameters and steady-state values for the chemical reactor of Equation 2.13.

$F$	=	4.998	$m^3/hr$
$V$	=	1.0	$m^3$
$R$	=	8.314	$KJ/kmol \cdot K$
$T_{A0}$	=	300.0	$K$
$C_{A0}$	=	4.0	$kmol/m^3$
$C_{B0}$	=	0.0	$kmol/m^3$
$\Delta H_1$	=	$-5.0 \times 10^4$	$KJ/kmol$
$\Delta H_2$	=	$-5.2 \times 10^4$	$KJ/kmol$
$\Delta H_3$	=	$-5.4 \times 10^4$	$KJ/kmol$
$k_{10}$	=	$3.0 \times 10^6$	$hr^{-1}$
$k_{20}$	=	$3.0 \times 10^5$	$hr^{-1}$
$k_{30}$	=	$3.0 \times 10^5$	$hr^{-1}$
$E_1$	=	$5.0 \times 10^4$	$KJ/kmol$
$E_2$	=	$7.53 \times 10^4$	$KJ/kmol$
$E_3$	=	$7.53 \times 10^4$	$KJ/kmol$
$\rho$	=	1000.0	$kg/m^3$
$c_p$	=	0.231	$KJ/kg \cdot K$
$T^s$	=	388.57	$K$
$C_A^s$	=	3.59	$kmol/m^3$
$C_B^s$	=	0.41	$kmol/m^3$

state. Operation at this point is typically sought to avoid high temperatures while, simultaneously, achieving reasonable reactant conversion. To accomplish this objective in the presence of control system failures, we consider the following manipulated input candidates:

1. Rate of heat input,  $u_1 = Q$ , subject to the constraint  $|Q| \leq u_{max}^1 = 748 \text{ KJ/s}$ .
2. Inlet stream temperature,  $u_2 = T_{A0} - T_{A0}^s$ , subject to the constraint  $|u_2| \leq u_{max}^2 = 100 \text{ K}$ .
3. Inlet reactant concentration,  $u_3 = C_{A0} - C_{A0}^s$ , subject to the constraint  $|u_3| \leq u_{max}^3 = 4 \text{ kmol/m}^3$ .

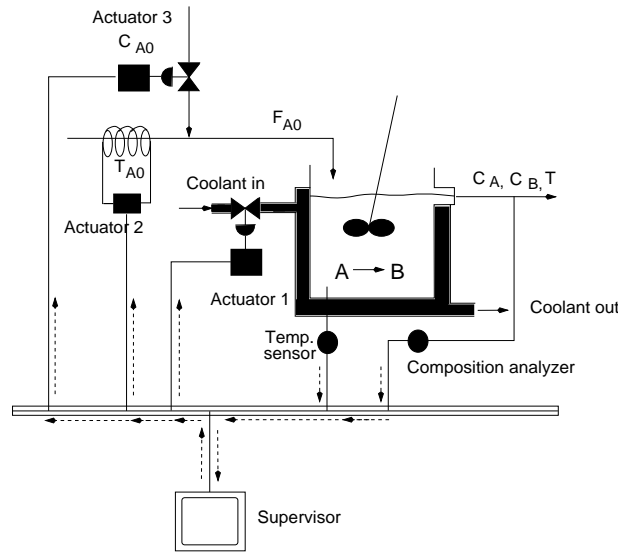


Figure 2.3: Fault-tolerant control structure for a single unit operation, integrating supervisory and feedback control over a communication network.

Each of the above manipulated inputs represents a unique control configuration (or control-loop) that, by itself, can stabilize the reactor using available measurements of the reactor temperature, reactant and product concentrations provided by the sensors.

The sensors and control actuators of each configuration are connected to the unit supervisor (for example, a distant control room) over a communication network (see Figure 2.3). The first loop involving the heat input,  $Q$ , as the manipulated variable will be considered as the primary control configuration. In the event of a total failure in this configuration, however, the supervisor will have to activate one of the other two fall-back configurations in order to maintain closed-loop stability. The main question that we address in this simulation study is how can the supervisor determine which control loop to activate once failure is detected in the active configuration and how overall communication delays influence this decision.

Following the proposed methodology, we initially synthesize, for each control configuration, a feedback controller that enforces asymptotic closed-loop stability in the presence of actuator constraints. This task is carried out on the basis of the process input/output dynamics. While our control objective is to achieve full-state stabilization, auxiliary process outputs are introduced here to facilitate transforming the system of Equation 2.13 into a form more suitable for explicit controller synthesis. In the case of the process of Equation 2.13, a further simplification can be obtained by noting that  $C_B$  does not affect the evolution of either  $T$  or  $C_A$  and, therefore, the controller design can be addressed on the basis of the  $T$  and  $C_A$  equations only. A controller that stabilizes the  $(T, C_A)$  subsystem also stabilizes the entire closed-loop system. For the first configuration with  $u_1 = Q$ , we consider the output  $y_1 = (C_A - C_A^s)/C_A^s$ . This choice yields a relative degree of  $r_1 = 2$  for the output with respect to the manipulated input. The coordinate transformation (in error variables form) takes the form:  $e_1 = (C_A - C_A^s)/C_A^s$ ,  $e_2 = (F/V)(C_{A0} - C_A)/C_A^s - \sum_{i=1}^3 k_{i0} \exp\left(\frac{-E_i}{RT}\right) C_A/C_A^s$ . For the second configuration with  $u_2 = T_{A0} - T_{A0}^s$ , we choose the output  $y_2 = (C_A - C_A^s)/C_A^s$  which yields the same relative degree as in the first configuration,



$r_2 = 2$ , and the same coordinate transformation. For the third configuration, with  $u_3 = C_{A0} - C_{A0}^s$ , we choose the output  $y_3 = (T - T^s)/T^s$  which yields a relative degree of  $r_3 = 2$  and a coordinate transformation of the form:  $e_1 = (T - T^s)/T^s$ ,  $e_2 = (F/V)(T_{A0} - T)/T^s + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p T^s} R_i(C_A, T) + \frac{Q}{\rho c_p V T^s}$ .

Note that since our objective is full-state stabilization, the choice of the output in each case is really arbitrary. However, to facilitate the controller design and subsequent stability analysis, we have chosen in each case an output that produces a system of relative degree 2. For each configuration, the corresponding state transformation yields a system, describing the input/output dynamics, of the following form:

$$\begin{aligned} \dot{e} &= Ae + l_k(e) + b\alpha_k u_k \\ &:= \bar{f}_k(e) + \bar{g}_k(e)u_k, \quad k = 1, 2, 3 \end{aligned} \quad (2.14)$$

where  $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ ,  $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ,  $l_k(\cdot) = L_{f_k}^2 h_k(x)$ ,  $\alpha_k(\cdot) = L_{g_k} L_{f_k} h_k(x)$ ,  $h_k(x) = y_k$  is the output associated with the  $k$ -th configuration,  $x = [x_1 \ x_2]^T$  with  $x_1 = (T - T^s)/T^s$ ,  $x_2 = (C_A - C_A^s)/C_A^s$ , and the functions  $f_k(\cdot)$  and  $g_k(\cdot)$  can be obtained by re-writing the  $(T, C_A)$  model equations in Equation 2.13 in the form of Equation 2.1. The explicit forms of these functions are omitted for brevity. Using 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  $A^T P_k + P_k A - P_k b b^T P_k < 0$ , we synthesize, for each control loop, a bounded nonlinear feedback control law of the form of Equations 2.3-2.4 and characterize the associated stability region with the aid of Equation 2.5. Figure 2.4 depicts the stability region, in the  $(T, C_A)$  space, for each configuration. The stability region of configuration 1 includes the entire area of the plot. The stability region of configuration 2 is the entire area to the left of the solid line, while the stability region of configuration 3 covers the area to the right of the dashed vertical line. The desired steady-state is depicted with an asterisk that lies in the intersection of the three stability regions.

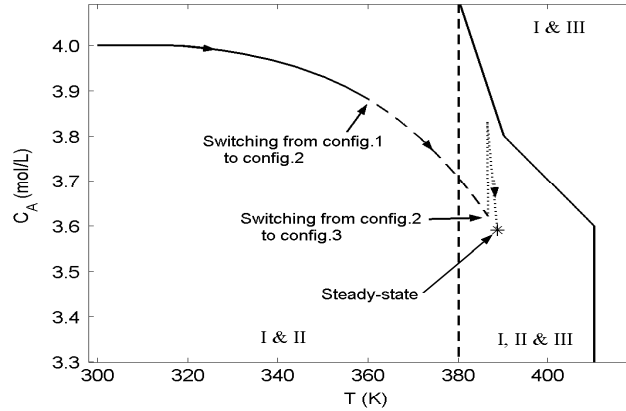


Figure 2.4: Stability regions of the three control configurations (I, II, III) considered for the chemical reactor example of Equation 2.13.

We consider first the case when no time-delays are involved and the supervisor can switch immediately between the different control loops in the event of failures. To this end, the reactor is initialized at  $T(0) = 300\text{ K}$ ,  $C_A(0) = 4.0\text{ kmol/m}^3$ ,  $C_B(0) = 0.0\text{ kmol/m}^3$ , using the  $Q$ -control configuration, and the supervisor proceeds to monitor the evolution of the closed-loop trajectory. As shown by the solid parts of the closed-loop trajectory in Figure 2.4, the state profiles in Figure 2.5 and the rate of heat input profile in Figure 2.6, the controller proceeds to drive the closed-loop trajectory towards the desired steady-state until the control actuators  $Q$ -configuration experiences a total failure after 2.0 hrs of startup (simulated by fixing  $Q = 0$  for all  $t \geq 2.0\text{ hr}$ ). From the solid part of the trajectory in Figure 2.4, it is clear that the failure of the primary control configuration 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, the supervisor immediately activates the second configuration, with  $T_{A0}$  as the manipulated input. The result is shown by the dashed parts of the closed-loop trajectory in Figure 2.4, the state profiles in Figure 2.5 and the inlet stream temperature

profile in Figure 2.6 where it is seen that, upon switching to the  $T_{A0}$ -configuration, the corresponding controller continues to drive the closed-loop trajectory closer to the desired steady-state. At  $t = 15.0$  hr, we consider another total failure in the control actuators of the  $T_{A0}$ -configuration (simulated by fixing  $T_{A0}$  for all  $t \geq 15.0$  hr). From the dashed part of the trajectory in Figure 2.4, it is clear that this failure 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, which then successfully stabilizes the reactor at the desired steady-state (see the dotted parts of the closed-loop trajectory in Figure 2.4, the state profiles in Figure 2.5 and the inlet reactant concentration in Figure 2.6).

To demonstrate the effect of delays on the implementation of the switching logic, we consider an overall delay, between the supervisor and the constituent control configurations, of  $\tau_{max} = 8.0$  min (accounting for possible delays in fault-detection, control computations, network transmission and actuator activation). In this case, the reactor is initialized at  $T(0) = 300$  K,  $C_A(0) = 4.0$  kmol/m<sup>3</sup>,  $C_B(0) = 0$  kmol/m<sup>3</sup> under the first control configuration (with  $Q$  as the manipulated input). The actual failure of this configuration occurs at  $t = 10$  hr which, as can be seen from Figure 2.7, is a time when the closed-loop state trajectory is within the intersection of all three stability regions. In the absence of delays, this suggests that switching to either configuration 2 or 3 should preserve closed-loop stability. We observe, however, from Figure 2.8 that, when the delay is present, activation of configuration 3 leads to instability (dotted profile) while activation of configuration 2 achieves stabilization at the desired steady-state (dashed profiles). The reason is the fact that, for the time period between the actual failure ( $t = 10$  hr) and the activation of the backup configuration ( $t = 10.13$  hr), the process evolves in an open-loop fashion leading the

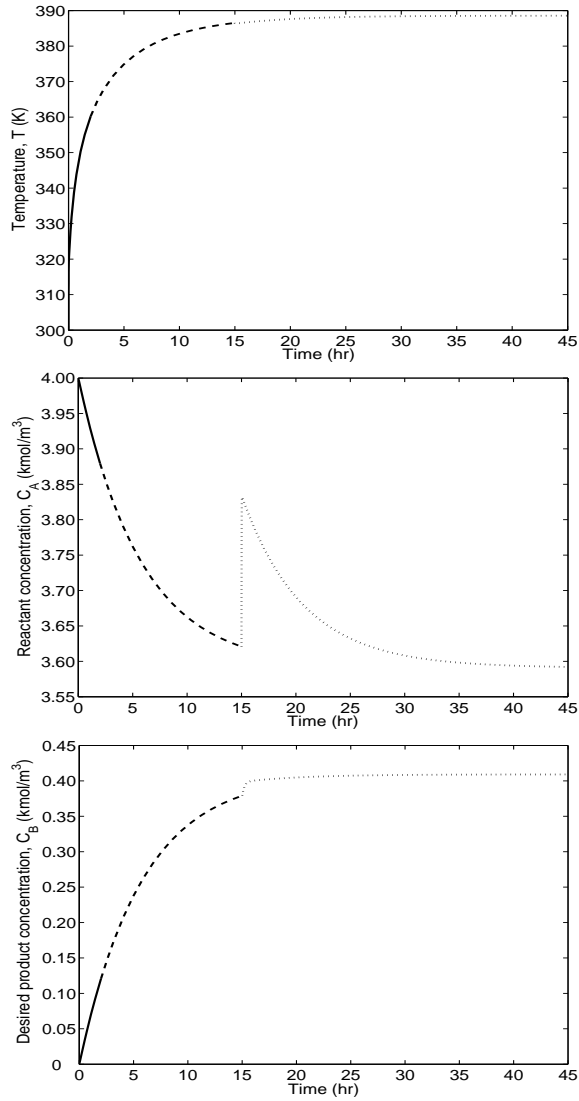


Figure 2.5: Evolution of the closed-loop state profiles under repeated control system failures and subsequent switching by the supervisor from configuration 1 (solid lines) to configuration 2 (dashed lines) to configuration 3 (dotted lines).

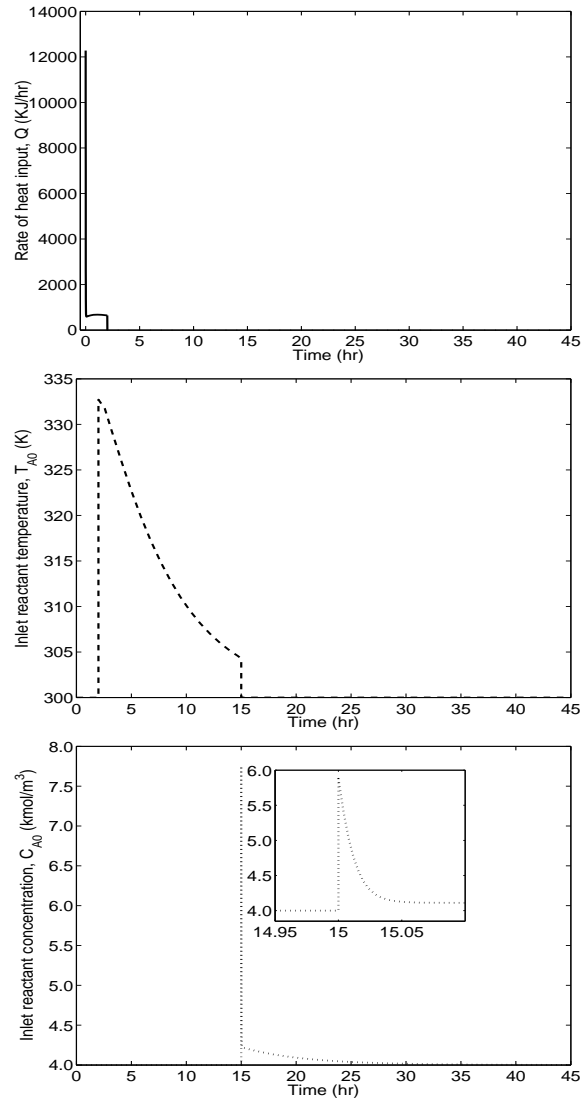


Figure 2.6: Manipulated input profiles for each control configuration as the supervisor switches from configuration 1 to configuration 2 at  $t = 2$  hr and from configuration 2 to configuration 3 at  $t = 15$  hr.

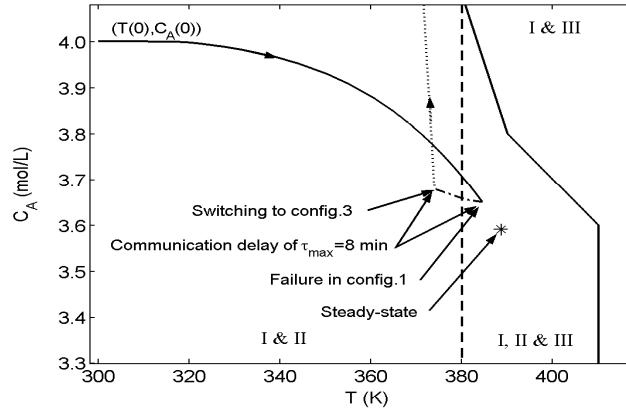


Figure 2.7: A phase plot showing the closed-loop state trajectory leaving the intersection zone (I, II, & III) during the delay period (dashed-dotted trajectory) rendering configuration 3 destabilizing (dotted trajectory).

trajectory to move out of the intersection zone such that at  $t = 10.13$  hr the state is within the stability region of configuration 2 and outside that of configuration 3. This is shown in Figure 2.7. The corresponding manipulated input profiles are shown in Figure 2.9. To activate the correct configuration in this case, the supervisor needs to predict where the state trajectory will be at the end of the communication delay period.

#### 2.4.2 Application to Two Chemical Reactors in Series

In this section, we revisit the two chemical reactors in series of Equation 2.2, introduced earlier in the motivating example section, to illustrate the implementation of the proposed fault-tolerant control methodology. To this end, the reactors are initialized at  $(T_1(0), C_{A1}(0)) = (300 \text{ K}, 4.0 \text{ kmol}/\text{m}^3)$ , and  $(T_2(0), C_{A2}(0)) = (440 \text{ K}, 4.0 \text{ kmol}/\text{m}^3)$ . Under normal operating conditions (with no failures), each reactor is controlled by manipulating the rate of heat input, using a bounded nonlinear control law of the form of Equations 2.3-2.4.

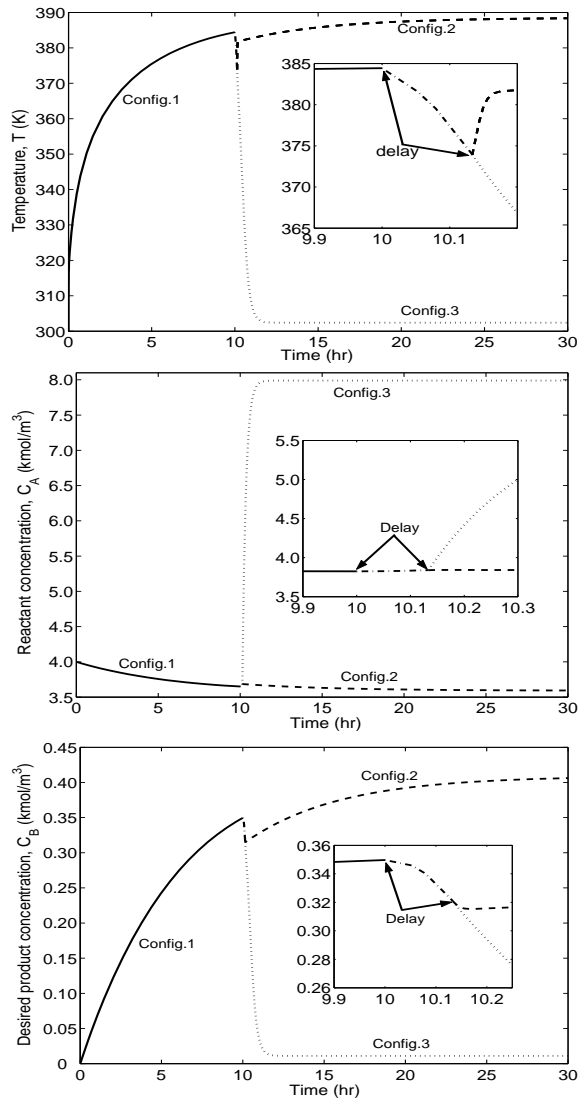


Figure 2.8: Evolution of the closed-loop state profiles when configuration 1 (solid lines) fails at  $t = 10$  hr and an overall delay of  $\tau_{max} = 8.0$  min elapses before the backup configuration is activated. Activation of configuration 2 preserves closed-loop stability (dashed lines) while activation of configuration 3 leads to instability (dotted lines).

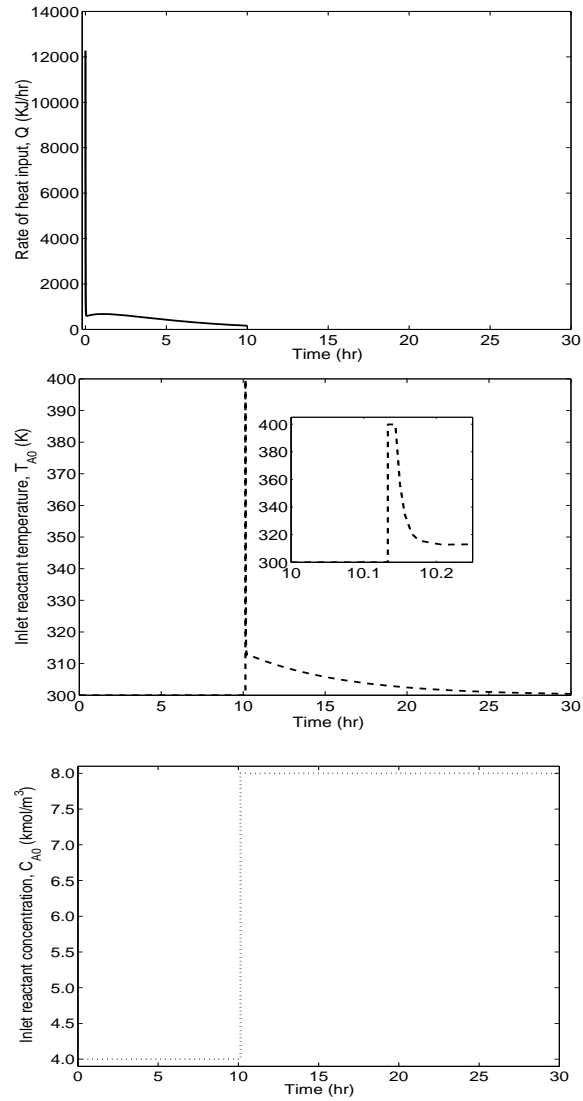


Figure 2.9: Manipulated input profiles when configuration 1 fails at  $t = 10$  hr and an overall delay of  $\tau_{max} = 8.0$  min elapses before the backup configuration is activated.



For the first CSTR, the controller design procedure is the same as the one used for the  $Q$ -configuration in the previous simulation example. For the second CSTR, we design the controller on the basis of the temperature equation only. Specifically, a quadratic function of the form  $V_2 = \frac{1}{2}a_2(x_2^{(1)})^2$ , where  $x_2^{(1)} = (T_2 - T_2^s)/T_2^s$ , is used to design the controller and estimate the resulting stability region using Equation 2.5. The values of the controller tuning parameters are chosen to be  $a_2 = 0.5$  and  $\rho_2 = 0.0001$ . Figure 2.10 (solid profiles) and Figure 2.11 show the resulting closed-loop state and manipulated input profiles when the controllers are implemented without failure for both reactors. We observe that each controller successfully stabilizes the corresponding reactor at the desired steady-state.

Consider now a total failure in the actuators of both control systems ( $Q_1$  and  $Q_2$ ) at  $T_f = 5$  min. In this case, both reactors revert to their open-loop mode of behavior and, consequently, if no fall-back control configuration is activated, the states move away from the desired steady-state, as shown by the dashed lines in Figure 2.10 for the first reactor, and Figure 2.12 for the second reactor (note that  $C_{A03}$  remains fixed for all times since it is not used as a manipulated variable in the pre-failure configuration). As stated in the motivation example subsection, we assume that the controller failure in the first reactor is permanent; and our objective is to prevent the propagation of this effect to the second reactor. A permanent failure in the first unit could be the result of lack of sufficient fall-back configurations or because failure occurs at a time when the state is outside the stability regions of all the available configurations for this unit.

Using the proposed methodology, the supervisor of CSTR 1, at the failure time, runs both open-loop and closed-loop simulations using the process model of CSTR 1 to estimate the size of the disturbance affecting CSTR 2, and transmits this infor-

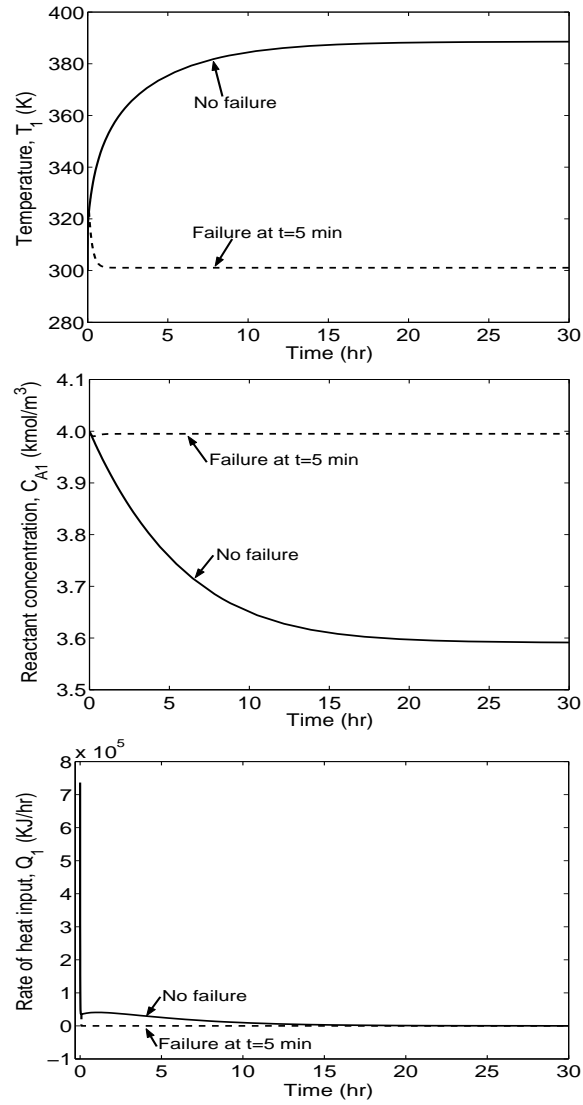


Figure 2.10: Evolution of the closed-loop state and manipulated input profiles for CSTR 1 under a well-functioning control system (solid lines) and when the control actuator fail at  $t = 5$  min (dashed lines).

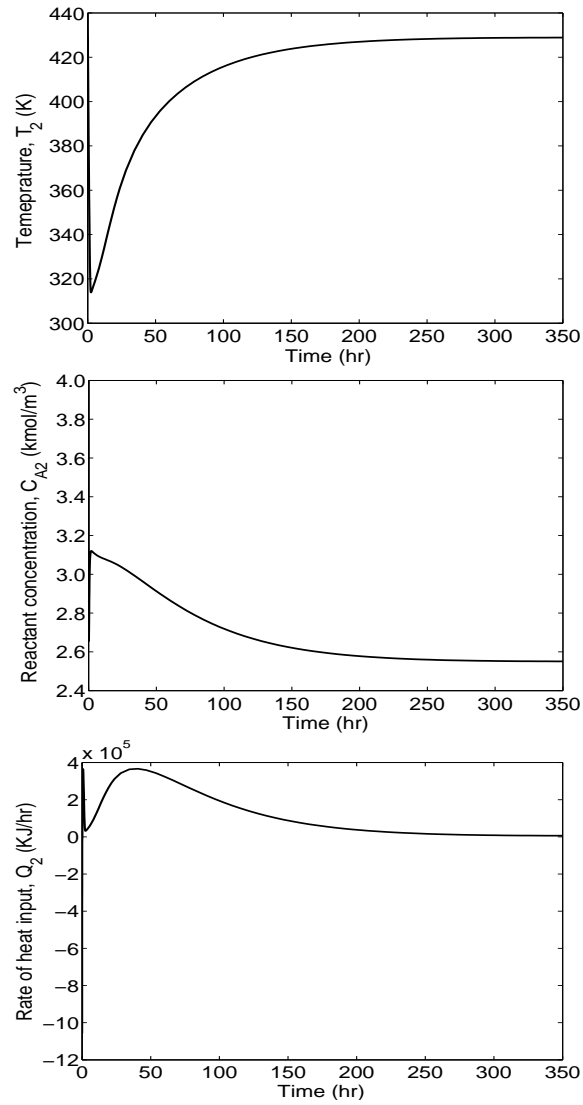


Figure 2.11: Evolution of the closed-loop state and manipulated input profiles for CSTR 2 under a well-functioning control system.

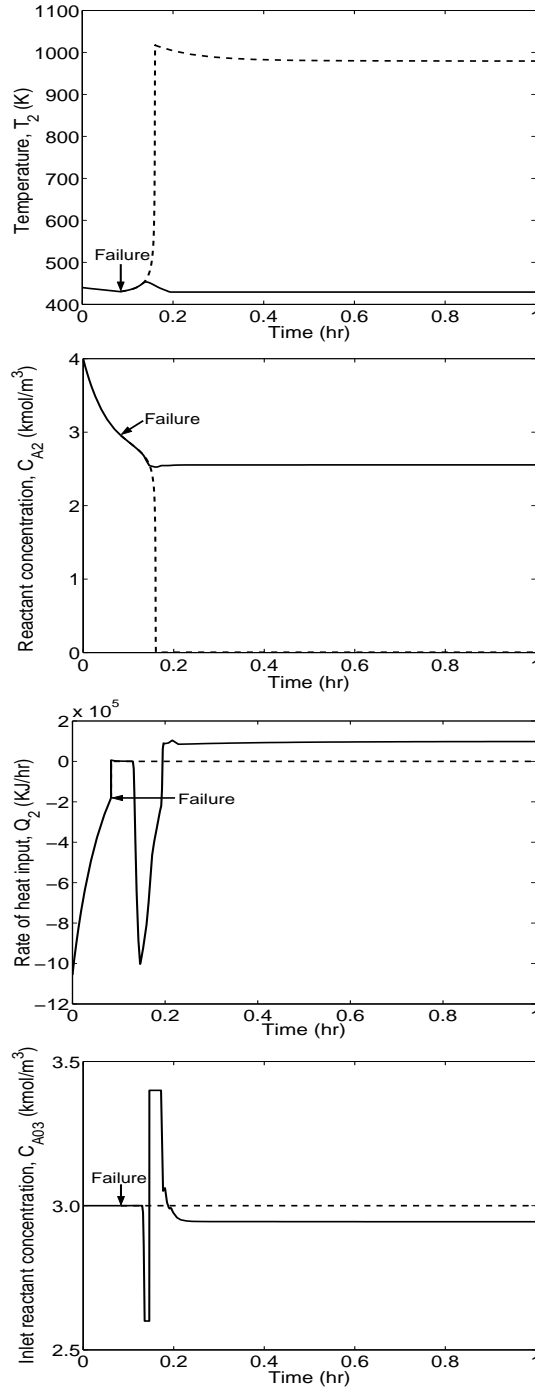


Figure 2.12: Evolution of the closed-loop state and manipulated input profiles for CSTR 2 when the controller of the fall-back configuration ( $Q_2, C_{A03}$ ) is activated immediately after the failure (solid lines), and the open-loop state and input profiles resulting when the fall-back configuration is not activated after the failure (dashed lines).

mation to the local supervisor of CSTR 2 through the communication network. The maximum disturbance size is proportional to the largest discrepancy (after the failure time) between the values of  $C_{A1}$ ,  $T_1$  in the well-functioning (solid lines in Figure 2.10) and in the failed (dashed lines in Figure 2.10) modes. Using this information, the local supervisor of CSTR 2 designs a robust control law of the form of Equations 2.7-2.10 to stabilize CSTR 2, using the available fall-back configuration with  $(Q_2, C_{A03})$  as the manipulated inputs, and constructs the associated fault-recovery region for this configuration. The controller design procedure involves re-writing the process model of CSTR 2 in Equation 2.2 in the form of Equation 2.6, using the dimensionless variables,  $x_i^{(1)} = (T_i - T_i^s)/T_i^s$ ,  $x_i^{(2)} = (C_{Ai} - C_{Ai}^s)/C_{Ai}^s$ ,  $i = 1, 2$ , and with the states of CSTR 1 re-defined as the disturbance variables,  $\theta_1(t) = [\theta_1^{(1)}(t) \theta_1^{(2)}(t)]^T$ , where  $\theta_1^{(1)}(t) = (F_1 T_1^s / V_2 T_2^s)(x_1^{(1)}(t) + 1)$  and  $\theta_1^{(2)}(t) = (F_1 C_{A1}^s / V_2 C_{A2}^s)(x_1^{(2)}(t) + 1)$ , for all  $t \geq T_f$ . Then, using a quadratic function of the form  $V_2 = \frac{1}{2}a_2(x_2^{(1)})^2 + \frac{1}{2}a_2(x_2^{(2)})^2$ , the controller of Equations 2.7-2.10 is constructed and its fault-recovery region is computed with the aid of Equation 2.11. The disturbance bound is computed as  $\theta_b^1 = \sup_{t \geq T_f} \|\theta_1(t)\|$ . The values of the controller tuning parameters are selected to be  $a_2 = 0.5$ ,  $\rho_2 = 0.0001$ ,  $\chi_2 = 2.0001$  and  $\phi_2 = 0.0001$ . The fault-recovery region is depicted by the shaded area in Figure 2.13.

From Figure 2.13, we observe that the failure occurs when the states of CSTR 2 are within the fault-recovery region. Therefore, assuming no delays in the fault-detection, computations and communication processing (i.e., instantaneous switching), when the fall-back controllers are activated, closed-loop stability is preserved and the closed-loop states converge close to the desired steady-state as shown by the solid lines in Figure 2.12.

When delay effects are taken into account, we see from Figure 2.13 (top plot)

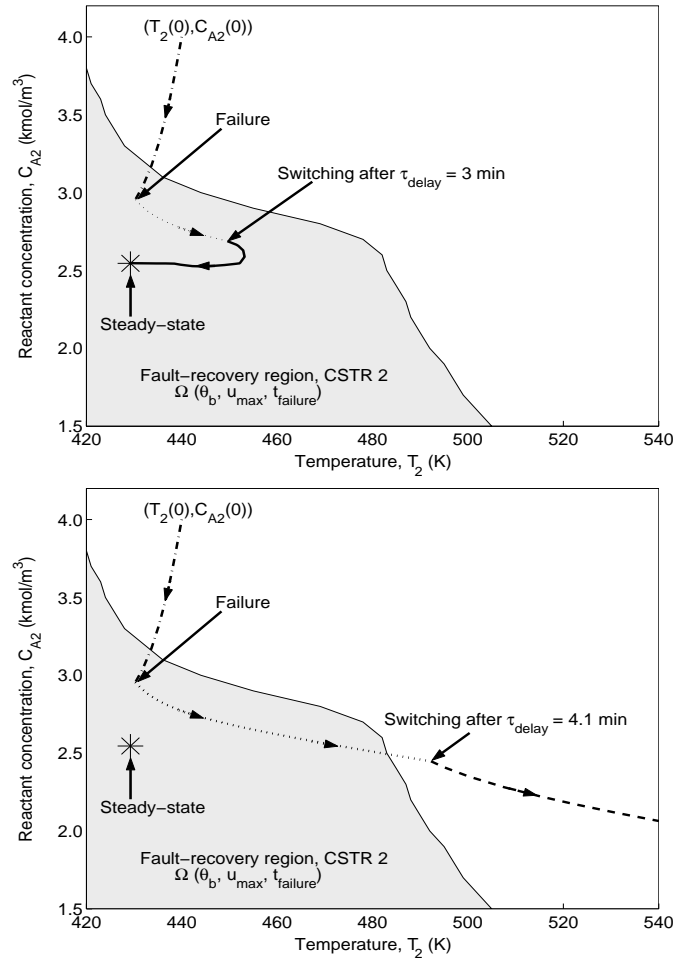


Figure 2.13: Fault-recovery region of the fall-back control configuration ( $Q_2, C_{A03}$ ) for CSTR 2, with constraints  $|Q_2| \leq 2.8 \times 10^6 \text{ KJ/hr}$  and  $|C_{A03} - C_{A03}^s| \leq 0.4 \text{ kmol/m}^3$  when failure occurs at  $T_f = 5$  min. Activation of the fall-back configuration after a 3 min delay preserves closed-loop stability (top plot), while activation after 4.1 min delay fails to ensure fault-tolerance (bottom plot).

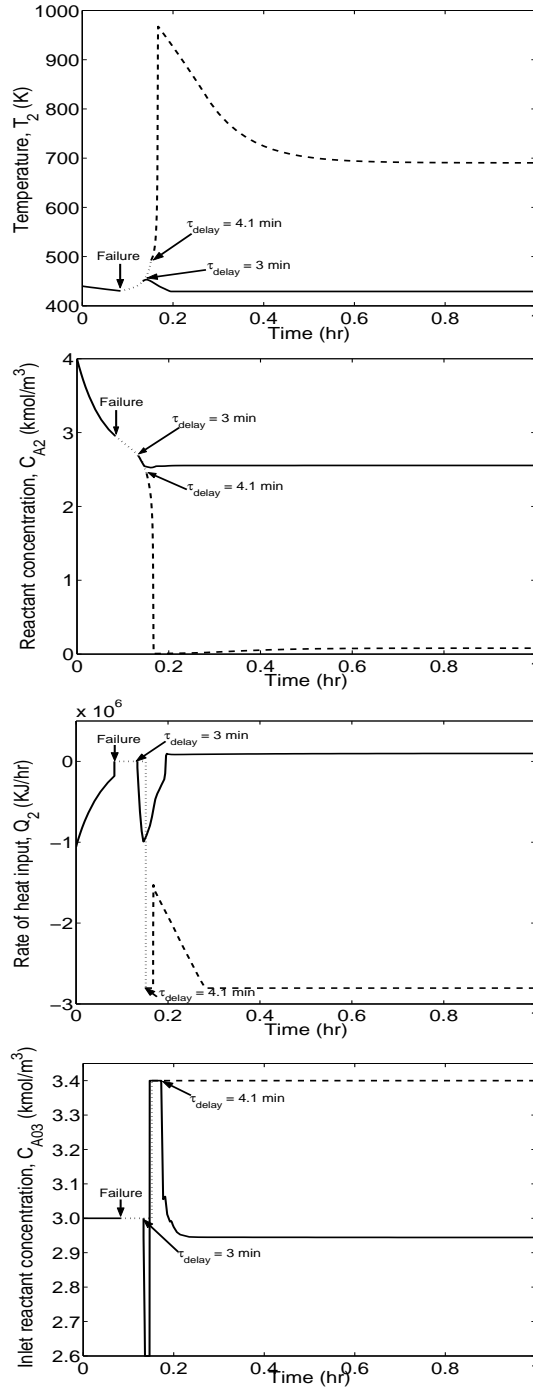


Figure 2.14: Evolution of the closed-loop state and input profiles when the failure occurs at  $T_f = 5$  min and the fall-back configuration ( $Q_2$ ,  $C_{A03}$ ), with constraints  $|Q_2| \leq 2.8 \times 10^6$   $KJ/hr$  and  $|C_{A03} - C_{A03}^s| \leq 0.4$   $kmol/m^3$  is activated after a total delay of 3 min (solid lines) and after a total delay of 4.1 min (dashed lines).

that if an overall delay of 3 min (accounting for delays in fault-detection, controller computations, information transmission and actuator activation) elapses between the failure and the activation of the  $(Q_2, C_{A03})$  configuration – during this delay, CSTR 2 evolves in an open-loop mode as indicated by the dotted line – the state at the end of the delay still resides within the fault-recovery region and, therefore, closed-loop stability is preserved by switching to the  $(Q_2, C_{A0}^s)$  configuration at the end of the delay. The corresponding state and input profiles are shown by the solid lines in Figures 2.13-2.14. By contrast, we see from the bottom plot in Figure 2.13 that when an overall delay of 4.1 min is considered, the state at the end of the delay lies outside the fault-recovery region; hence the fall-back configuration cannot stabilize the system at the desired steady-state, as can be seen from the dashed lines in Figures 2.13-2.14.

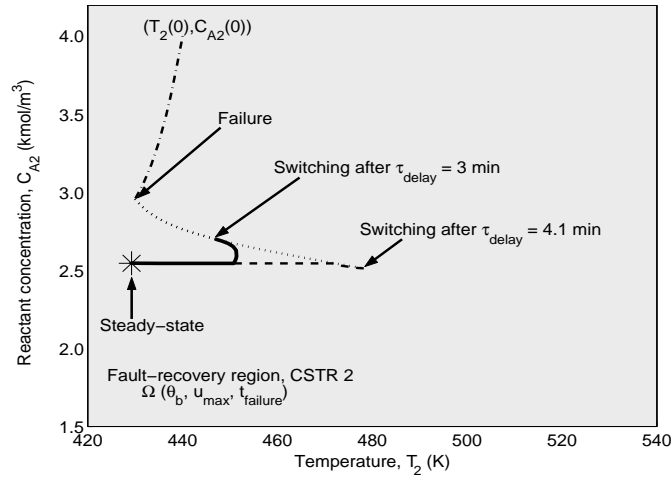


Figure 2.15: Fault-recovery region of the fall-back control configuration  $(Q_2, C_{A03})$  for CSTR 2, with constraints  $|Q_2| \leq 1.4 \times 10^7 \text{ KJ/hr}$  and  $|C_{A03} - C_{A03}^s| \leq 2.0 \text{ kmol/m}^3$  when failure occurs at  $T_f = 5 \text{ min}$ . Activation of the fall-back configuration after a delay of either 3 min or 4.1 min ensures fault-tolerance.

Examination of Figure 2.13 provides useful insights into how the tradeoff between



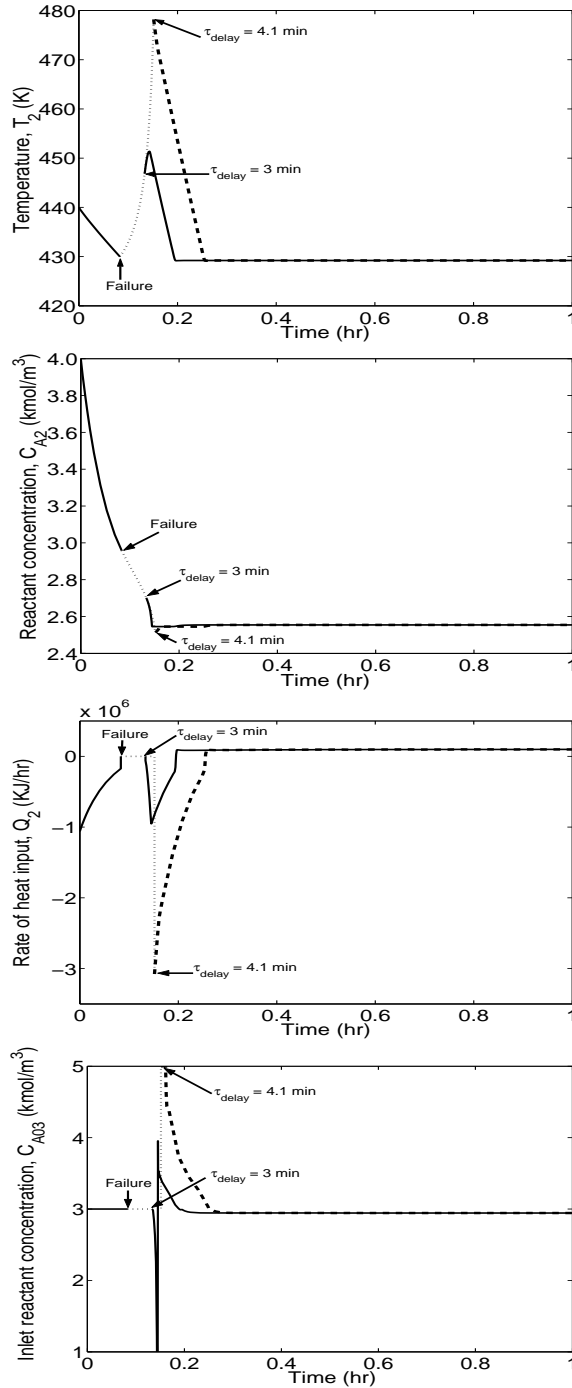


Figure 2.16: Evolution of the closed-loop state and manipulated input profiles when the failure occurs at  $T_f = 5$  min and the fall-back configuration ( $Q_2$ ,  $C_{A03}$ ), with constraints  $|Q_2| \leq 1.4 \times 10^7$  KJ/hr and  $|C_{A03} - C_{A03}^s| \leq 2.0$  kmol/m<sup>3</sup> is activated after a total delay of 3 min (solid lines) and after a total delay of 4.1 min. (dashed lines).

the controller design, switching and communication logics can be managed to ensure fault-tolerance. For example, the figure suggests that with a larger fault-recovery region, even large delays maybe tolerated by switching to this particular configuration. A larger region can be obtained by relaxing the constraints. Figure 2.15 shows the resulting fault-recovery region for the  $(Q_2, C_{A03})$  configuration when the constraints are relaxed to  $|Q_2| \leq u_{max}^{Q_2} = 1.4 \times 10^7 \text{ KJ/hr}$  and  $|C_{A03} - C_{A03s}| \leq u_{max}^{C_{A03}} = 2.0 \text{ kmol/m}^3$ . In this case, the fault-recovery region includes the entire area of the plot. As a result, activation of the fall-back configuration, whether after 3 min or 4.1 min from the failure time, stabilizes the reactor since the state at the end of the delay in both cases is contained within the fault-recovery region. Figure 2.16 shows the corresponding closed-loop state and input profiles of CSTR 2 for both scenarios.

## 2.5 Conclusions

In this chapter, we proposed a methodology for the design of fault-tolerant control systems for chemical plants with distributed interconnected processing units. Bringing together tools from Lyapunov-based nonlinear control and hybrid systems theory, the approach is based on a hierarchical architecture that integrates low-level feedback control of the individual units with high-level logic-based supervisory control over communication networks. The local control system for each unit consists of a family of control configurations for each of which a stabilizing feedback controller is designed and the stability region is explicitly characterized. The actuators and sensors of each configuration are connected, via a local communication network, to a local supervisor that orchestrates switching between the constituent configurations, on the basis of the stability regions, in the event of failures. The local supervisors communicate, through a plant-wide communication network, with a plant supervi-

sor responsible for monitoring the different units and coordinating their responses in a way that minimizes the propagation of failure effects. The communication logic is designed to ensure efficient transmission of information between units while also respecting the inherent limitations in network resources by minimizing unnecessary network usage and accounting explicitly for the effects of possible delays due to fault-detection, control computations, network communication and actuator activation. Explicit guidelines for managing the various interplays between the coupled tasks of feedback control, fault-tolerance and communication were provided. The efficacy of the proposed approach was demonstrated through chemical process examples.

## Chapter 3

# Fault-Tolerant Control of Nonlinear Processes: Performance-Based Reconfiguration and Robustness

### 3.1 Introduction

In Chapter 2, we presented a hybrid system approach to fault-tolerant control where upon occurrence of a fault, stability region-based reconfiguration is implemented to achieve fault-tolerant control. The reconfiguration in Chapter 2, however, does not incorporate performance or robustness considerations, which can lead to performance-loss or even instability for processes subject to uncertainty.

Motivated by these considerations, in this chapter, we consider the problem of control system/actuator failures in nonlinear processes subject to input constraints and present two approaches for fault-tolerant control that focus on incorporating performance and robustness considerations, respectively. In both approaches, first

a family of candidate control configurations, characterized by different manipulated inputs, is identified for the process under consideration, and then performance and robustness considerations are incorporated in the implementation of fault-tolerant control [118].

We first introduce the class of systems considered, present a motivating example, and review two control approaches for handling process nonlinearity, inputs and constraints. Next, we present performance-based reconfiguration where performance considerations are incorporated in the controller design and in the switching logic. Specifically, we design a Lyapunov-based predictive controller that enforces closed-loop stability from an explicitly characterized set of initial conditions. The switching logic uses stability considerations (evaluated via the presence of the state in the stability region of a control configuration) to ascertain the suitability of a candidate backup configuration and then performance considerations are again considered in choosing between the suitable backup configurations. We demonstrate the implementation of the method on the chemical process example.

To show robustness considerations, we consider the problem of implementing fault-tolerant control to nonlinear processes subject to input constraints and uncertainty. To this end, we first design a robust hybrid predictive controller for each candidate control configuration that guarantees stability from an explicitly characterized set of initial conditions, subject to uncertainty and constraints. A switching policy is then derived to orchestrate the activation/deactivation of the constituent control configurations. We demonstrate the implementation of the robust fault-tolerant controller on the chemical process example.

## 3.2 Preliminaries

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

$$\begin{aligned} \dot{x} &= f(x) + G_k(x)u_k + W_k(x)\theta_k(t), u_k \in \mathbf{U}_k, \theta_k \in \Theta_k \\ k(t) &\in \mathcal{K} = \{1, \dots, N\}, N < \infty \end{aligned} \quad (3.1)$$

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  $\mathbf{U}_k$  of  $\mathbb{R}^m$ , where  $\mathbf{U}_k = \{u \in \mathbb{R}^m : \|u\| \leq u_k^{max}\}$ ,  $\|\cdot\|$  is the Euclidean norm of a vector,  $u_k^{max} > 0$  is the magnitude of input constraints, and  $\theta_k(t) = [\theta_k^1(t) \cdots \theta_k^q(t)]^T \in \Theta_k \subset \mathbb{R}^q$  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  $G_k(x) = [g_k^1(x) \cdots g_k^m(x)]$  and  $W(x) = [w_k^1(x) \cdots w_k^q(x)]$ , where  $g_k^i(x) \in \mathbb{R}^n$ ,  $i = 1 \cdots m$ , and  $w_k^i(x) \in \mathbb{R}^n$ ,  $i = 1 \cdots q$ , are assumed to be sufficiently smooth on their domains of definition.  $k(t)$ , which takes values in the finite index set  $\mathcal{K}$ , represents a discrete state that indexes the vector field  $g_k(\cdot)$  as well as the manipulated input  $u_k(\cdot)$ . For each value that  $k$  assumes in  $\mathcal{K}$ , the process is controlled via a different manipulated input which defines a given control configuration. Switching between the available  $N$  control configurations is controlled by a higher-level supervisor, thus determining the temporal evolution of the discrete state,  $k(t)$ . The supervisor ensures that only one control configuration is active at any given time, and allows only a finite number of switches over any finite interval of time. 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,  $\Omega$ . Throughout the manuscript, we assume that for any

$u_k \in \mathbf{U}_k$  the solution of the system of Equation 3.1 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  $t$ .

### 3.2.1 Motivating Example

To illustrate how performance and robustness considerations are incorporated in the fault-tolerant control design, we use chemical reactor example introduced in Section 2.4.1. The values of the process parameters are given in Table 2.2. It was verified that under these conditions, the open-loop process of Equation 2.13 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 manipulated input variables available for use within a control configuration include (see Figure 2.3) rate of heat input,  $u_1 = Q$ , inlet stream temperature,  $u_2 = T_{A0} - T_{A0s} := \Delta T_{A0}$  and inlet reactant concentration,  $u_3 = C_{A0} - C_{A0s} := \Delta C_{A0}$ , subject to the constraints  $|Q| \leq u_1^{max} = 748 \text{ KJ/s}$ ,  $|u_2| \leq u_2^{max} = 100 \text{ K}$ , with  $T_{A0s} = 300 \text{ K}$  and  $|u_3| \leq u_3^{max} = 4 \text{ kmol/m}^3$ , with  $C_{A0s} = 4 \text{ kmol/m}^3$ , respectively.

The first loop involving the heat input,  $Q$ , will be considered as the primary configuration. In the event of some failure in this configuration, however, the plant supervisor will have to activate one of the other two backup configurations in order to maintain closed-loop stability. Note, however, that the presence of constraints on the manipulated inputs limits the set of initial conditions starting from where the process states can be driven to a given (open-loop unstable) equilibrium point. Given that the primary control configuration fails, it is important to pick the appropriate backup control configuration that preserves closed-loop stability (safety criterion), and upon availability of more than one backup control configurations that satisfy the

safety criterion, to formulate and evaluate a performance index to choose between them (performance consideration). To this end, for each individual control configuration subject to constraints, it is important to implement control laws that provide an explicit estimate of the set of initial conditions starting from where closed-loop stability can be achieved. Such estimates of the stability region can subsequently be used to evaluate the suitability of a given backup control configuration.

Lyapunov-based nonlinear controllers are an example of such controllers that provide an explicit estimate of the stability regions. These controllers, however, are typically not designed to be optimal with respect to arbitrarily specified performance criterion. Model predictive controllers, while typically not allowing for an explicit characterization of their stability region, allow for incorporation of performance considerations, via the objective function or as constraints on the state variables. In the remainder of the paper, we will use a combination of analytical and predictive approaches, at the level of design and analysis or via directly switching between the two control approaches for incorporating performance and robustness considerations in fault-tolerant control of processes. We next review an example of a Lyapunov-based nonlinear controller followed by a representative description of the model predictive control approach.

### 3.2.2 Bounded Lyapunov-Based Control

Referring to the system of Equation 3.1, for a fixed value of  $k \in \mathcal{K}$ , we assume that the uncertain variable term,  $W_k(x)\theta_k$ , 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 [62]),  $V_k$  exists. Consider also, the bounded state feedback



control law (see [49, 46] for details on controller design):

$$u_k^b = - \left( \frac{\alpha_k(x) + \sqrt{(\alpha_{1,k}(x))^2 + (u_k^{max} \beta_k(x))^4}}{(\beta_k(x))^2 \left[ 1 + \sqrt{1 + (u_k^{max} \beta_k(x))^2} \right]} \right) (L_{G_k} V_k)^T := b_k(x) \quad (3.2)$$

when  $L_{G_k} V_k \neq 0$  and  $u_k = 0$  when  $L_{G_k} V_k = 0$ , where

$$\begin{aligned} \alpha_k(x) &= L_f V_k + (\rho_k \|x\| + \chi_k \theta_k^b \|L_{W_k} V_k\|) \left( \frac{\|x\|}{\|x\| + \phi_k} \right) \\ \alpha_{1,k}(x) &= L_f V_k + \rho_k \|x\| + \chi_k \theta_k^b \|L_{W_k} V_k\| \\ \beta_k(x) &= \|(L_{G_k} V_k)^T\| \\ L_{G_k} V_k &= [L_{g_k^1} V_k \cdots L_{g_k^m} V_k] \\ L_{W_k} V_k &= [L_{w_k^1} V_k \cdots L_{w_k^q} V_k], \end{aligned}$$

$L_{G_k} V_k$  and  $L_{W_k} V_k$  are row vectors,  $\theta_k^b$  is a positive real number such that  $\|\theta_k(t)\| \leq \theta_k^b$ , for all  $t \geq 0$ , and  $\rho_k$ ,  $\chi_k$  and  $\phi_k$  are adjustable parameters that satisfy  $\rho_k > 0$ ,  $\chi_k > 1$  and  $\phi_k > 0$ . Let  $\Pi_k$  be the set defined by  $\Pi_k(\theta_k^b, u_k^{max}) = \{x \in \mathbb{R}^n : \alpha_{1,k}(x) \leq u_k^{max} \beta_k(x)\}$  and assume that  $\Omega_k := \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\} \subseteq \Pi_k(\theta_k^b, u_k^{max})$  for some  $c_k^{max} > 0$ . Then, given any positive real number,  $d_k^r$ , such that:

$$\mathbb{D}_k^r := \{x \in \mathbb{R}^n : \|x\| \leq d_k^r\} \subset \Omega_k \quad (3.3)$$

and for any initial condition  $x_0 \in \Omega_k$ , it can be shown that there exists a positive real number  $\epsilon_k^{r*}$  such that if  $\phi_k/(\chi_k - 1) < \epsilon_k^{r*}$ , the states of the closed-loop system of Equations 3.1-3.2 satisfy  $x(t) \in \Omega_k \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k^r$ .

**Remark 3.1** Referring to the above controller design, it is important to note that a general procedure for the construction of RCLFs for nonlinear systems of the form of Equation 3.1 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 (see, for example, [97, 62]). Note

also that the computation of the stability region above only involves algebraic computations, and furthermore, for implementation purposes, the entire stability region information is contained in the value of the level set  $c_k^{max}$  which defines the boundary of the stability region. The presence of a given initial condition in the stability region can be ascertained by simply checking the value of the Lyapunov function for the given initial condition against  $c_k^{max}$  ( $V(x(0)) \leq c_k^{max}$  implies  $x(0) \in \Omega_k$ ). Note also that possibly larger estimates of the stability region can be computed using constructive procedures such as Zubov's method [42] or by using a combination of several Lyapunov functions.

### 3.2.3 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. 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:

$$\begin{aligned}
u_k(\cdot) &= \operatorname{argmin}\{ \max\{J_s(x, t, u_k(\cdot)) | \theta_k(\cdot) \in \Theta_k\} | u_k(\cdot) \in S_k\} := M_k \\
s.t. \quad \dot{x}(t) &= f(x(t)) + G_k(x)u_k + W_k(x)\theta_k(t) \\
x(0) &= x_0, \quad x(t + T_k) \in \Omega_{MPC}(x, t, \theta_k) \\
J_s(x, t, u_k(\cdot)) &= \int_t^{t+T_k} (x'(s)Q_k x(s) + u'(s)R_k u(s))ds + F_k(x(t + T_k))
\end{aligned} \tag{3.4}$$

and  $S_k = S_k(t, T)$  is the family of piecewise continuous functions, with period  $\Delta_k$ , mapping  $[t, t+T_k]$  into the set of admissible controls,  $T_k$  is the horizon length and  $\theta_k$  is the bounded uncertainty assumed to belong to a set  $\Theta_k$ . 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  $t \in [j\Delta_k, (j+1)\Delta_k)$ .  $J_s$  is the performance index,  $R_k$  and  $Q_k$  are strictly pos-

itive definite, symmetric matrices and the function  $F_k(\cdot)$  represents a penalty on the states at the end of the horizon. The maximization over  $\theta_k$  may not be carried out if the MPC version used is not a min-max type of formulation. The set  $\Omega_{MPC}(x, t, \theta_k)$  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.

### **3.3 Fault-Tolerant Control: Performance-Based Reconfiguration**

To clearly illustrate the main idea behind incorporating performance considerations in fault-tolerant control of processes, in this section we consider processes without uncertainty. The performance considerations are incorporated both at the lower-level; by using a predictive control design described in next section that incorporates performance objectives without sacrificing the explicit characterization of the stability region (essential to implementing fault-tolerant control in the proposed method) and also at the upper-level; by incorporating performance considerations in the switching rule.

### 3.3.1 Lyapunov-Based Predictive Control

We review 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 more details, see [115]). Preparatory to the characterization of the stability properties of the Lyapunov-based predictive controller, we first present a proposition stating the stability properties of the bounded controller of Equation 3.2. Specifically, the bounded controller of Equation 3.2 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 ( $\Delta$ ). The control law ensures that, for all initial conditions in  $\Omega_k$ , the closed-loop state remains in  $\Omega_k$  and eventually converges to some neighborhood of the origin whose size depends on  $\Delta$ . This robustness property, stated below in Proposition 3.1, is exploited in the Lyapunov-based predictive controller design (for a proof, see [115]). For further results on the analysis and control of sampled-data nonlinear systems, the reader may refer to [71, 128, 87, 182].

**Proposition 3.1** *Consider the constrained system of Equation 3.1 for a fixed value of  $k$  with  $\theta_k(t) = 0 \forall t \geq 0$ , under the bounded control law of Equation 3.2 designed using the Lyapunov function  $V_k$  and  $\rho_k > 0$ , and the stability region estimate  $\Omega_k$  under continuous implementation. Let  $u_k(t) = u_k(j\Delta_k)$  for all  $j\Delta_k \leq t < (j+1)\Delta_k$  and  $u_k(j\Delta_k) = b_k(x(j\Delta_k))$ ,  $j = 0, \dots, \infty$ . Then, given any positive real number  $d_k$ , there exist positive real numbers  $\Delta_k^*$ ,  $\delta_k'$  and  $\epsilon_k^*$  such that if  $\Delta_k \in (0, \Delta_k^*]$  and  $x(0) := x_0 \in \Omega_k$ , then  $x(t) \in \Omega_k \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$ . Also, if  $V_k(x_0) \leq \delta_k'$  then  $V_k(x(\tau)) \leq \delta_k' \forall \tau \in [0, \Delta_k)$  and if  $\delta_k' < V_k(x_0) \leq c_k^{max}$ , then  $\dot{V}_k(x(\tau)) \leq -\epsilon_k^* \forall \tau \in [0, \Delta_k)$ .*

For the Lyapunov-based 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

$$P(x, t) : \min\{J(x, t, u_k(\cdot)) | u_k(\cdot) \in S_k\} \quad (3.5)$$

$$s.t. \quad \dot{x} = f_k(x) + G_k(x)u_k \quad (3.6)$$

$$\dot{V}_k(x(\tau)) \leq -\epsilon_k \quad \text{if} \quad V_k(x(t)) > \delta'_k, \quad \tau \in [t, t + \Delta_k) \quad (3.7)$$

$$V_k(x(\tau)) \leq \delta'_k \quad \text{if} \quad V_k(x(t)) \leq \delta'_k, \quad \tau \in [t, t + \Delta_k) \quad (3.8)$$

where  $\epsilon_k, \delta'_k$  are defined in Proposition 3.1,  $S_k = S_k(t, T)$  is the family of piecewise continuous functions (functions continuous from the right), with period  $\Delta_k$ , mapping  $[t, t + T_k]$  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  $u_k[j] := u_k(j\Delta_k)$  and satisfies  $u_k(t) = u_k[j]$  for all  $t \in [j\Delta_k, (j+1)\Delta_k)$ . The performance index is given by

$$J(x, t, u_k(\cdot)) = \int_t^{t+T} \left[ \|x^u(s; x, t)\|_{Q_k}^2 + \|u_k(s)\|_{R_k}^2 \right] ds \quad (3.9)$$

where  $Q_k, R_k$  are positive semi-definite, strictly positive definite, symmetric matrices, respectively, and  $x^u(s; x, t)$  denotes the solution of Equation 3.1, due to control  $u_k$ , 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  $[t, t + \Delta_k)$  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 stated in Proposition 3.2 below (for a proof and more details, see [115]).

**Proposition 3.2** Consider the constrained system of Equation 3.1 for a fixed value of  $k$  with  $\theta_k(t) = 0 \forall t \geq 0$  under the MPC control law of Equations 3.5-3.9, designed using a control Lyapunov function  $V_k$  that yields a stability region  $\Omega_k$  under continuous implementation of the bounded controller of Equation 3.2 with a fixed  $\rho_k > 0$ . Then, given any positive real number  $d_k$ , there exist positive real numbers  $\Delta_k^*$  and  $\delta_k'$ , 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  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k$ .

**Remark 3.2** Note that Lyapunov-based predictive control approaches (see, for example, [93, 137]) typically incorporate a similar Lyapunov function decay constraint, albeit requiring the constraint of Equation 3.7 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 Equations 3.5-3.9 requires that the value of the Lyapunov function decrease during the first step only, allowing for the use of the auxiliary controller to explicitly characterize the set of initial conditions starting from where the predictive controller is guaranteed to be feasible and stabilizing.

**Remark 3.3** 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  $\Delta_k$ , the bounded controller can also only guarantee that the state of the closed-loop system converges to a neighborhood of the origin the size of which is limited by the value of the hold time,  $\Delta_k$  (in the limit as  $\Delta_k$  goes to zero – continuous implementation – the bounded controller and the predictive controller enforces asymptotic stability). Note also, that any other Lyapunov-based nonlinear 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 3.4** One of the key challenges that impact on the practical implementation of MPC is the inherent difficulty of characterizing, *a priori*, the set of initial conditions starting from where a given MPC 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.

### 3.3.2 Performance-Based Reconfiguration

The main idea behind the fault-tolerant control design is as follows: (1) use the presence of the state in the stability regions of the candidate control configurations to compute the set of suitable backup configurations, and (2) use the auxiliary Lyapunov-based nonlinear controller to estimate the ‘cost’ under each of the suitable control configurations, and choose the one with the minimum cost. To formalize this idea, consider the constrained nonlinear system of Equation 3.1 without uncertainty (i.e.,  $\theta_k(t) = 0 \forall t \geq 0$  and  $\forall k = 1, \dots, N$ ) for which the bounded controllers of the form of Equation 3.2 and Lyapunov-based predictive controllers of the form of Equations 3.5-3.8 have been designed and the stability regions  $\Omega_j$ ,  $j = 1, \dots, N$  under the Lyapunov-based predictive controllers have been explicitly characterized. Let  $d_{max} = \max_{j=1, \dots, N} d_j$ , where  $d_j$  was defined in proposition 3.1 and let  $\Omega_U = \bigcup_{j=1}^N \Omega_j$ . For a given control configuration, define  $J_j(t) = \int_t^{t+T_j} [\|x^u(s; x, t)\|_Q^2 + \|b_k(s)\|_R^2] ds$  where  $t + T_j \geq t$  is the earliest time at which the state of the closed-loop system under the bounded controller enters the level set defined by  $V_j(x) = \delta'_j$ , and  $Q_j$ ,  $R_j$  are the penalty matrices used in the predictive controller design. Theorem 3.1 below formalizes the

result.

**Theorem 3.1** *Let  $k(0) = i$  for some  $i \in \mathcal{K}$  and  $x(0) := x_0 \in \Omega_i$ . Let  $T_i^f$  be the earliest time that a fault occurs. Furthermore, let  $\mathcal{F} \in \mathcal{K} := \{j : j \neq i, x(T_i^f) \in \Omega_j\}$ , and let  $l$  be such that  $J_l = \min_{j \in \mathcal{F}} J_j$  then the following switching rule:*

$$k(t) = \begin{cases} i, & 0 \leq t < T_i^f \\ l, & t \geq T_i^f \end{cases} \quad (3.10)$$

*guarantees that  $x(t) \in \Omega_U \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_{max}$ .*

**Proof of Theorem 3.1** We consider the two possible cases; first if no switching occurs, and second if a switch occurs at a time  $T_i^f$ .

*Case 1:* The absence of a switch implies  $k(t) = i \forall t \geq 0$ . Furthermore, since  $x(0) \in \Omega_i$ , and control configuration  $i$  is implemented for all times in this case, we have that  $x(t) \in \Omega_i \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_i$ . Finally, since  $\Omega_i \subseteq \Omega_U$  and  $d_i \leq d_{max}$ , we have that  $x(t) \in \Omega_U \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_{max}$ .

*Case 2:* At time  $T_i^f$ , the supervisor switches to a control configuration  $l$  for which  $x(T_i^f) \in \Omega_l$ . From this time onwards, since configuration  $l$  is implemented in the closed-loop system for all times, and since  $x(T_i^f) \in \Omega_l$ , we have that  $x(t) \in \Omega_l \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_l$ . As in case 1, since  $\Omega_l \subseteq \Omega_U$  and  $d_l \leq d_{max}$ , we have that  $x(t) \in \Omega_U \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_{max}$ .

This completes the proof of Theorem 3.1.

**Remark 3.5** The fault-tolerant controller is implemented as follows:

- Given the nonlinear process of Equation 3.1, identify the available control configurations  $k = 1, \dots, N$  and for each control configuration, design the controllers of Equation 3.2, and Equations 3.5-3.8 and calculate an estimate of the stability region  $\Omega_k$ ,  $k = 1, \dots, N$ .
- Given any  $x_0 \in \Omega_i$ , initialize the closed-loop system under the Lyapunov-based predictive controller of Equations 3.5-3.8.



- At any time  $T_i^f$  that a fault occurs, out of the available backup configurations ascertain the suitability of a candidate backup configuration  $j \neq i$  (i.e., other than the current one) via checking whether or not the state of the closed-loop system resides in the stability region estimate under the candidate control configuration (i.e., to check if  $x(T_i^f) \in \Omega_j$ ). If the state of the closed-loop system resides in the stability region of control configuration  $j$ , include its index in the set  $\mathcal{F}$ . For all the backup-configurations whose index is present in the set  $\mathcal{F}$ , compute the cost  $J_j$ , by running closed-loop simulations under the bounded controller of Equation 3.2, over a time by which the bounded controller drives the closed-loop state into the neighborhood of the origin defined by the level set of  $V_j(x) = \delta'_j$ .
- Pick the control configuration that yields the lowest cost. Apply the Lyapunov-based predictive controller using this control configuration to achieve closed-loop stability.

**Remark 3.6** Fault-tolerant controller in this chapter incorporates performance considerations in the switching logic as well as in computing the control action under the fall-back control configurations. 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, the performance considerations expressed in the objective function are used in choosing which control configuration should be implemented in the closed-loop system. Note, however, that the receding horizon implementation of the predictive controller renders it unsuitable for evaluating online an estimate of the value of the objective function in driving the state from the current value to the equilibrium point (the cost-to-go). To this end, the auxiliary controller is used in estimating the control configuration that yields a lower cost; the practical justification behind doing this is that (1) the Lyapunov-based predictive controller enforces the decay of the same Lyapunov function that is used in the auxiliary controller, and (2) the auxiliary controller provides an explicit control law, thus making

it easier to estimate the ‘cost-to-go’ using fast simulations. In case that the cost-to-go can be computed using other computational techniques, these can be used within the proposed approach to pick the appropriate backup control configuration that yields the lowest cost. Either ways, once the cost has been estimated, the optimization problem in the switching logic involves only finding the minimum out of a set of numbers (costs), and picking out the index that corresponds to the minimum cost. Note also that if the state at the time of a failure lies outside the stability region of all of 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.

### 3.3.3 Application to Chemical Process Example

We first design the Lyapunov-based predictive controller and compute an estimate of the stability region under each control configuration using the auxiliary Lyapunov-based bounded controller. In the simulations, the constraints of Equations 3.7-3.8 are replaced by a constraint of the form  $V_k(x(t + \Delta_k)) \leq V_k^b(x(t + \Delta_k))$  (with  $\Delta_k = 0.02$  min) where  $V_k^b(x(t + \Delta_k))$  is the predicted value of the Lyapunov function at  $t + \Delta_k$  under the auxiliary controller. Note that once again the control action computed by the auxiliary controller provides a feasible solution to this constraint. Figure 3.1 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 under 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.

We first demonstrate the overriding stability considerations in the choice of the backup control configuration, i.e., a case where at the time of the failure of the primary control configuration, the state of the closed-loop system resides in the stability region of only one of the backup control configurations, and only a switch to that control configuration achieves closed-loop stability. As shown by the solid lines in Figures 3.1-3.2, the controller proceeds to drive the closed-loop trajectory towards the desired steady-state, up until the  $Q$ -configuration fails after 1 minute of reactor startup (see Figure 3.3(a)). If the supervisor switches arbitrarily, and in particular, switches to backup configuration 3, closed-loop stability is not achieved (dashed lines in Figures 3.1-3.2). Note that this happens because the closed-loop process state is outside the stability region of the third control configuration, and even though the third control configuration does not encounter a fault, the limited control action available in this configuration is unable to achieve closed-loop stability. From Figure 3.1, it is clear that the failure of the primary control configuration occurs when the closed-loop trajectory is within the stability region of the second control configuration. Hence, on the basis of the switching logic of Equation 3.10, when the supervisor activates the second configuration (with  $T_{A0}$  as the manipulated input, see Figure 3.3(b)), the result is that upon switching to the  $T_{A0}$ -configuration, the corresponding controller stabilizes the closed-loop system.

We next demonstrate the scenario where performance considerations dictate the choice of the backup control configuration. To this end, consider the closed-loop system from the same initial condition as before under control configuration 1, but that control configuration 1 continues to be operative until 5.5 minutes, and at the

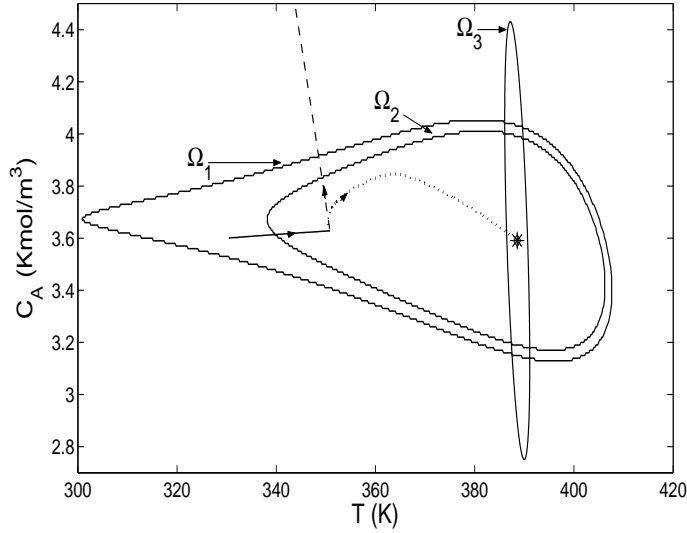
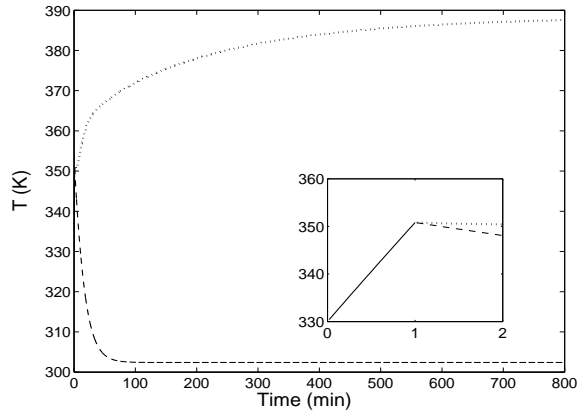
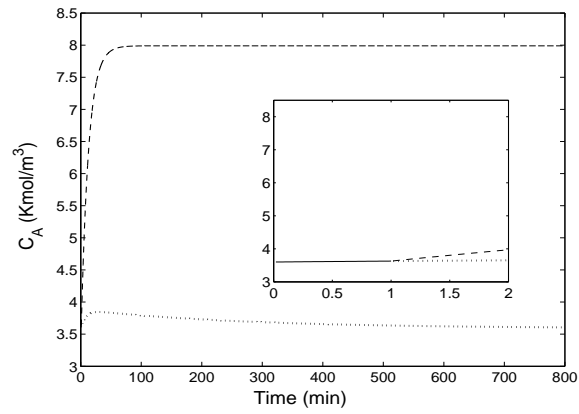


Figure 3.1: Evolution of closed-loop state profiles subject to failure in control configuration 1 (solid line) under the switching rule of Theorem 3.1 (dotted line) and under arbitrary switching (dashed line).

time of the failure, the closed-loop state resides in the stability region of both the backup control configurations (see Figure 3.4). The auxiliary controllers are used to estimate the cost under the control configurations 2 and 3, and yield costs of 307.88 and 105.31, respectively. Using the switching rule, control configuration 3 is implemented in closed-loop system and stabilizes the closed-loop incurring a cost of 105.31. In contrast, if one were to use configuration 2, the cost incurred would be 276.94 which is lower than the estimate obtained using the auxiliary controller, yet more than the cost incurred under control configuration 3 (the corresponding state and input profiles are showed by dashed and dotted lines in Figures 3.5-3.6, respectively).

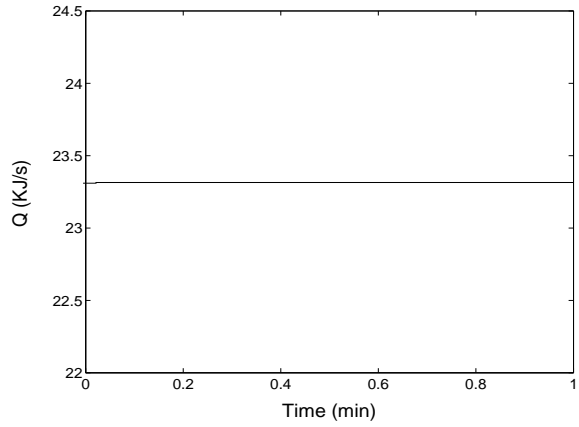


(a)

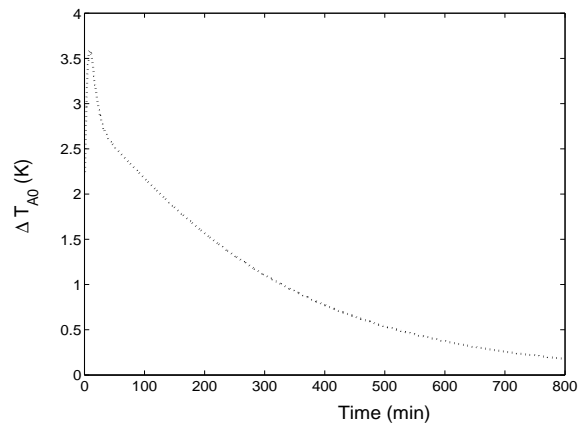


(b)

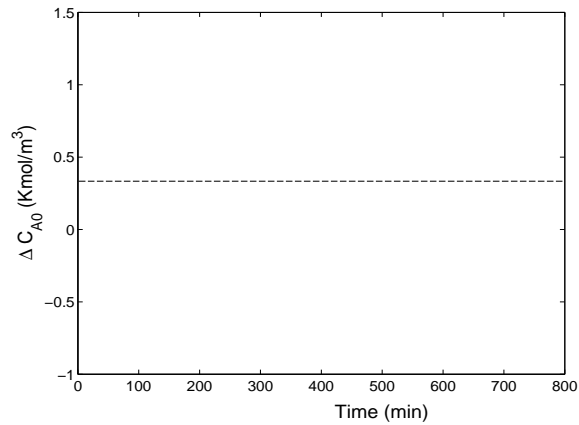
Figure 3.2: Evolution of closed-loop (a) temperature and (b) concentration subject to failure in control configuration 1 (solid lines) under the switching rule of Theorem 3.1 (dotted lines) and under arbitrary switching (dashed lines).



(a)



(b)



(c)

Figure 3.3: Manipulated input profiles under (a) control configuration 1 (solid line), (b) control configuration 2 (under the switching rule of Theorem 3.1 (dotted line)), and (c) control configuration 3 (under arbitrary switching (dashed line)).

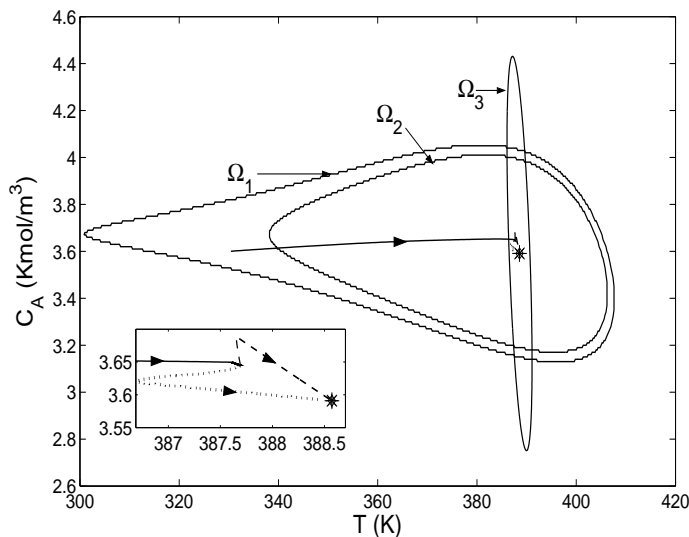
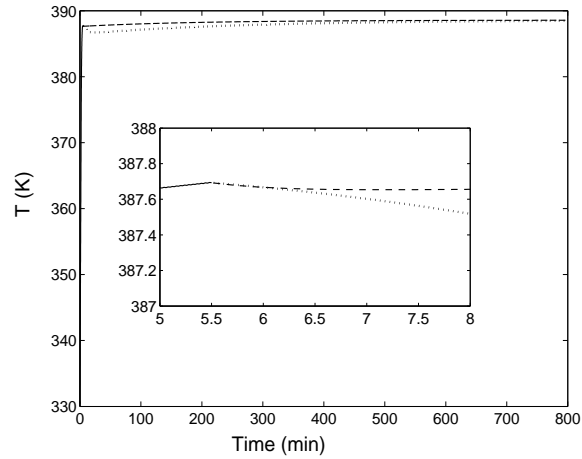


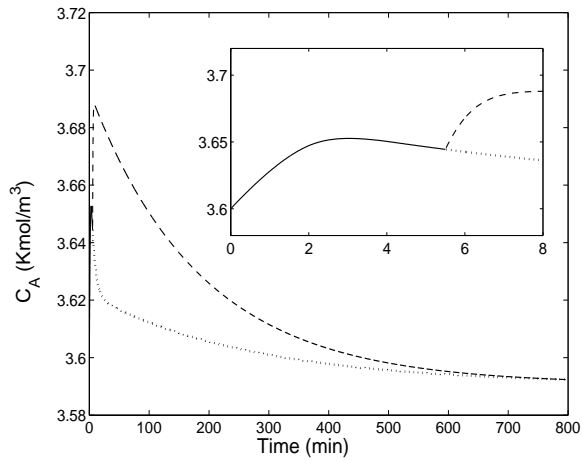
Figure 3.4: Evolution of closed-loop state profiles subject to failure in control configuration 1 (solid line) and switching to configuration 2 (dotted line) and, according to the switching rule of Theorem 3.1, to configuration 3 (dashed line).

### 3.4 Fault-Tolerant Control: Robustness Considerations

In this section, we consider the problem of incorporating robustness into the fault-tolerant control method. Note that in the presence of uncertainty, the feasibility guarantees of the predictive controller of Equations 3.5-3.8 may no longer hold, or it may happen that the predictive controller is feasible but not stabilizing (enforcing negative-definiteness of  $\dot{V}$  *without* accounting for the uncertainty does not imply that  $\dot{V} < 0$  in the presence of uncertainty). Preparatory to its use within the robust fault-tolerant controller, we review a robust hybrid predictive controller that provides an explicit characterization of the stability region in the presence of uncertainty and input constraints.



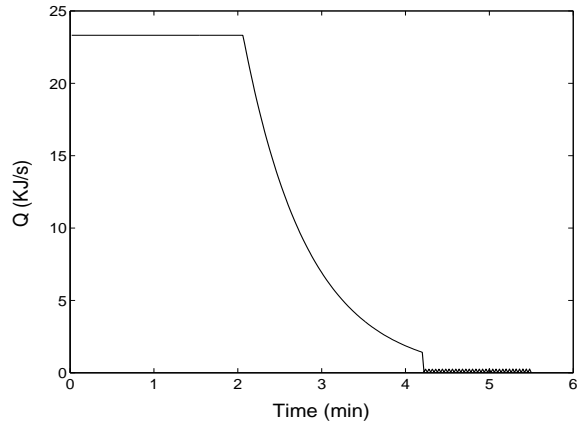
(a)



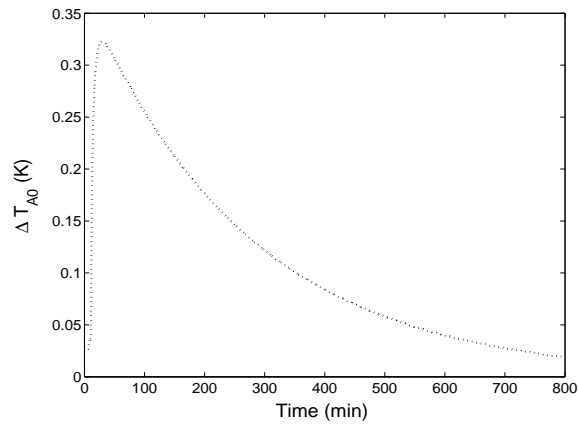
(b)

Figure 3.5: Evolution of closed-loop (a) temperature and (b) concentration subject to failure in control configuration 1 (solid line) and switching to configuration 2 (dotted lines) and, according to the switching rule of Theorem 3.1, to configuration 3 (dashed lines).

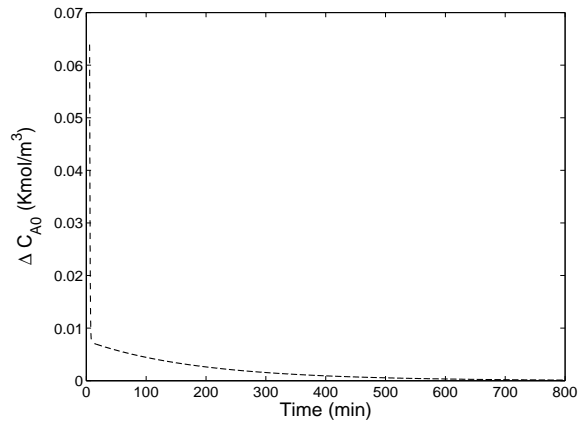




(a)



(b)



(c)

Figure 3.6: Manipulated input profiles under (a) control configuration 1 (solid line), (b) control configuration 2 (dotted lines) and (c) according to the switching rule of Theorem 3.1 to control configuration 3 (dashed lines).

### 3.4.1 Robust Hybrid Predictive Controller

In this section, we employ switching between the bounded robust controller of Equation 3.2 and the representative predictive controller of Equation 3.4, to provide the switched (the switching here is between control algorithms, and not control configurations) closed-loop system with an explicit characterization of the closed-loop stability region. To this end, we first cast the system of Equation 3.1, for a fixed value of  $k$  (i.e., under a given control configuration) as a switched system of the form:

$$\dot{x} = f(x) + G_k(x)u_k^j + W_k(x)\theta_k(t), \quad \|u_k^j\| \leq u_k^{max} \quad (3.11)$$

where  $j : [0, \infty) \rightarrow \{1, 2\}$  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,  $j(t)$ , represents a discrete state that indexes the control input,  $u_k$ , with the understanding that  $j(t) = 1$  if and only if MPC is used and  $j(t) = 2$  if and only if bounded control is used. Theorem 3.2 below presents the robust hybrid predictive controller (for the proof and more details, see [116]).

**Theorem 3.2** *Consider the switched nonlinear system of Equation 3.11, the model predictive controller of Equation 3.4 and the bounded controller of Equation 3.2. Let  $x(0) = x_0 \in \Omega_k$ , and initially set  $T_k^s = T_k^D = T_k^{inf} = \infty$ . At the earliest time  $t \geq 0$  for which the closed-loop state under MPC satisfies  $V_k(x(t^-)) = c_k^{max}$  set  $T_k^s = t$ . At the earliest time for which the closed-loop state under MPC satisfies  $\|x(t)\| \leq d_k^r$  where  $d_k^r$  was defined in Equation 3.3, set  $T_k^D = t$ . Finally, at the earliest time  $t$  that MPC is infeasible, set  $T_k^{inf} = t$ . Define  $T_k^{switch} = \min\{T_k^s, T_k^D, T_k^{design}, T_k^{inf}\}$ , where  $0 \leq T_k^{design} < \infty$  is arbitrary. Then, the switching rule*

$$j(t) = \left\{ \begin{array}{ll} 1, & 0 \leq t < T_k^{switch} \\ 2, & t \geq T_k^{switch} \end{array} \right\} \quad (3.12)$$

guarantees that  $x(t) \in \Omega_k \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_k^r$ .

**Remark 3.7** The robust hybrid predictive controller of Theorem 3.1 is designed and implemented to achieve closed-loop stability using a control configuration  $k$  as follows:

- Given the nonlinear system of Equation 3.11,  $\theta_k^b$  and  $u_k^{max}$ , design the bounded robust controller of Equation 3.2, and calculate an estimate of its stability region  $\Omega_k$  for the control configuration  $k$ .
- 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 Equation 3.4.
- Given any  $x_0 \in \Omega_k$ , check the feasibility of the optimization problem in Equation 3.4 at  $t = 0$ , and if feasible, start implementing MPC.
- If at any time, MPC becomes infeasible ( $t = T_k^{inf}$ ), or the states of the closed-loop system approach the boundary of  $\Omega_k$  ( $t = T_k^s$ ), or the closed-loop states enter the set  $\mathbb{ID}_k^r$  ( $t = T_k^D$ ), 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  $T_k^s$ ,  $T_k^D$ ,  $T_k^{design}$ , or  $T_k^{inf}$ , whichever comes earliest, to achieve practical closed-loop stability under the  $k$ -th control configuration.

**Remark 3.8** The purpose of switching to the bounded robust controller after the time  $T_k^{design}$  is to ensure convergence to  $\mathbb{ID}_k^r$  and avoid possible cases where the closed-loop states, under MPC, could wander inside  $\Omega_k$  without actually converging to, and staying within,  $\mathbb{ID}_k^r$ . Convergence to  $\mathbb{ID}_k^r$  could also be achieved (see, for example, [55, 54]), by switching to the bounded controller when  $\dot{V}_k \geq 0$  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 ( $T_k^{design}$  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 3.9** 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.

### 3.4.2 Robust Fault-Tolerant Control

The robust fault-tolerant controller is implemented as follows:

1. Given the nonlinear process of Equation 3.1, identify the available control configurations  $k = 1, \dots, N$  and for each control configuration, design the robust hybrid predictive controllers of Theorem 3.2 and calculate an estimate of the stability region  $\Omega_k$ ,  $k = 1, \dots, N$ .
2. Given any  $x_0 \in \Omega_k$ , initialize the closed-loop system under the robust hybrid predictive controller of Theorem 3.2.
3. At any time  $T_1^f$  that a fault occurs, implement the control configuration  $j$  for which the closed-loop state resides in its stability regions estimate ( $\Omega_j$ ) to achieve closed-loop stability.

**Remark 3.10** Note that robustness considerations are incorporated in the controller design (use of robust hybrid predictive controllers) and also in characterizing the stability region. Performance considerations can be incorporated in the switching rule in a similar fashion as in the previous section, and in the design of controllers via use of robust predictive control designs as a component of the robust hybrid predictive controllers (for a demonstration, see the simulation example below).

### 3.4.3 Application to Chemical Process Example with Uncertainty and Disturbance

In this section, we consider once again the motivating example of chemical process reactor, albeit with uncertainty and disturbances. In particular, we consider parametric uncertainty in the heat of reactions, and in particular a 50% uncertainty in the heats of reactions, i.e.,  $\theta_i(t) = 0.5(-\Delta H_{i,nom})$ ,  $i = 1, \dots, 3$ , and disturbance in the inlet feed temperature, simulated by  $\theta_4(t) = 0.5T_{A0s} \sin t$ . Figure 3.7 depicts the stability region computed using the bounded robust controller with  $\rho = 0.0001$ ,  $\phi = 0.0001$ ,  $\chi = 1.0001$ , in the  $(T, C_A)$  space, for the control configurations using  $Q$  as the manipulated input variable and using  $T_{A0}$  as the manipulated input variable. The desired steady-state is depicted with an asterisk that lies in the intersection of the two stability regions (note the reduction in the estimate of the stability region as a result of accounting for the presence of uncertainty).

The hybrid predictive control structure allows for the use of any predictive controller formulation, while still guaranteeing stability from an explicitly characterized set of initial conditions. Within the hybrid predictive controller, we use a modification of the Lyapunov-based predictive controller of Section 3.3.1. In particular, for the 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

$$P(x, t) : \min\{J(x, t, u_k(\cdot)) | u_k(\cdot) \in S_k, \theta_k(t) = \theta_0 \in \Theta_k\} \quad (3.13)$$

$$s.t. \dot{x} = f_k(x) + G_k(x)u_k + W_k(x)\theta_k(t) \quad (3.14)$$

$$V_k(x(t + \Delta_k)) \leq V_k^b(x(t + \Delta_k)) \quad (3.15)$$

where  $V_k^b(x(t + \Delta_k))$  is the predicted value of the Lyapunov function at  $t + \Delta$  under the robust bounded controller with  $\theta_k(t) = \theta_0 \in \Theta_k$ ,  $S_k = S_k(t, T)$  is the family of piecewise continuous functions (functions continuous from the right), with period  $\Delta_k$ , mapping  $[t, t + T_k]$  into  $\mathcal{U}_k$ ,  $T_k$  is the specified horizon and  $V_k$  is the Lyapunov function used in the bounded controller design. The performance index is given by

$$J(x, t, u_k(\cdot), \theta_0) = \int_t^{t+T_k} [\|x^u(s; x, t)\|_{Q_k}^2 + \|u_k(s)\|_{R_k}^2] ds \quad (3.16)$$

where  $Q_k, R_k$  are positive semi-definite, strictly positive definite, symmetric matrices, respectively, and  $x^u(s; x, t)$  denotes the solution of Equation 3.1, due to control  $u_k$  under a fixed value of uncertainty  $\theta_k(t) = \theta_0$ , 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  $[t, t + \Delta_k)$  and the procedure is repeated indefinitely.

Note that as in the case without uncertainty, initial feasibility of the optimization problem of Equations 3.13-3.16 is guaranteed for all initial conditions within the stability region of the bounded robust controller. There is no guarantee, however, that the control action computed by the predictive controller will lead to a decay in the value of the Lyapunov function; this is so because the control action is computed by using only a fixed value of the uncertainty, and is not computed to ensure the satisfaction of the Lyapunov-function decay constraint for all possible realizations of the uncertainty, as is customarily done in robust predictive control approaches.

The modification used in the simulation example, however, while not providing rigorous robust stability guarantees, incorporates some robustness consideration in the Lyapunov-based predictive controller without making the computation intractable by requiring min-max computations.

The reactor under the first control configuration is initialized at  $T(0) = 360\text{ K}$ ,  $C_A(0) = 3.7\text{ kmol/m}^3$ ,  $C_B(0) = 0.0\text{ kmol/m}^3$ , using the  $Q$ -control configuration, under the hybrid predictive controller for configuration 1 (with  $T^{design} = 100\text{ min}$ ) and the supervisor proceeds to monitor the evolution of the closed-loop trajectory. As shown by the solid lines in Figures 3.7-3.8, 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 Figure 3.9(a)). Until this time, only the predictive controller component of the robust hybrid predictive controller is used for the first control configuration. From Figure 3.7, it is clear that the failure of the primary control configuration occurs when the closed-loop trajectory is within the stability region of the second control configuration. Hence, on the basis of the switching algorithm, when the supervisor activates the second configuration (with  $T_{A0}$  as the manipulated input, see Figure 3.9(b)), the result is that upon switching to the  $T_{A0}$ -configuration, the corresponding robust hybrid predictive controller stabilizes the closed-loop system. Note also that in operating the second control configuration, the robust Lyapunov-based predictive controller is able to drive the state trajectory sufficiently close to the origin, and the robust bounded controller is used only toward the end to drive the state trajectory into the desired neighborhood of the origin.

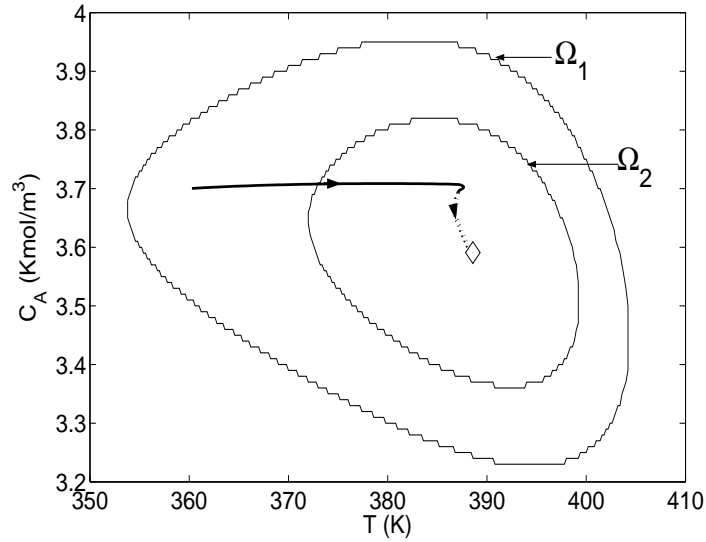
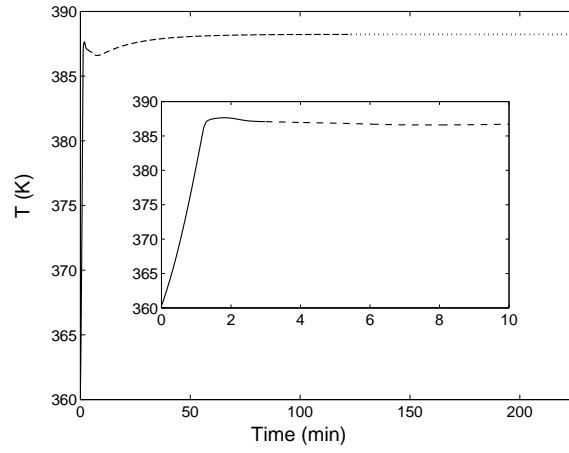


Figure 3.7: Evolution of closed-loop state profiles under the switching rule of Section 3.4.2 subject to failure in control system 1.

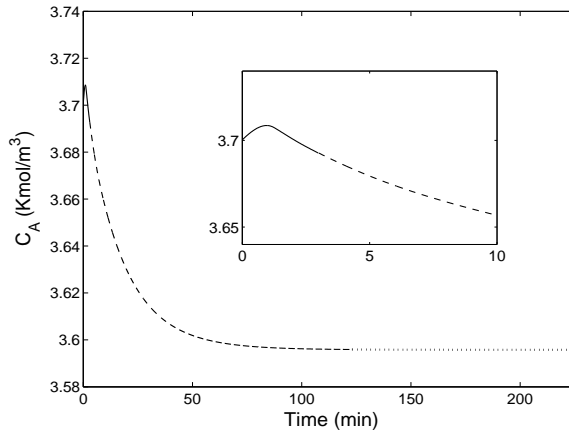
### 3.5 Conclusions

In this chapter, we considered the problem of control system/actuator failures in nonlinear processes subject to input constraints and presented two approaches for fault-tolerant control that focussed on incorporating performance and robustness considerations, respectively. Performance considerations were incorporated in the design of the controllers (via the use of predictive control approach) as well as in the reconfiguration logic to achieve fault-tolerant control. To handle the problem of uncertainty, robust hybrid predictive controllers were designed for the individual control configurations. The application of the fault-tolerant control methods incorporating performance and robustness considerations was demonstrated via a benchmark chemical reactor example.



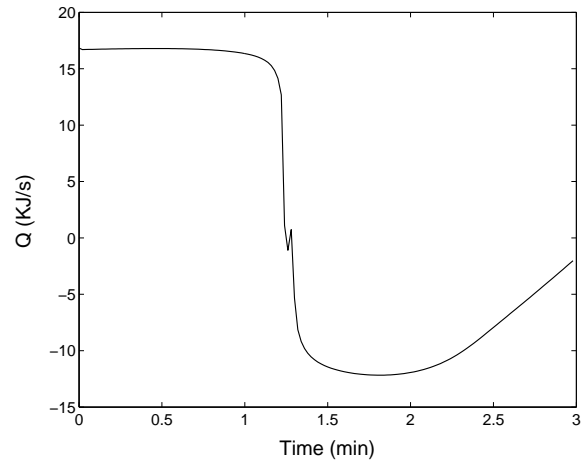


(a)

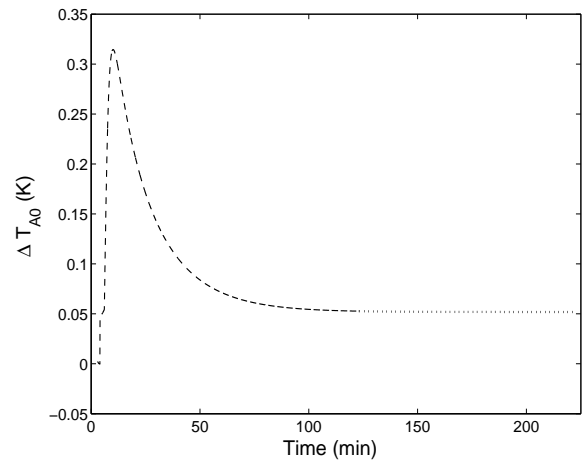


(b)

Figure 3.8: Evolution of closed-loop (a) temperature and (b) concentration under the switching rule of Section 3.4.2 subject to failure in control system 1.



(a)



(b)

Figure 3.9: Manipulated input profiles under (a) control configuration 1 and (b) control configuration 2 under the switching rule of Section 3.4.2 subject to failure in control system 1.

## Chapter 4

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

### 4.1 Introduction

In process control, given the complex dynamics of chemical processes (for example, nonlinearities, 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.

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 [14, 180, 79]; for a more recent review, see [29]. 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 (for example, [66, 109, 59]), Lyapunov-based control (for example, [103, 158, 85, 92, 46, 48]) and hybrid predictive control (for example, [54, 116]).

The occurrence of faults in chemical processes and subsequent switching to fallback 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 (for example, [72, 68, 80, 39, 13, 49]), 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 Chapter 2, 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 (for example, [96, 145, 131, 44, 126, 43, 35, 156, 4, 187]) 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 [108, 60, 61, 112]; and more recently, some existential results in the context of nonlinear systems have been derived [146, 37].

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 as well as to investigate an application in the presence of uncertainty and measurement noise [119].

## 4.2 Preliminaries

### 4.2.1 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:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g_{k(t)}(x(t))(u_{k(t)} + m_{k(t)}), & y_m &= h_m(x) \\ k(t) &\in \mathcal{K} = \{1, \dots, N\}, \quad N < \infty, & |u_{k(t)}| &\leq u_{max}^k \end{aligned} \quad (4.1)$$

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 this chapter, 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 [91]. 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 Equation 4.1 exists and is continuous for all  $t$ .

### 4.2.2 Motivating Example

To illustrate our fault-tolerant control design methodology, we use a benchmark chemical reactor example introduced in Section 2.4.1. 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. 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 con-

trol 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. Note also, that while the above example will be used to illustrate the main ideas behind the proposed fault-detection and fault-tolerant control method, we also investigate in the simulation studies an application to a network of chemical reactors in the presence of uncertainty and measurement noise.

### 4.2.3 Bounded Lyapunov-Based Control

Consider the system of Equation 4.1, for which a family of control Lyapunov functions (CLFs),  $V_k(x)$ ,  $k \in \mathcal{K} \equiv \{1, \dots, N\}$  has been found (see below for a discussion on the construction of CLFs). Using each control Lyapunov function, we construct, using the results in [103] (see also [46]), the following continuous bounded control law:

$$u_k(x) = -\frac{L_f^* V_k(x) + \sqrt{\left(L_f^* V_k(x)\right)^2 + (u_{max}^k \|(L_{g_k} V_k)(x)\|)^4}}{\|(L_{g_k} V_k)(x)\|^2 \left[1 + \sqrt{1 + (u_{max}^k \|(L_{g_k} V_k)(x)\|)^2}\right]} (L_{g_k} V_k)(x) \quad (4.2)$$

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  $\Pi_k$  be the set defined by

$$\Pi_k(u_{max}^k) = \{x \in \mathbb{R}^n : L_f^* V_k(x) \leq u_{max}^k \|(L_{g_k} V_k)(x)\|\} \quad (4.3)$$

and assume that

$$\Omega_k := \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\} \subseteq \Pi_k(u_{max}^k) \quad (4.4)$$

for some  $c_k^{max} > 0$ . It can be shown, using standard Lyapunov arguments, that in the absence of faults ( $m_{k(t)} = 0$ ),  $\Omega_k$  provides an estimate of the stability region,



starting from where the control law of Equation 4.2 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  $\beta_i$ ,  $i = 1, \dots, N$ , such that  $\|x(t)\| \leq \beta_i(\|x(0)\|, t)$ . We will use this property later in the design of the output feedback controllers.

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 Equation 4.1 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, [97, 62]). Second, given that a CLF,  $V_k$ , has been obtained for the system of Equation 4.1, it is important to clarify the essence and scope of the additional assumption that there exists a level set,  $\Omega_k$ , of  $V_k$  that is contained in  $\Pi_k$ . Specifically, the assumption that the set,  $\Pi_k$ , 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  $\Pi_k$  remain within  $\Pi_k$  for all times). Moreover, the assumption that  $\Omega_k$  is a level set of  $V_k$  is made only to simplify the construction of  $\Omega_k$ . 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  $\Pi_k$  is not available. However, the proposed control method remains applicable if the invariant set  $\Omega_k$  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 [42] or by using a combination of several Lyapunov functions.

## 4.3 Integrated Fault-Detection and Fault-Tolerant Control: State Feedback Case

### 4.3.1 State Feedback Fault-Tolerant Control

Consider the system of Equation 4.1, 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 Equation 4.2. 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:

$$\dot{w}(t) = f_f(w, x), \quad r(t) = h_f(w, x), \quad k(t) = \varphi(r, w, x) \quad (4.5)$$

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 Equation

4.1 for which a bounded controller of the form of Equation 4.2 has been designed for each control configuration, and the stability region,  $\Omega_j$ ,  $j = 1, \dots, N$  has been explicitly characterized. The fault-detection filter and the fault-tolerant control design are described in Theorem 4.1 below.

**Theorem 4.1** *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*

$$\dot{w} = f(w) + g_i(w)u_i(w); \quad r = \|w - x\| \quad (4.6)$$

where  $w \in \mathbb{R}^n$  is the filter state and  $u_i(\cdot)$  is the feedback control law defined in Equation 4.2. 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:

$$k(t) = \left\{ \begin{array}{ll} i, & 0 \leq t < T_i^s \\ j \neq i, & t \geq T_i^s, x(T_i^s) \in \Omega_j \end{array} \right\} \quad (4.7)$$

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

**Proof of Theorem 4.1** 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 Equation 4.7.

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

$$\dot{w}(T_i^f) - \dot{x}(T_i^f) = f(x_{T_i^f}) + g(x_{T_i^f})(u_i(x_{T_i^f}) + m_i(T_i^f)) - (f(w_{T_i^f}) + g(w_{T_i^f})u_i(w_{T_i^f})) \quad (4.8)$$

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

$$f(x_{T_i^f}) + g(x_{T_i^f})(u_i(x_{T_i^f}) + m_i(T_i^f)) - (f(w_{T_i^f}) + g(w_{T_i^f})u_i(w_{T_i^f})) = g(x_{T_i^f})m_i(T_i^f) \quad (4.9)$$

Furthermore, since  $g(x_{T_i^f}) \neq 0$ , we have that

$$\dot{w}(T_i^f) - \dot{x}(T_i^f) = g(x_{T_i^f})m_i(T_i^f) \neq 0 \quad (4.10)$$

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

$$w(T_i^{f+}) - x(T_i^{f+}) \neq 0 \quad (4.11)$$

or

$$r(T_i^{f+}) = \|w(T_i^{f+}) - x(T_i^{f+})\| > 0 \quad (4.12)$$

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 4.1.

The fault-detection filter and fault-tolerant controller are designed and implemented as follows (see also Figure 4.1):

- Given any  $x_0 \in \Omega_i$ , initialize the filter states as  $w(0) = x_0$  and integrate the filter dynamics using Equation 4.6.
- 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

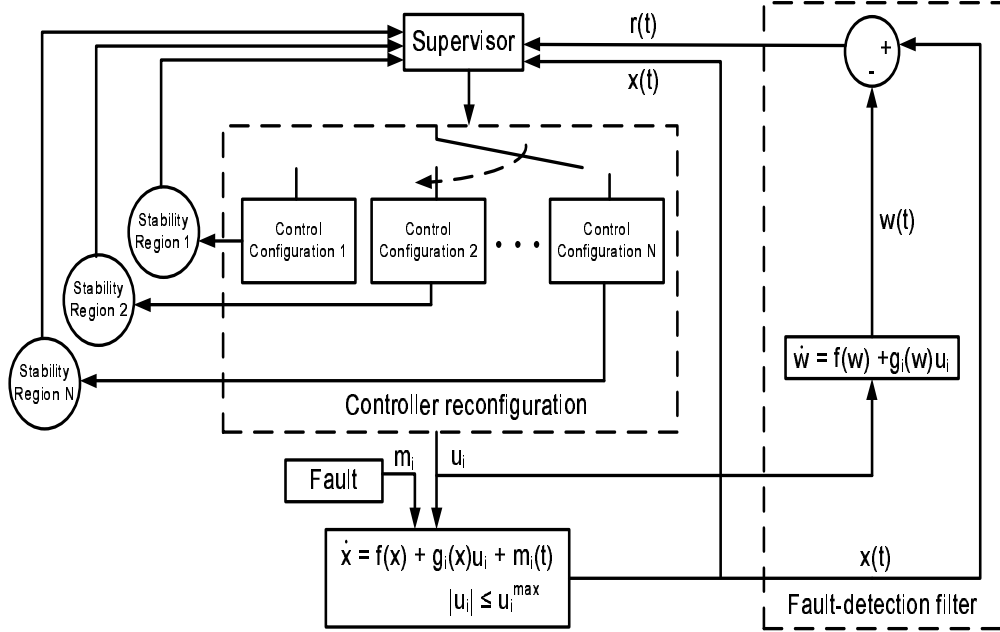


Figure 4.1: Integrated fault-detection and fault-tolerant control design: state feedback case.

system.

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.

**Remark 4.1** 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. Note that while the bounded controller is used for a demonstration of the main ideas, other control approaches, that provide an explicit characterization of the set of initial conditions for which closed-loop stability is guaranteed (achieved, for example, via the use of the hybrid predictive approach [54] or via a Lyapunov-based model predictive control design [115]) can be used within the proposed framework. Note also that early detection of a fault enhances the chances that corrective action can be taken in time to achieve fault-tolerant control (Theorem 4.1 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 the simulation example).

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 (Chapter 3). 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 config-

urations. 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 4.2** 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,  $\delta$ , where  $\delta$  accounts for the deviation of the plant dynamics from the nominal dynamics in the absence of faults (please see the simulation example for a demonstration of this idea in an application to a network of chemical reactors in the presence of uncertainty and measurement noise) 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 4.1 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 4.3** 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. For a detailed discussion and illustrative examples on integrated fault-detection and isolation and fault-tolerant control (FDIFTC) of nonlinear systems, please see [121] and [122].

**Remark 4.4** Note that the fault-detection filter presented in Theorem 4.1 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.

### 4.3.2 Simulation Results

In this section, we illustrate the implementation of the proposed fault-detection and fault-tolerant control methodology to the chemical reactor introduced as a motivating example. 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 Equation 2.13 into the following one describing the in-



put/output dynamics:

$$\dot{e} = Ae + l_k(e) + b\alpha_k u_k \quad := \quad \bar{f}_k(e) + \bar{g}_k(e)u_k \quad (4.13)$$

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 Equation 2.13 in the form of Equation 4.1. 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  $x = [T - T_s \ C_A - C_{As}]'$  with  $P = \begin{bmatrix} 0.011 & 0.019 \\ 0.019 & 0.101 \end{bmatrix}$ .

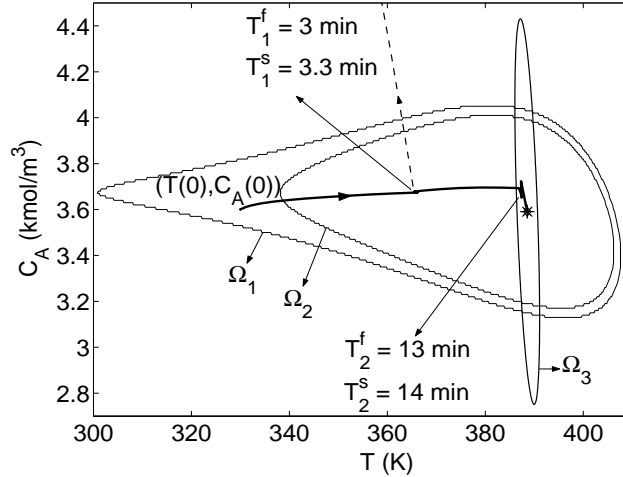
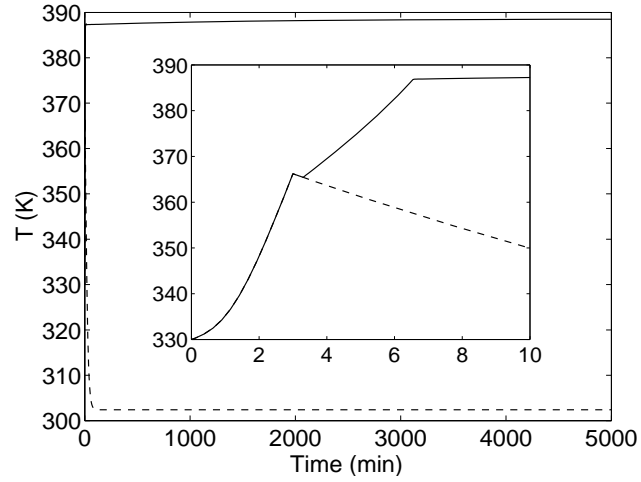


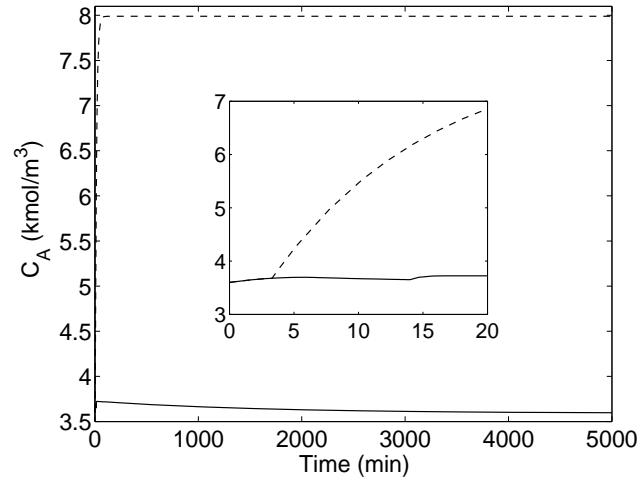
Figure 4.2: Evolution of the closed-loop state profiles under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid line) and under arbitrary switching (dashed line).

Figure 4.2 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 Figures 4.2-4.3, 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 Figure 4.5(a)). As can be seen in Figure 4.4(a), 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 Figures 4.2-4.3). 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 config-



(a)



(b)

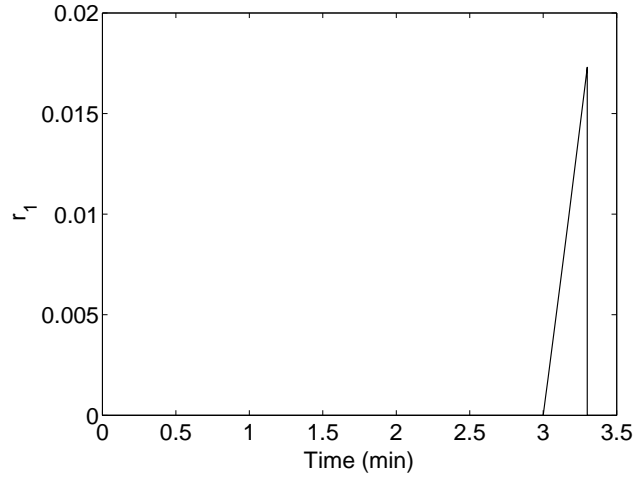
Figure 4.3: Evolution of the closed-loop (a) temperature and (b) concentration under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).

uration does not encounter a fault ( $r_3(t) = 0$ ; see dashed line in Figure 4.4(b)), the limited control action available in this configuration is unable to achieve closed-loop stability. On the basis of the switching logic of Equation 4.7, the supervisor activates the second configuration (with  $T_{A0}$  as the manipulated input, see Figure 4.5(b)), which continues to drive the state trajectory closer to the desired steady-state.

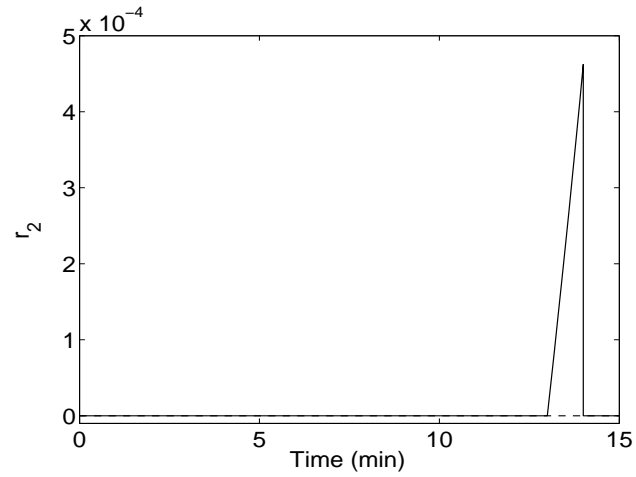
To demonstrate the implementation of the proposed FDFTC strategy when faults occur in successive control configurations, we consider the case when a second failure occurs (this time in the  $T_{A0}$ -configuration) at  $t = 13$  minutes. Once again, the filter detects this failure via an increase in the value of  $r_2(t)$  (solid line in Figure 4.4(b)) using the fault-detection filter for control configuration 2. From Figure 4.2, 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 Figure 4.5(c)) which finally stabilizes the reactor at the desired steady-state.

#### **4.4 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

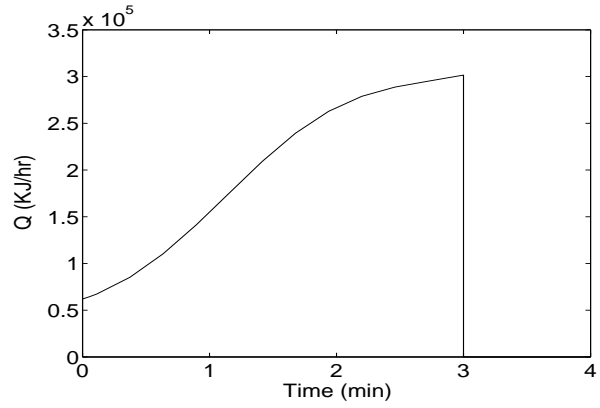


(a)

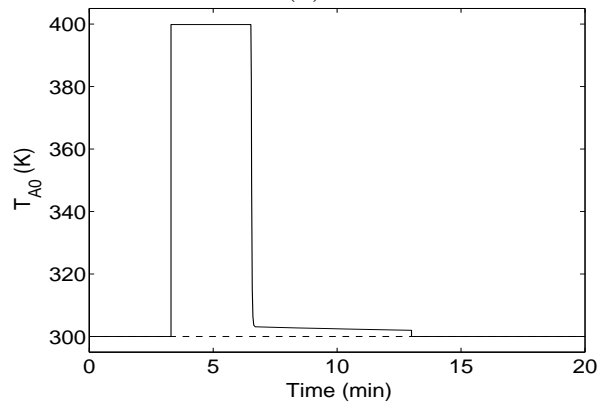


(b)

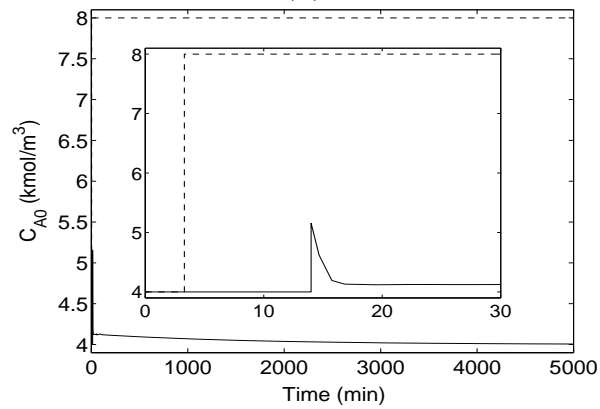
Figure 4.4: Evolution of the closed-loop residual under the fault-detection filter for (a) control configuration 1 and (b) control configurations 2 and 3 under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).



(a)



(b)



(c)

Figure 4.5: Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Equation 4.7 subject to failures in control systems 1 and 2 (solid lines) and under arbitrary switching (dashed lines).

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 [48], based on a combination of a high-gain observer and a state feedback controller (see also [106, 89, 90, 154, 27] 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.

#### 4.4.1 Output Feedback Control

To facilitate the design of a state estimator with the required convergence properties, we make the following assumption:

**Assumption 4.1** *For each  $i \in \mathcal{K}$ , there exists a set of coordinates*

$$\begin{bmatrix} \xi_i \end{bmatrix} = \begin{bmatrix} \xi_i^1 \\ \xi_i^2 \\ \vdots \\ \xi_i^n \end{bmatrix} = \chi_i(x) = \begin{bmatrix} h_m(x) \\ L_f h_m(x) \\ \vdots \\ L_f^{n-1} h_m(x) \end{bmatrix} \quad (4.14)$$

such that the system of Equation 4.1 takes the form

$$\begin{aligned}
\dot{\xi}_i^1 &= \xi_i^2 \\
&\vdots \\
\dot{\xi}_i^{n-1} &= \xi_i^n \\
\dot{\xi}_i^n &= L_f^n h_m(\chi_i^{-1}(\xi)) + L_{g_i} L_f^{n-1} h_m(\chi_i^{-1}(\xi))(u_{i(t)} + m_{i(t)})
\end{aligned} \tag{4.15}$$

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  $\xi_i$  is uniquely determined by the transformation  $\xi_i = \chi_i(x)$ . This implies that if one can estimate the values of  $\xi_i$  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 4.1 below presents the output feedback controller used for each mode and characterizes its stability properties. 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 Equation 4.4.

**Proposition 4.1** *Consider the nonlinear system of Equation 4.1, for a fixed mode,  $k(t) = i$ , and with  $m_i(t) \equiv 0$ , under the output feedback controller:*

$$\begin{aligned}
\dot{\tilde{y}} &= \begin{bmatrix} -L_i a_1^{(i)} & 1 & 0 & \cdots & 0 \\ -L_i^2 a_2^{(i)} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -L_i^n a_n^{(i)} & 0 & 0 & \cdots & 0 \end{bmatrix} \tilde{y} + \begin{bmatrix} L_i a_1^{(i)} \\ L_i^2 a_2^{(i)} \\ \vdots \\ L_i^n a_n^{(i)} \end{bmatrix} y_m \\
u_i &= u_i^c(\hat{x}, u_i^{max})
\end{aligned} \tag{4.16}$$



where  $u_i^c$  is defined in Equation 4.2, the parameters,  $a_1^{(i)}, \dots, a_n^{(i)}$  are chosen such that the polynomial  $s^n + a_1^{(i)}s^{n-1} + a_2^{(i)}s^{n-2} + \dots + a_n^{(i)} = 0$  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  $\beta_\zeta$  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  $\epsilon_i^{**}$  such that if  $\epsilon_i \in (0, \epsilon_i^{**}]$  then  $\|x(t) - \hat{x}(t)\| \leq e_{m,i}$  for all  $t \geq T_i^b$ .

**Proof of Proposition 4.1** The proof of the proposition, which invokes singular perturbation arguments (for a result on input-to-state stability with respect to singular perturbations, and further references, see [31]), is a special case of the proof of Theorem 4.2 in [48], and is omitted for brevity.

The state observer in Equation 4.16 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 [123, 141]). For such observers, however, it is difficult in general to obtain a transparent relationship between the tunable observer parameters and the error decay rate.

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 4.2 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 [114].

**Proposition 4.2** *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 4.2** 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  $\gamma_i$  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  $\gamma_i$  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 Proposition 4.2.

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 4.2, 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  $\Omega_i$ , and

furthermore, once the error has sufficiently decreased,  $\Omega_{s,i}$  can be taken to be almost equal to  $\Omega_{b,i}$ .

#### 4.4.2 Integrating Fault-Detection and Fault-Tolerant Output Feedback Control

In this section 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 Figure 4.6). 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:

$$\begin{aligned} \dot{w}(t) &= f(w) + g_i(w)u_i(w) \\ r(t) &= \|\hat{x}(t) - w(t)\| \end{aligned} \tag{4.17}$$

Note that, as in the full state feedback case, the state equation for the filter in Equation 4.17 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 4.3 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.

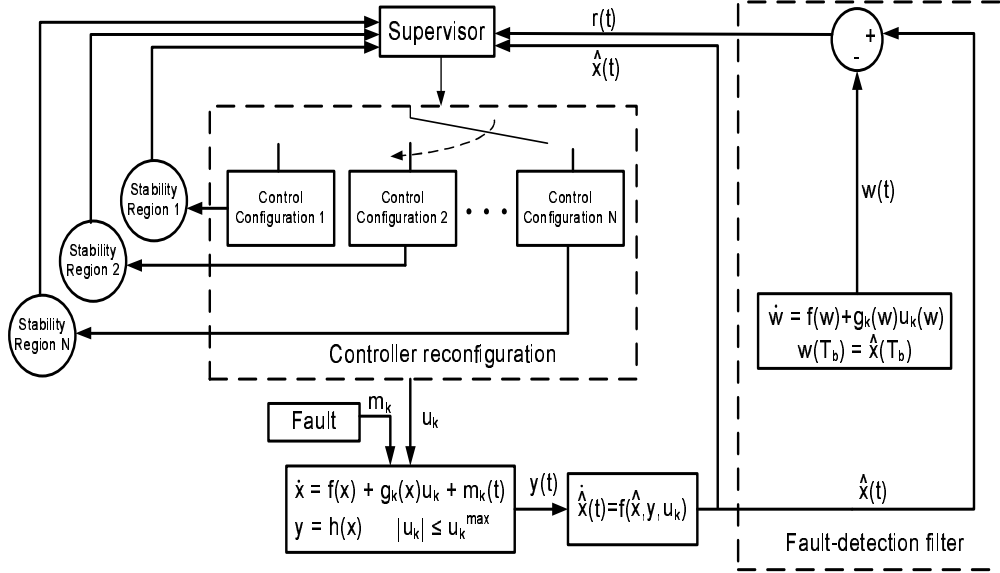


Figure 4.6: Integrated fault-detection and fault-tolerant control design under output feedback.

**Proposition 4.3** Consider the nonlinear system of Equation 4.1, for a fixed mode,  $k(t) = i$ , and with  $m_i(t) \equiv 0$ , under the output feedback controller of Equation 4.16. Consider also the system of Equation 4.17. 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 4.3** Consider the system of Equation 4.1 with  $m_i(t) \equiv 0$  under the output feedback controller of Equation 4.16. From the result of Proposition 4.1, we have that given  $x(0) \in \Omega_{b,i}$  and any positive real number  $T_i^b$ , there exists a real positive number  $\epsilon_i^{**}$  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  $t \geq T_i^b$ :

$$\dot{x}(t) = f(x(t)) + g_i(x(t))u_i(\hat{x}(t)) \quad (4.18)$$

$$\dot{w}(t) = f(w(t)) + g_i(w(t))u_i(w(t)) \quad (4.19)$$

where  $w(T_i^b) = \hat{x}(T_i^b)$ . The system of Equation 4.19 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  $\Omega_i$ . The system of Equation 4.18 is the closed-loop system under output feedback and (from Proposition 4.1) 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 Equations 4.18-4.19 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  $\eta$  is a continuous function that depends smoothly on  $\epsilon_i$ , then we can write

$$\begin{aligned}\dot{x}(t) &= F_i(x(t), \epsilon_i), & x(T_i^b) &= \eta(\epsilon_i) \\ \dot{w}(t) &= F_i(w(t)), & w(T_i^b) &= \eta(0)\end{aligned}\tag{4.20}$$

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 [91]) 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 [91] 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  $\epsilon_i' = \min\{\epsilon_i^{**}, \epsilon_i'', \delta_{m,i}/(k_1 + k_2)\}$ .

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 Proposition 4.3.

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 Equation 4.1 where, for each control configuration, an output feedback controller of the form of Equation 4.16 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 (for example,  $\epsilon_i \leq \min\{\epsilon_i^*, \epsilon_i', \epsilon_i^{**}\}$  to guarantee both stability and satisfaction of the desired bound on the residual) and the sets  $\Omega_{s,i}$  (see Proposition 4.2) have been computed. The implementation of the fault-detection filter and fault-tolerant controller is described in Theorem 4.2 below.

**Theorem 4.2** *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

$$k(t) = \left\{ \begin{array}{ll} i, & 0 \leq t < T_i^s \\ j \neq i, & t \geq T_i^s, \hat{x}(T_i^s) \in \Omega_j^s \end{array} \right\} \quad (4.21)$$

the origin of the closed-loop system is asymptotically stable.

**Proof of Theorem 4.2** Consider the nonlinear system of Equation 4.1, under the output feedback controller of Equation 4.16, and the system of Equation 4.17, 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  $\epsilon_i^*$ ,  $\epsilon_i^{**}$  were defined in Proposition 4.1 and  $\epsilon'_i$  was defined in Proposition 4.3. 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 4.1.

(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 Equation 4.21, that  $k(t) = i$  for  $0 \leq t < T_i^s$ . From the stability of the  $i$ -th closed-loop system established in Proposition 4.1, 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 4.1.

This completes the proof of Theorem 4.2.

The design and implementation of the fault-detection filter and fault-tolerant controller proceed as follows:

1. Given the nonlinear process of Equation 4.1, identify the available control configurations,  $k = 1, \dots, N$ . For each configuration, design the output feedback

controller of Equation 4.16, and for a given choice of the output feedback stability region,  $\Omega_{b,i}$ , determine a stabilizing observer gain,  $\epsilon_i^*$ .

2. Given any positive real numbers,  $\delta_{m,i}$  and  $T_i^b$ , determine the observer gain,  $\epsilon_i'$ , 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,  $\epsilon_i^{**}$ , 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 Equation 4.17 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 ini-



tialization 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.

Theorem 4.2 considers faults that are “observable” from the filter’s residual, in the sense that if the residual in Equation 4.17 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. 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 4.2 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).

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 4.5** 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 the simulation 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 4.2, 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 4.2 and the rule-based fault-detection filter discussed above differ only in that the model-based filter of Theorem 4.2 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 escape the stability region.

#### 4.4.3 Simulation Results

In this section, we first illustrate the implementation of the proposed fault-tolerant control methodology to the chemical reactor introduced as a motivating example to clearly explain the main ideas behind the application of the proposed fault-detection and fault-tolerant control method, and then demonstrate an application to a networked chemical reactor example, investigating issues such as uncertainty and measurement noise.

For the chemical reactor of the motivating example, Figure 4.10 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 Equation 4.16 is designed. For thresholds of  $\delta_m = 0.0172$  and  $0.00151$  in the fault detection filters, the parameters in the observer of Equation 4.16 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:

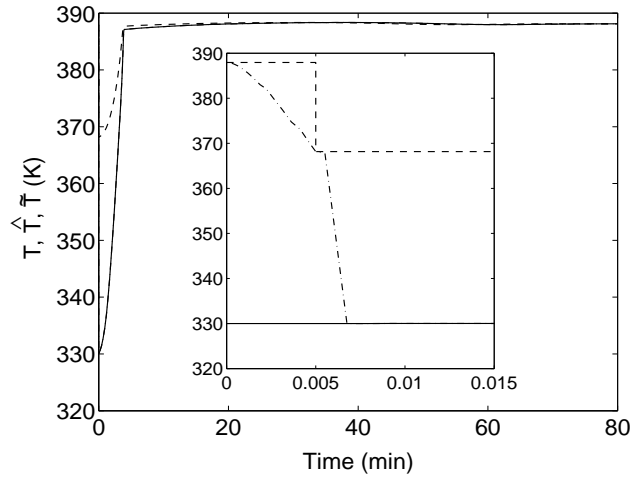
$$\begin{aligned} \frac{d\hat{T}}{dt} &= \frac{F}{V}(T_{A0} - \hat{T}) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} k_{i0} e^{\frac{-E_i}{R\hat{T}}} \hat{C}_A + \alpha_1(C_A - \hat{C}_A) \\ \frac{d\hat{C}_A}{dt} &= \frac{F}{V}(C_{A0} - \hat{C}_A) - \sum_{i=1}^3 k_{i0} e^{\frac{-E_i}{R\hat{T}}} \hat{C}_A + \alpha_2(C_A - \hat{C}_A) \end{aligned} \quad (4.22)$$

where  $\alpha_1 = -10^4$  and  $\alpha_2 = 10$  and  $C_A$  is the measured output. The reactor is

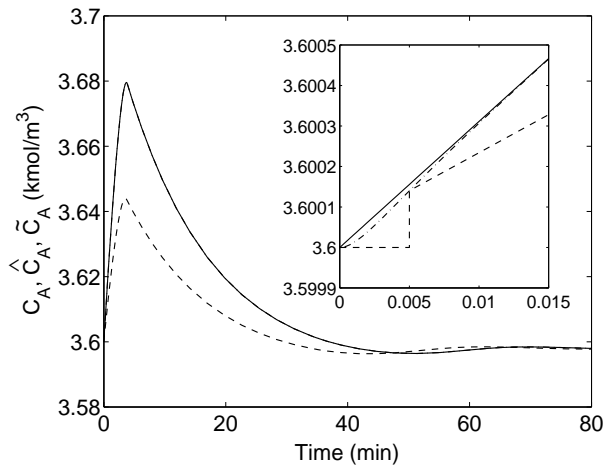
initialized at  $T(0) = 330 \text{ K}$ ,  $C_A(0) = 3.6 \text{ kmol/m}^3$ ,  $C_B(0) = 0.0 \text{ kmol/m}^3$ , using the  $Q$ -control configuration, while the state estimates are initialized at  $\hat{T}(0) = 390 \text{ K}$ ,  $\hat{C}_A(0) = 3.6 \text{ kmol/m}^3$  and the supervisor proceeds to monitor 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 Figure 4.7) have not converged to the true values (solid lines in Figure 4.7). As a result, the fault-detection filter shows a false alarm (see Figure 4.8(a)) by crossing the threshold even when control configuration 1 is functioning properly (see Figure 4.8(b)) 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.

We now demonstrate the application of the fault-detection filter and fault-tolerant controller of Theorem 4.2. Starting from the same initial conditions, the estimates of  $T$  and  $C_A$  (dash-dotted lines in Figures 4.9(a-b)) converge very quickly to the true values of the states (solid lines in Figures 4.9(a-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 Figure 4.11(a)). As shown by the solid lines in Figure 4.10, 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

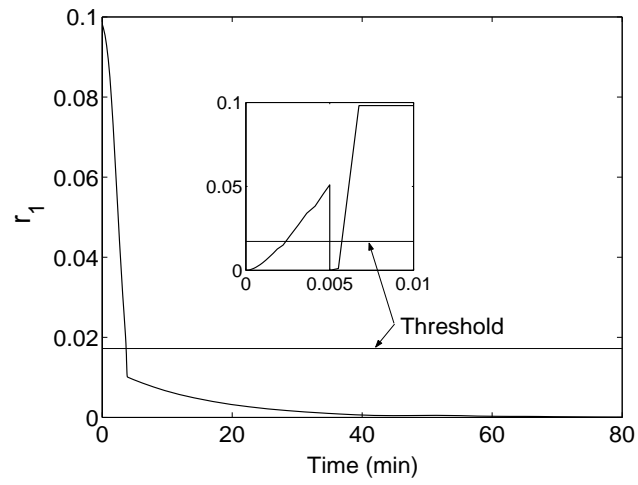


(a)

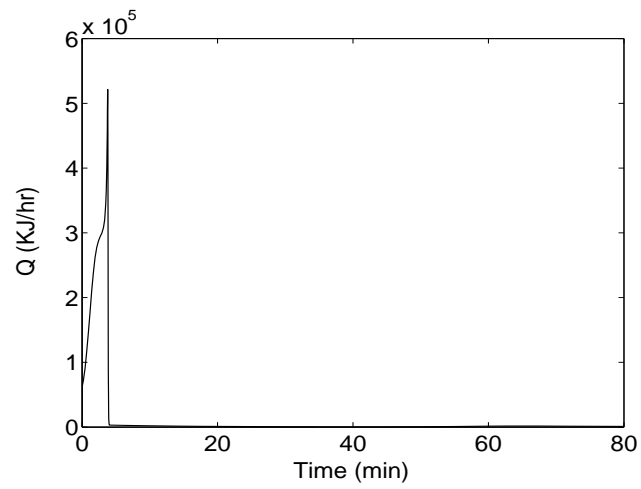


(b)

Figure 4.7: 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.



(a)



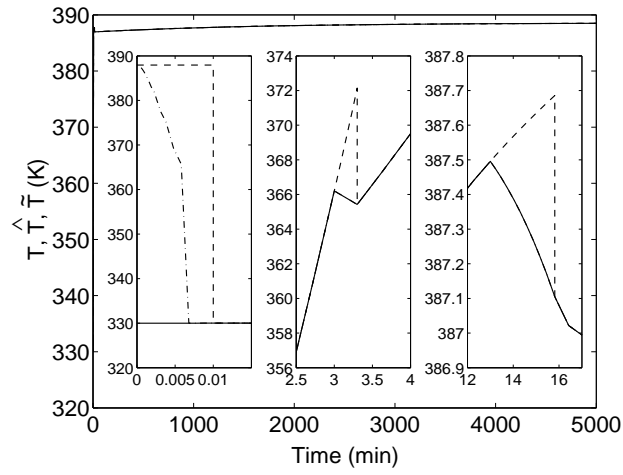
(b)

Figure 4.8: 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.

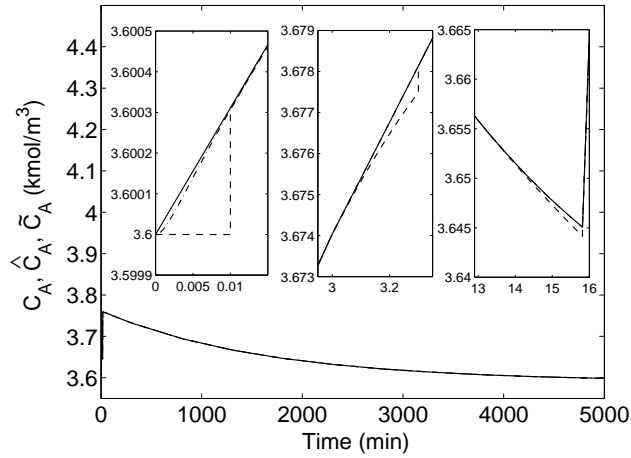
Figure 4.13(a)). Note that at this time, the value of  $r_1(t)$  becomes non-zero and hits the threshold at  $t = 3.3$  minutes  $\equiv T_1^s$ . From Figure 4.10, it is clear that the failure of the primary control configuration occurs when the closed-loop trajectory is within the stability region of the second control configuration, and outside the stability region of the third control configuration. Therefore, on the basis of the switching logic of Equation 4.21, the supervisor activates the second configuration (with  $T_{A0}$  as the manipulated input). The result is shown by the solid line in Figure 4.10 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 Figure 4.13(b)) before the process has reached the steady state, the filter detects this failure via the value of  $r_2(t)$  hitting the threshold (see Figure 4.11(b)). From the solid line in Figure 4.10, 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 Figure 4.10, see also Figure 4.12 and dashed lines in Figure 4.13). 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 Figure 4.10).

Having illustrated the application and effectiveness of the proposed fault-detection and fault-tolerant control method in the case of a single reactor, we next demonstrate



(a)



(b)

Figure 4.9: 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 Equation 4.21 subject to failures in control systems 1 and 2.



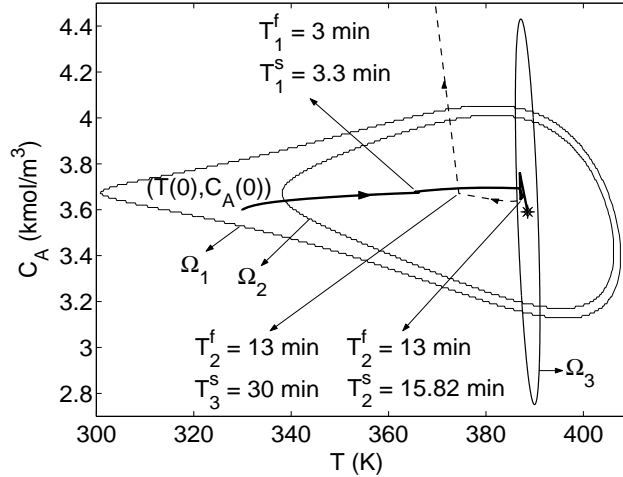
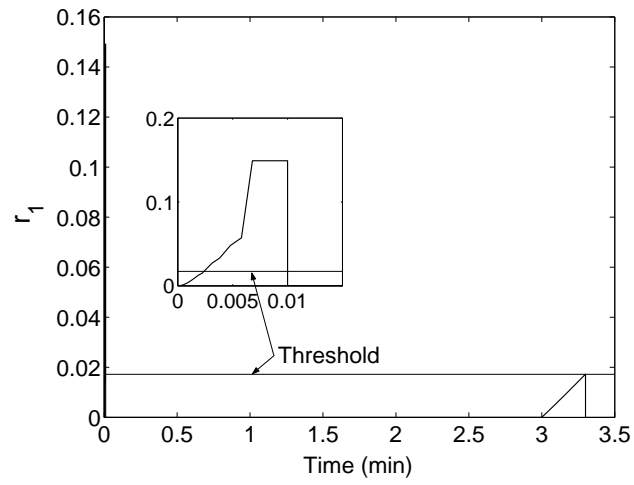
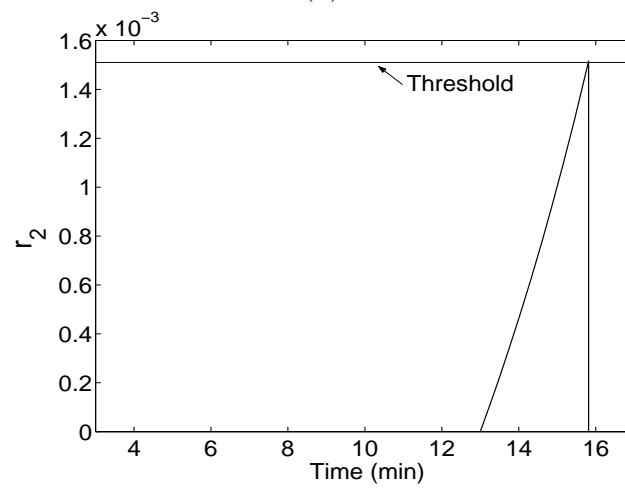


Figure 4.10: Evolution of the closed-loop state trajectory under the switching rule of Equation 4.21 subject to failures in control systems 1 and 2, using an appropriate fault-detection filter (solid line) and in the absence of a fault-detection filter (dashed line).

application of the method to a networked chemical reactor example in the presence of uncertainty and measurement noise. To this end, consider the two well-mixed, non-isothermal continuous stirred tank reactors shown in Figure 4.14. 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 in each reactor, where  $A$  is the reactant species,  $B$  is the desired product,  $U$  and  $R$  are undesired byproducts. The feed to the first reactor consists of pure  $A$  at a flow rate  $F_0$ , molar concentration  $C_{A0}$  and temperature  $T_0$ . The output from the first reactor is fed to the second reactor along with a fresh feed that consists of pure  $A$  at a flow rate  $F_3$ , molar concentration  $C_{A03}$ , and temperature  $T_{03}$ . Due to the non-isothermal nature of the reactors, a jacket is used to remove heat from or provide heat to the reactor. Under standard modeling assumptions, a mathematical model of the process can be derived from material and energy balances and takes the following

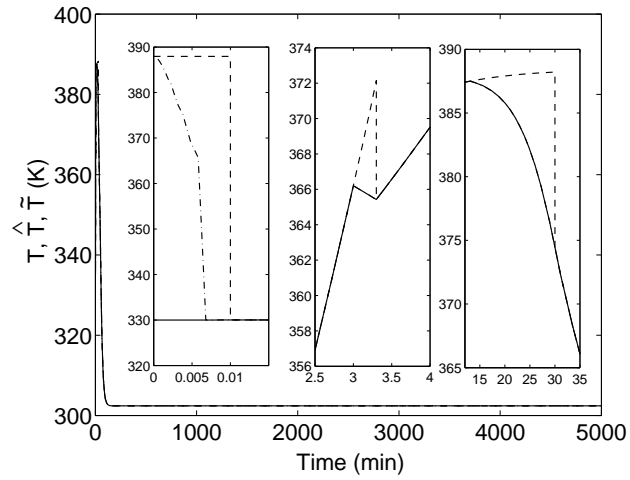


(a)

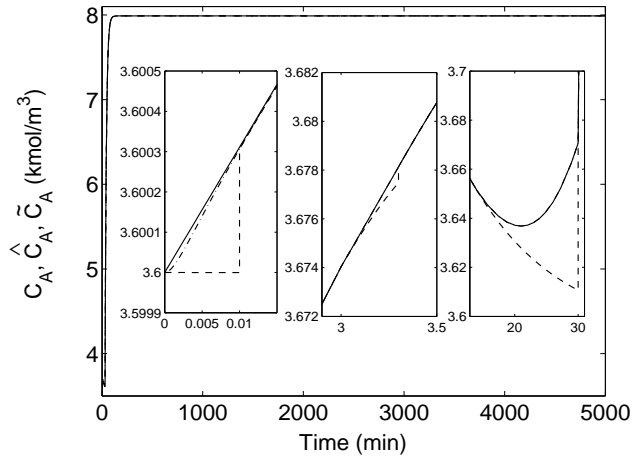


(b)

Figure 4.11: Evolution of the residual for (a) the first control configuration and (b) the second control configuration.

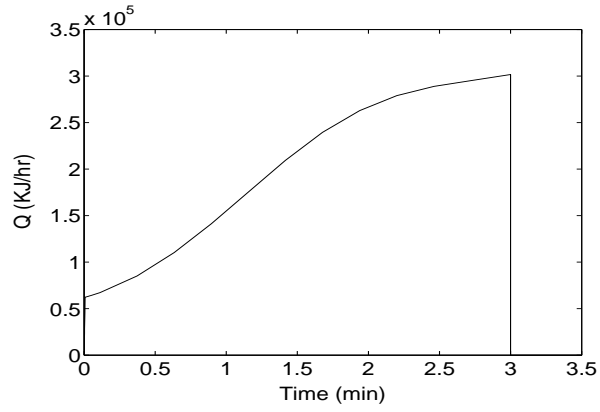


(a)

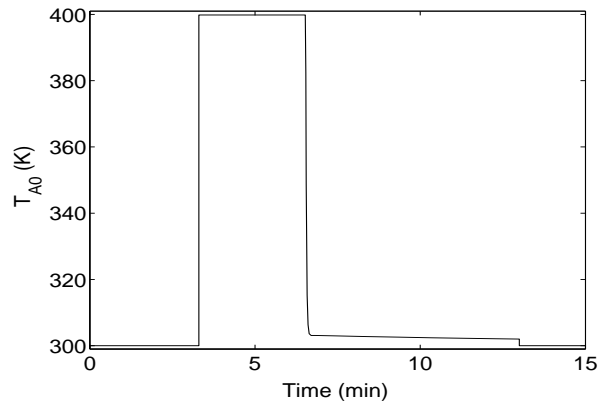


(b)

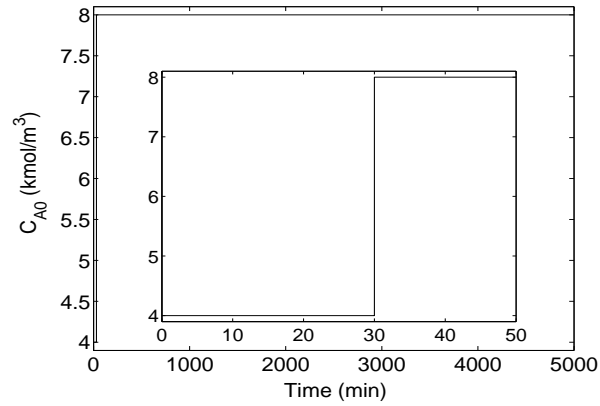
Figure 4.12: Evolution of the closed-loop (a) temperature (solid line), estimate of temperature (dash-dotted line) and the temperature profile generated by the filter (dashed line) and (b) concentration (solid line), estimate of concentration (dash-dotted line) and the concentration profile generated by the filter (dashed line) under the switching rule of Equation 4.21 subject to failures in control systems 1 and 2 in the absence of a fault-detection filter.



(a)



(b)



(c)

Figure 4.13: Manipulated input profiles under (a) control configuration 1, (b) control configuration 2, and (c) control configuration 3 under the switching rule of Equation 4.21 subject to failures in control systems 1 and 2 in the presence (solid lines) and absence (dashed lines) of a fault-detection filter.

form:

$$\begin{aligned}
\frac{dT_1}{dt} &= \frac{F_0}{V_1}(T_0 - T_1) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_{A1}, T_1) + \frac{Q_1}{\rho c_p V_1} \\
\frac{dC_{A1}}{dt} &= \frac{F_0}{V_1}(C_{A0} - C_{A1}) - \sum_{i=1}^3 R_i(C_{A1}, T_1) \\
\frac{dT_2}{dt} &= \frac{F_0}{V_2}(T_1 - T_2) + \frac{F_3}{V_2}(T_{03} - T_2) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_{A2}, T_2) + \frac{Q_2}{\rho c_p V_2} \\
\frac{dC_{A2}}{dt} &= \frac{F_0}{V_2}(C_{A1} - C_{A2}) + \frac{F_3}{V_2}(C_{A03} - C_{A2}) - \sum_{i=1}^3 R_i(C_{A2}, T_2)
\end{aligned} \tag{4.23}$$

where  $R_i(C_{Aj}, T_j) = k_{i0} \exp\left(\frac{-E_i}{RT_j}\right) C_{Aj}$ , for  $j = 1, 2$ .  $T$ ,  $C_A$ ,  $Q_i$  ( $i = 1, 2$ ), and  $V$  denote the temperature of the reactor, the concentration of species  $A$ , the rate of heat input/removal from the reactor, and the volume of reactor, respectively, with subscript 1 denoting CSTR 1 and subscript 2 denoting CSTR 2.  $\Delta H_i$ ,  $k_i$ ,  $E_i$ ,  $i = 1, 2, 3$ , denote the enthalpies, pre-exponential constants and activation energies of the three reactions, respectively,  $c_p$  and  $\rho$  denote the heat capacity and density of the fluid in the reactor. For the values of the process parameters given in Table 4.1 and for  $Q_1 = Q_2 = 0$  the process model of Equation 4.23 has multiple steady states.

The control objective is to stabilize at the open-loop unstable steady-state where  $(T_1^s, C_{A1}^s) = (388.57 \text{ K}, 3.59 \text{ kmol/m}^3)$  and  $(T_2^s, C_{A2}^s) = (433.96 \text{ K}, 2.8811 \text{ kmol/m}^3)$ . The measurements of temperature and concentrations are assumed to contain noise of magnitude  $1\text{K}$  and  $0.1 \text{ kmol/m}^3$ , respectively. Also, the concentrations of  $A$  in the inlet streams  $C_{A0}$  and  $C_{A03}$  used in the process model are 10% smaller than the values used in the filter equations and the controller. The available manipulated inputs include the rate of heat input into reactor one,  $Q_1$ , subject to the constraint  $|Q_1| \leq 2.333(10^6) \text{ kJ/hr}$ , the rate of heat input into reactor two,  $Q_2$ , subject to the constraint  $|Q_2| \leq 1.167(10^6) \text{ kJ/hr}$  and a duplicate backup heating configuration for reactor two,  $Q_3$ , subject to the constraint  $|Q_3| \leq 1.167(10^6) \text{ kJ/hr}$ .

The primary control configuration consists of the manipulated inputs  $Q_1$  and  $Q_2$ ,

Table 4.1: Process parameters and steady-state values for the chemical reactors of Equation 4.23.

$F_0$	=	4.998	$m^3/hr$
$F_1$	=	4.998	$m^3/hr$
$F_3$	=	4.998	$m^3/hr$
$V_1$	=	1.0	$m^3$
$V_2$	=	0.5	$m^3$
$R$	=	8.314	$KJ/kmol \cdot K$
$T_0$	=	300.0	$K$
$T_{03}$	=	300.0	$K$
$C_{A0}$	=	4.0	$kmol/m^3$
$C_{A03}^s$	=	3.0	$kmol/m^3$
$\Delta H_1$	=	$-5.0 \times 10^4$	$KJ/kmol$
$\Delta H_2$	=	$-5.2 \times 10^4$	$KJ/kmol$
$\Delta H_3$	=	$-5.4 \times 10^4$	$KJ/kmol$
$k_{10}$	=	$3.0 \times 10^6$	$hr^{-1}$
$k_{20}$	=	$3.0 \times 10^5$	$hr^{-1}$
$k_{30}$	=	$3.0 \times 10^5$	$hr^{-1}$
$E_1$	=	$5.0 \times 10^4$	$KJ/kmol$
$E_2$	=	$7.53 \times 10^4$	$KJ/kmol$
$E_3$	=	$7.53 \times 10^4$	$KJ/kmol$
$\rho$	=	1000.0	$kg/m^3$
$c_p$	=	0.231	$KJ/kg \cdot K$
$T_1^s$	=	388.57	$K$
$C_{A1}^s$	=	3.59	$kmol/m^3$
$T_2^s$	=	433.96	$K$
$C_{A2}^s$	=	2.88	$kmol/m^3$

while the backup configuration is comprised of manipulated inputs  $Q_1$  and  $Q_3$ . As before, quadratic Lyapunov functions of the form  $V_k = x^T P_k x$  are used for controller design, where  $P_k$  is a positive-definite symmetric matrix that satisfies the Riccati inequality  $A^T P_k + P_k A - P_k b_k b_k^T P_k < 0$  for  $A$  and  $b$  obtained via linearization of the system around the desired steady-state with  $x = [T_1 - T_{1s} \ C_{A1} - C_{A1s} \ T_2 - T_{2s} \ C_{A2} - C_{A2s}]'$ , and are not reported here for the sake of brevity. The controller design yields a stability region estimate with  $c_1^{max}$  and  $c_2^{max}$  both approximately equal to 9.4. Note that all the information about the stability region is completely contained in the values of  $c_1^{max}$  and  $c_2^{max}$ , and the computation of these values is sufficient for the task of implementing the proposed method to the four-state system in this example. Specifically, the presence of the closed-loop state in the stability region can be ascertained by simply evaluating the value of the Lyapunov-function and checking against the value of  $c^{max}$  (for example,  $V(x) < c_1^{max}$  implies that  $x \in \Omega_1$ ).

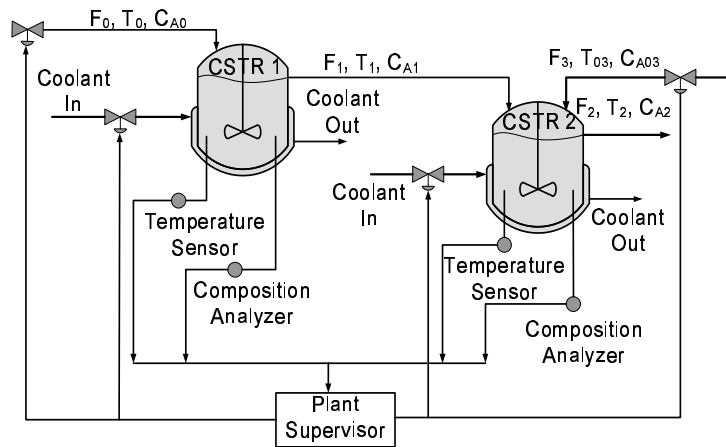


Figure 4.14: Flow diagram showing two CSTRs operating in series.

Note that unlike the single reactor example, each control configuration consists of more than one manipulated input, which necessitates designing filters that detect as

well as *isolate* faults. To this end, fault detection and isolation filters are designed that are dedicated to each manipulated input in the control configurations. The filter designs for  $Q_1$  and  $Q_2$  in the primary control configuration take the form:

$$\begin{aligned}\frac{d\tilde{T}_1}{dt} &= \frac{F_0}{V_1}(T_0 - \tilde{T}_1) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_{A1}, \tilde{T}_1) + \frac{Q_1}{\rho c_p V_1} \\ r_1 &= \tilde{T}_1 - T_1\end{aligned}\quad (4.24)$$

$$\begin{aligned}\frac{d\tilde{T}_2}{dt} &= \frac{F_0}{V_2}(T_1 - \tilde{T}_2) + \frac{F_3}{V_2}(T_{03} - \tilde{T}_2) + \sum_{i=1}^3 \frac{(-\Delta H_i)}{\rho c_p} R_i(C_{A2}, \tilde{T}_2) + \frac{Q_2}{\rho c_p V_2} \\ r_2 &= \tilde{T}_2 - T_2\end{aligned}\quad (4.25)$$

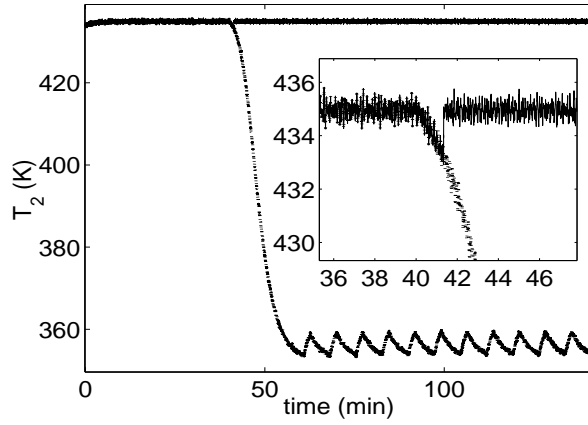
As can be seen, the fault-detection and isolation filter for  $Q_1$  includes a state  $\tilde{T}_1$  whose dynamics are a copy of the model state, however, the dynamics are evaluated using the state measurements together with using  $\tilde{T}_1$  in place of  $T_1$ . The value of the manipulated variable is also calculated in the same manner. For example,  $Q_1$  in the filter is computed using  $(\tilde{T}_1, C_{A1}, T_2, C_{A2})$ . The filters for the other manipulated inputs are designed similarly. Note that due to the presence of measurement noise and disturbances, the values of the residual are non-zero even in the absence of faults, therefore, faults are declared only if the value of the residual exceeds a non-zero threshold value, where the threshold is obtained by evaluating the maximum value of the residual in the *absence* of faults to account for the effects of uncertainty and measurement noise.

In the first scenario the ability to detect a fault in the presence of multiple disturbances and noise is demonstrated. The reactors as well as the fault detection filter for the first control configuration are initialized at the desired steady state  $T_1(0) = 388.57\text{ K}$ ,  $C_{A1}(0) = 3.591\text{ kmol/m}^3$ ,  $T_2(0) = 433.96\text{ K}$  and  $C_{A2}(0) = 2.881\text{ kmol/m}^3$ . For the sake of brevity, we show here only the evolution of  $T_2$  and of the residuals. As can be seen in Figure 4.15(a), the controller proceeds to stabilize the closed-loop

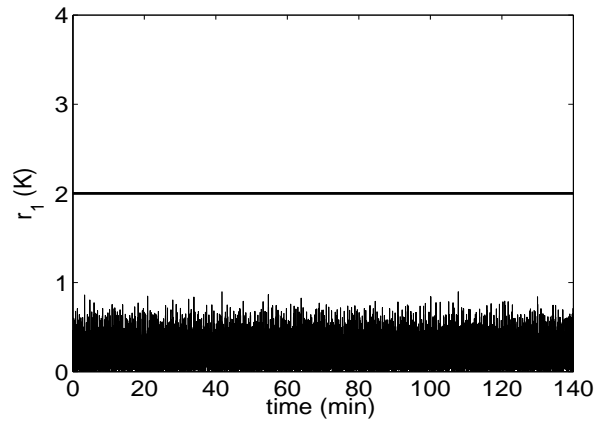


trajectory near the desired steady-state until heating jacket two ( $Q_2$ ) fails 40 minutes after reactor startup. If a fault-detection filter is not in place, and the fault is not detected, closed-loop stability is not achieved (dotted lines in Figure 4.15(a)). The fault-detection filter design of the form of Equations 4.24-4.25, however, detects this fault, and the value of residual  $r_2(t)$  becomes greater than the threshold value of 2.0 at 40.79 minutes (see Figure 4.15(c)) while  $r_1(t)$  (Figure 4.15(b)) remains below the threshold of 2.0, allowing the detection and isolation of the fault. While at the time of the failure ( $t = 40$  min), the state of the closed-loop system is within the stability region of the backup-configuration, by the time that the failure is detected (at  $t = 40.79$  min), operation of reactor 2 in an open-loop fashion (for 0.79 min) results in the closed-loop state moving out of the stability region of the backup configuration ( $V_2 = 73.17 > c_2^{max} = 9.4$ ) and stability is not guaranteed after switching. However, it is possible that stability may still be achieved by using the fall-back configuration. In particular, having been alerted by the fault-detection filter of the occurrence of the fault, the supervisor activates the fall-back configuration (with  $Q_1$  and  $Q_3$  as the manipulated inputs, solid lines in Figure 4.15(a)) and is able to drive the system to the desired steady state and enforce closed-loop stability.

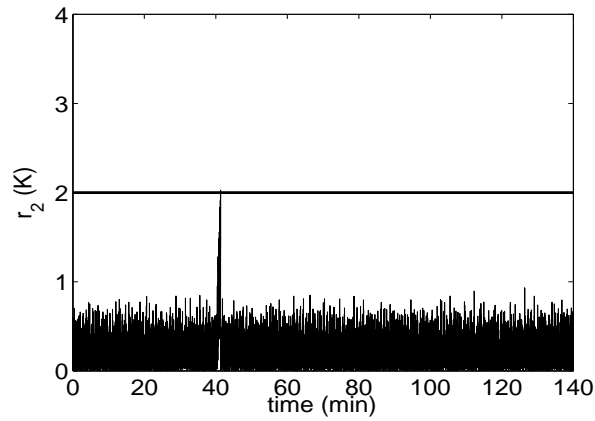
Detection of faults in the presence of process disturbances and noise is clearly possible using the methodology above. In order to guarantee stability after switching, however, the disturbances acting on the system should be reduced or the constraints on the control action should be relaxed to enlarge the closed-loop stability region. In the second scenario, the ability to detect a fault in the presence of noise and a single disturbance (in contrast to two disturbances in the first scenario), then switch to a fall-back configuration with guaranteed stability is demonstrated. In this case, the measurements of temperature and concentrations are again assumed to contain noise



(a)

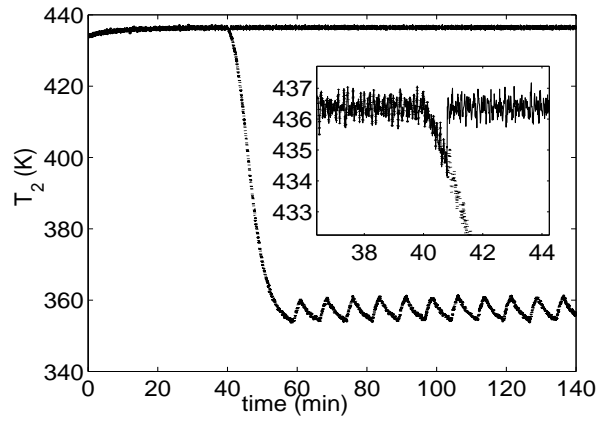


(b)

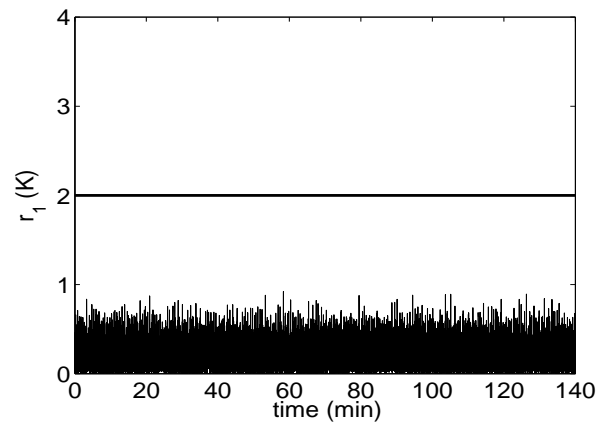


(c)

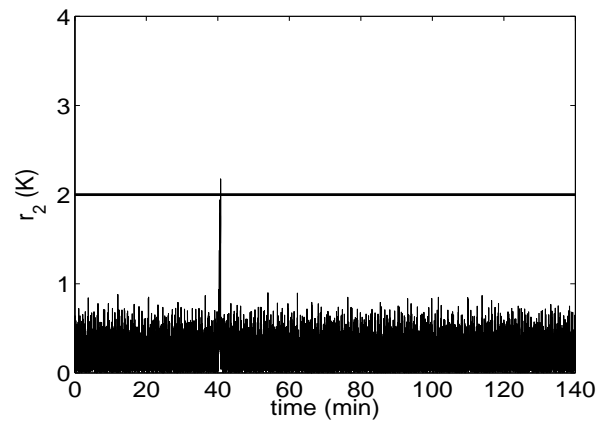
Figure 4.15: Two reactors in series scenario one: (a) temperature profile of reactor two with reconfiguration (solid line) and without reconfiguration (dotted line), (b)  $Q_1$  residual profile, (c)  $Q_2$  residual profile (note fault detection at time  $t = 40.79$  min).



(a)



(b)



(a)

Figure 4.16: Two reactors in series scenario two: (a) temperature profile of reactor two with reconfiguration (solid line) and without reconfiguration (dotted line), (b)  $Q_1$  residual profile, (c)  $Q_2$  residual profile (note fault detection at time  $t = 41.33$  min).

of magnitude  $1K$  and  $0.1 \text{ kmol}/m^3$ , respectively. Also, the concentration of  $A$  in the inlet stream  $C_{A03}$  used in the process model is 10% smaller than the values used in the filter equations and the controller.

The reactors as well as the fault detection filter for the first control configuration are initialized at the desired steady state  $T_1(0) = 388.57 \text{ K}$ ,  $C_{A1}(0) = 3.591 \text{ kmol}/m^3$ ,  $T_2(0) = 433.96 \text{ K}$ ,  $C_{A2}(0) = 2.881 \text{ kmol}/m^3$ . As can be seen in Figure 4.16(a), the controller proceeds to stabilize the closed-loop trajectory near the desired steady-state until heating jacket two ( $Q_2$ ) fails 40 minutes after reactor startup. If a fault-detection filter is not in place, and the fault is not detected, closed-loop stability is not achieved (dotted lines in Figure 4.16(a)). The fault-detection filter design of the form of Equations 4.24-4.25, however, detects this fault, and the value of residual  $r_2(t)$  becomes greater than the threshold value of 2.0 at 41.33 minutes (see Figure 4.16(c)) while  $r_1(t)$  (Figure 4.16(b)) remains below the threshold of 2.0, allowing the detection and isolation of the fault. In this scenario, at the time of the failure and by the time that the fault is detected, the state of the closed-loop system resides within the stability region of configuration two ( $V_2 = 8.03 < c_2^{max} = 9.4$ ). Therefore, the supervisor activates the fall-back configuration (with  $Q_1$  and  $Q_3$  as the manipulated inputs, solid lines in Figure 4.16(a)) and the control system is able to drive the process to the desired steady state and enforce closed-loop stability.

## 4.5 Conclusions

In this chapter, we presented an integrated fault-detection and fault-tolerant control (FDFTC) structure, for nonlinear processes with input constraints subject to control actuator failures. 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 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 illustrate the main ideas behind the proposed method as well as to successfully demonstrate an application in the presence of uncertainty and measurement noise.

## Chapter 5

# Fault-Tolerant Control of a Polyethylene Reactor

### 5.1 Introduction

Industrial processes stand to gain from an application of fault-tolerant control structures that prevent loss of product (due, for example, to limit cycles) and possible loss of equipment (due, for example, to unacceptably high temperatures) in the event of a fault in the control configuration, while accounting explicitly for the complex process characteristics manifested in the form of nonlinearities, constraints, and uncertainty. One of the prerequisites in implementing fault-tolerant control is the ability to detect and isolate the occurrence of faults. 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 (for example, [96, 145, 131, 44, 126, 43, 35, 156, 4, 187]) 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 ex-

tensively in the context of linear systems [108, 60, 61, 40, 112] and more recently some existential results in the context of nonlinear systems have been derived [146, 37].

This chapter focuses on fault-detection and fault-tolerant control of an industrial gas phase polyethylene reactor modeled by seven nonlinear ordinary differential equations (ODEs). Polyethylene is the most popular of all synthetic commodity polymers, with current worldwide production of more than 40 billion tonnes per year. Large proportion of this polyethylene is produced in gas phase reactors using Ziegler-Natta catalysts. In a gas phase polyethylene reactor, the temperature in the reaction zone is kept above the dew point of the reactant and below the melting point of the polymer to prevent melting and consequent agglomeration of the product particles. Most commercial gas phase fluidized bed polyethylene reactors are operated in a relatively narrow temperature range between  $75^{\circ}C$  and  $110^{\circ}C$  [174]. It has been demonstrated [26, 111, 82] that without feedback temperature control (or in the event of failure in the control configuration), industrial gas phase polyethylene reactors are prone to unstable steady-states, limit cycles, and excursions toward unacceptable high temperature steady-states which can lead to loss of product as well as damage the equipment.

To develop a fault-tolerant control system for the gas phase polyethylene reactor [64], we initially describe the process evolution on the basis of a detailed model and identify a family of candidate control configurations. For each control configuration, a bounded nonlinear feedback 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 using Lyapunov-based tools. Next, a fault-detection filter is designed to detect the occurrence of a fault in the control actuator by observing the deviation of the process states from the expected closed-loop behavior. 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 of control system faults. Closed-loop system simulations demonstrate the effectiveness of the fault-tolerant control strategy as well as investigate an application in the presence of measurement noise.

## 5.2 Process Description and Modeling

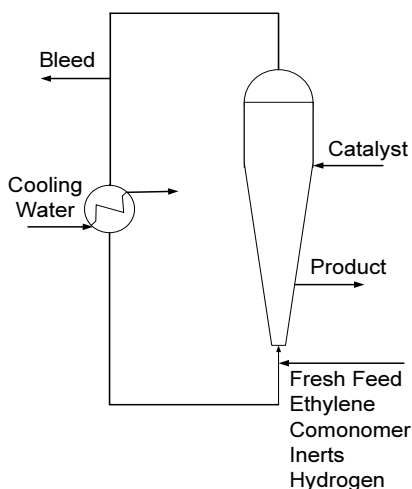


Figure 5.1: Industrial gas phase polyethylene reactor system.

Figure 5.1 shows a schematic of an industrial gas phase polyethylene reactor system. The feed to the reactor consists of ethylene, comonomer, hydrogen, inerts, and catalyst. A stream of unreacted gases flows from the top of the reactor and is cooled by passing through a heat exchanger in counter-current flow with cooling water. Cooling rates in the heat exchanger are adjusted by instantaneously blending cold and warm water streams while maintaining a constant total cooling water flowrate through the heat exchanger. Mass balance on hydrogen and comonomer have not been considered in this study because hydrogen and comonomer have only mild effects on the reactor



dynamics [111]. A mathematical model for this reactor has the form [32]:

$$\begin{aligned}
\frac{d[In]}{dt} &= \frac{F_{In} - \frac{[In]}{[M_1] + [In]} b_t}{V_g} \\
\frac{d[M_1]}{dt} &= \frac{F_{M_1} - \frac{[M_1]}{[M_1] + [In]} b_t - R_{M_1}}{V_g} \\
\frac{dY_1}{dt} &= F_c a_c - k_{d_1} Y_1 - \frac{R_{M_1} M_{W_1} Y_1}{B_w} \\
\frac{dY_2}{dt} &= F_c a_c - k_{d_2} Y_2 - \frac{R_{M_1} M_{W_1} Y_2}{B_w} \\
\frac{dT}{dt} &= \frac{H_f + H_{g_1} - H_{g_0} - H_r - H_{pol}}{M_r C_{pr} + B_w C_{ppol}} \\
\frac{dT_{w_1}}{dt} &= \frac{F_w}{M_w} (T_{w_i} - T_{w_1}) - \frac{UA}{M_w C_{pw}} (T_{w_1} - T_{g_1}) \\
\frac{dT_{g_1}}{dt} &= \frac{F_g}{M_g} (T - T_{g_1}) + \frac{UA}{M_g C_{pg}} (T_{w_1} - T_{g_1})
\end{aligned} \tag{5.1}$$

where

$$\begin{aligned}
b_t &= V_p C_v \sqrt{([M_1] + [In]) \cdot RR \cdot T - P_v} \\
R_{M_1} &= [M_1] \cdot k_{p0} \cdot \exp\left[\frac{-E_a}{R} \left(\frac{1}{T} - \frac{1}{T_f}\right)\right] \cdot (Y_1 + Y_2) \\
C_{pg} &= \frac{[M_1]}{[M_1] + [In]} C_{pm1} + \frac{[In]}{[M_1] + [In]} C_{pIn} \\
H_f &= F_{M_1} C_{pm1} (T_{feed} - T_f) + F_{In} C_{pIn} (T_{feed} - T_f) \\
H_{g_1} &= F_g (T_{g_1} - T_f) C_{pg} \\
H_{g_0} &= (F_g + b_t) (T - T_f) C_{pg} \\
H_r &= H_{reac} M_{W_1} R_{M_1} \\
H_{pol} &= C_{ppol} (T - T_f) R_{M_1} M_{W_1}
\end{aligned} \tag{5.2}$$

Table 5.1 includes the definition of all the variables used in Equations 5.1-5.2. The values of the process parameters are listed in Table 5.2. Under these operating conditions, the open-loop system behaves in an oscillatory fashion (i.e., the system possesses an open-loop unstable steady-state surrounded by a limit cycle).

Table 5.1: Process variables.

$a_c$	active site concentration of catalyst
$b_t$	overhead gas bleed
$B_w$	mass of polymer in the fluidized bed
$C_{pm1}$	specific heat capacity of ethylene
$C_v$	vent flow coefficient
$C_{pw}, C_{pIn}, C_{ppol}$	specific heat capacity of water, inert gas and polymer
$E_a$	activation energy
$F_c, F_g$	flow rate of catalyst and recycle gas
$F_{In}, F_{M_1}, F_w$	flow rate of inert, ethylene and cooling water
$H_f$	enthalpy of fresh feed stream
$H_{g0}$	enthalpy of total gas outflow stream from reactor
$H_{g1}$	enthalpy of cooled recycle gas stream to reactor
$H_{pol}$	enthalpy of polymer
$H_r$	heat liberated by polymerization reaction
$H_{reac}$	heat of reaction
$[In]$	molar concentration of inerts in the gas phase
$k_{d1}, k_{d2}$	deactivation rate constant for catalyst site 1, 2
$k_{p0}$	pre-exponential factor for polymer propagation rate
$[M_1]$	molar concentration of ethylene in the gas phase
$M_g$	mass holdup of gas stream in heat exchanger
$M_r C_{pr}$	product of mass and heat capacity of reactor walls
$M_w$	mass holdup of cooling water in heat exchanger
$M_{W_1}$	molecular weight of monomer
$P_v$	pressure downstream of bleed vent
$R$	ideal gas constant, unit of $\frac{J}{mgl \cdot K}$
$RR$	ideal gas constant, unit of $\frac{m^3 \cdot atm}{mol \cdot K}$
$T$	reactor temperature
$T_f$	reference temperature
$T_{feed}$	feed temperature
$T_{g1}$	temperature of recycle gas stream from exchanger
$T_{w1}$	temperature of cooling water stream from exchanger
$T_{wi}$	inlet cooling water temperature to heat exchanger
$UA$	product of heat exchanger coefficient with area
$V_g$	volume of gas phase in the reactor
$V_p$	bleed stream valve position
$Y_1, Y_2$	moles of active site type 1, 2

Table 5.2: Parameter values and units.

$V_g$	=	500	$m^3$
$V_p$	=	0.5	
$P_v$	=	17	<i>atm</i>
$B_w$	=	$7 \cdot 10^4$	<i>kg</i>
$k_{p0}$	=	$85 \cdot 10^{-3}$	$\frac{m^3}{mol \cdot s}$
$E_a$	=	(9000)(4.1868)	$\frac{J}{mol}$
$C_{pm1}$	=	(11)(4.1868)	$\frac{mol \cdot K}{mol \cdot K}$
$C_v$	=	7.5	$atm^{-0.5} \frac{mol}{s}$
$C_{pw}$	=	( $10^3$ )(4.1868)	$\frac{J}{kg \cdot K}$
$C_{pIn}$	=	(6.9)(4.1868)	$\frac{J}{mol \cdot K}$
$C_{ppol}$	=	( $0.85 \cdot 10^3$ )(4.1868)	$\frac{kg \cdot K}{kg \cdot K}$
$k_{d1}$	=	0.0001	$s^{-1}$
$k_{d2}$	=	0.0001	$s^{-1}$
$M_{W1}$	=	$28.05 \cdot 10^{-3}$	$\frac{kg}{mol}$
$M_w$	=	$3.314 \cdot 10^4$	<i>kg</i>
$M_g$	=	6060.5	<i>mol</i>
$M_r C_{pr}$	=	( $1.4 \cdot 10^7$ )(4.1868)	$\frac{J}{K}$
$H_{reac}$	=	( $-894 \cdot 10^3$ )(4.1868)	$\frac{J}{kg}$
$UA$	=	( $1.14 \cdot 10^6$ )(4.1868)	$\frac{J}{K \cdot s}$
$F_{In}$	=	5	$\frac{s}{mol}$
$F_{M1}$	=	190	$\frac{s}{mol}$
$F_g$	=	8500	$\frac{s}{mol}$
$F_w$	=	( $3.11 \cdot 10^5$ )( $18 \cdot 10^{-3}$ )	$\frac{kg}{s}$
$F_c^s$	=	$\frac{5.8}{3600}$	$\frac{kg}{s}$
$T_f$	=	360	<i>K</i>
$T_{feed}^s$	=	293	<i>K</i>
$T_{wi}$	=	289.56	<i>K</i>
$RR$	=	$8.20575 \cdot 10^{-5}$	$\frac{m^3 \cdot atm}{mol \cdot K}$
$R$	=	8.314	$\frac{mol \cdot K}{mol \cdot K}$
$a_c$	=	0.548	$\frac{kg}{K}$
$u_1^{max}$	=	$5.78 \cdot 10^{-4}$	$\frac{s}{mol}$
$u_2^{max}$	=	$3.04 \cdot 10^{-4}$	$\frac{s}{mol}$
$[In]_s$	=	439.68	$\frac{mol}{m^3}$
$[M1]_s$	=	326.72	$\frac{mol}{m^3}$
$Y_{1s}, Y_{2s}$	=	3.835, 3.835	<i>mol</i>
$T_s$	=	356.21	<i>K</i>
$T_{w1s}$	=	290.37	<i>K</i>
$T_{g1s}$	=	294.36	<i>K</i>

The control objective is to stabilize the reactor. To accomplish this objective we consider the following manipulated input candidates:

1. Feed temperature,  $u_1 = \frac{F_{M_1}C_{pm1}+F_{In}C_{pIn}}{M_rC_{pr}+B_wC_{ppol}}(T_{feed} - T_{feed}^s)$ , subject to the constraint  $|u_1| \leq u_{max}^1 = \frac{F_{M_1}C_{pm1}+F_{In}C_{pIn}}{M_rC_{pr}+B_wC_{ppol}}(20) \frac{K}{s}$ .
2. Catalyst flowrate,  $u_2 = (F_c - F_c^s)a_c$ , subject to the constraint  $|u_2| \leq u_{max}^2 = (\frac{2}{3600})a_c \frac{mol}{s}$ .

Each of the above manipulated inputs represents a unique control configuration (or control loop) that, by itself, can stabilize the reactor. The first control configuration, with feed temperature ( $T_{feed}$ ) as the manipulated input, will be considered as the primary configuration. In the event of some faults in this configuration, however, the plant supervisor, will have to activate the fall-back configuration in order to maintain closed-loop stability. The question which we address in the next section, is how the supervisor determines, from observing the evolution of the process, that a fault has occurred in the control configuration and whether or not the fall-back control configuration will be able to stabilize the reactor if the primary control configuration fails.

### 5.3 Fault-Tolerant Control

Having identified the candidate control configurations that can be used, we outline in this section the main steps involved in the fault-tolerant control system design procedure. These include: (a) the synthesis of a stabilizing feedback controller for each control configuration, (b) the explicit characterization of the constrained stability region associated with each configuration, (c) the design of a fault-detection filter, and (d) the design of a switching law that orchestrates the re-configuration of control

system in a way that guarantees closed-loop stability in the event of faults in the active control configuration.

To present our results in a compact form, we write the model of Equation 5.1 in a deviation (from the operating unstable steady-state) variable form, by defining  $x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7]^T$  where  $x_1 = In - In_s$ ,  $x_2 = M_1 - M_{1s}$ ,  $x_3 = Y_1 - Y_{1s}$ ,  $x_4 = Y_2 - Y_{2s}$ ,  $x_5 = T - T_s$ ,  $x_6 = T_{w1} - T_{w1s}$ ,  $x_7 = T_{g1} - T_{g1s}$ , and obtain a continuous-time nonlinear system with the following state-space description:

$$\begin{aligned} \dot{x}(t) &= f_{k(t)}(x(t)) + g_{k(t)}(x(t))u_{k(t)} \\ |u_{k(t)}| &\leq u_k^{max} \\ k(t) &\in K = \{1, 2\} \end{aligned} \quad (5.3)$$

where  $x(t) \in \mathbb{R}^7$  denotes the vector of state variables and  $u_k(t) \in [-u_k^{max}, u_k^{max}] \subset \mathbb{R}$  denotes the constrained manipulated input associated with the  $k$ -th control configuration.  $k(t)$ , which takes values in the finite index set  $K$ , represents a discrete state that indexes the vector fields  $f_k(\cdot)$ ,  $g_k(\cdot)$  as well as the manipulated input  $u_k(\cdot)$ . The explicit form of the vector fields  $f_{k(t)}(x(t))$  and  $g_{k(t)}(x(t))$  can be obtained by comparing Equation 5.1 and Equation 5.3 and is omitted for brevity. For each value that  $k$  assumes in  $K$ , the process is controlled via a different manipulated input which defines a given control configuration. Switching between the two available control configurations is controlled by a higher-level supervisor that monitors the process and orchestrates, accordingly, the transition between the different control configurations in the event of control system fault. This in turn determines the temporal evolution of the discrete state,  $k(t)$ . The supervisor ensures that only one control configuration is active at any given time, and allows only a finite number of switches over any finite interval of time. The control objective is to stabilize the process of Equation 5.3 in the presence of actuator constraints and faults in the control system. The basic problem is how to detect the occurrence of a fault and coordinate switching

between the different control configurations (or manipulated inputs) in a way that respects actuator constraints and guarantees closed-loop stability in the event of faults. To simplify the presentation of our results, we will focus only on the state feedback problem where measurements of all process states are available for all times.

### 5.3.1 Constrained Feedback Controller Synthesis

In this step, we synthesize, for each control configuration, a feedback controller that enforces asymptotic closed-loop stability in the presence of actuator constraints. This task is carried out on the basis of the process input/output dynamics. While our control objective is to achieve full state stabilization, process outputs are introduced only to facilitate transforming the system of Equation 5.1 into a form more suitable for explicit controller synthesis.

1. For the primary control configuration with  $u_1 = \frac{F_{M_1}C_{pm1}+F_{In}C_{pIn}}{M_rC_{pr}+B_wC_{ppol}}(T_{feed} - T_{feed}^s)$ , we consider the output  $y_1 = T - T_s$ . This choice yields a relative degree of  $r_1 = 1$  with respect to  $u_1$ . The input/output dynamics can be then expressed in terms of the time-derivative of the variable:  $e_1 = T - T_s$ .

2. For the fall-back control configuration with  $u_2 = (F_c - F_c^s)a_c$ , we choose the output  $y_2 = T - T_s$  which yields a relative degree of  $r_2 = 2$  and the corresponding variables for describing the input/output dynamics take the form:  $e_2^1 = T - T_s$ ,  $e_2^2 = \frac{H_f+H_{g1}-H_{g0}-H_r-H_{pol}}{M_rC_{pr}+B_wC_{ppol}}$ . In particular, for the fall-back control configuration, the system describing the input/output dynamics has the following form:

$$\begin{aligned} \dot{e}_2 &= A_2 e_2 + l_2(e_2) + b_2 \alpha_2 u_2 \\ &:= \bar{f}_2(e_2) + \bar{g}_2(e_2) u_2 \end{aligned} \quad (5.4)$$

where  $A_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ ,  $b_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ,  $e_2 = \begin{bmatrix} e_2^1 \\ e_2^2 \end{bmatrix}$ ,  $l_2(\cdot) = L_{f_2}^2 h_2(x)$ ,  $\alpha_2(\cdot) = L_{g_2} L_{f_2} h_2(x)$ ,  $h_2(x) = y_2$  is the output associated with the fall-back control configuration (the explicit form of the functions  $f_2(\cdot)$  and  $g_2(\cdot)$  is omitted for brevity).

The inverse dynamics, for both the first and second control configurations, have the following form:

$$\begin{aligned}\dot{\eta}_1 &= \Psi_{1,k}(e, \eta) \\ &\vdots \\ \dot{\eta}_{7-r_k} &= \Psi_{7-r_k,k}(e, \eta)\end{aligned}\tag{5.5}$$

where  $k = 1, 2$  and  $\Psi_{1,k} \cdots \Psi_{7-r_k,k}$  are nonlinear functions of their arguments describing the evolution of the inverse dynamics of the  $k$ -th mode.

Using a quadratic Lyapunov function of the form  $V_k = e_k^T P_k e_k$ , where  $P_k$  is a positive-definite symmetric matrix that satisfies the Riccati inequality  $A_k^T P_k + P_k A_k - P_k b_k b_k^T P_k < 0$ , we synthesize, for each control-loop, a bounded nonlinear feedback control law (see [103, 46, 48]) of the form:

$$u_k = -r(x, u_k^{max}) L_{\bar{g}_k} V_k\tag{5.6}$$

where  $r(x, u_k^{max}) =$

$$\frac{L_{\bar{f}_k}^* V_k + \sqrt{(L_{\bar{f}_k}^* V_k)^2 + (u_k^{max} |L_{\bar{g}_k} V_k|)^4}}{(|L_{\bar{g}_k} V_k|)^2 \left[ 1 + \sqrt{1 + (u_k^{max} |L_{\bar{g}_k} V_k|)^2} \right]}\tag{5.7}$$

and  $L_{\bar{f}_k}^* V_k = L_{\bar{f}_k} V_k + \rho |e_k|^2$ ,  $\rho > 0$ . The scalar function  $r(\cdot)$  in Equations 5.6-5.7 can be considered as a nonlinear controller gain. It can be shown that each control configuration asymptotically stabilizes the  $e$  states in each mode. This result, together with the property of the  $\eta$  states to be input-to-state stable, can be used to show, via a small gain argument, asymptotic stability for each control configuration (verified through simulation and analysis of the system of Equation 5.7 with  $e_k = 0$  for both  $k = 1$  and  $k = 2$ ). This controller gain, which depends on both the size of actuator constraints,  $u_k^{max}$ , and the particular configuration used is shaped in a way that guarantees constraint satisfaction and asymptotic closed-loop stability within a well-characterized region in the state-space. The characterization of this region is discussed in the next step.

### 5.3.2 Characterization of Constrained Stability Regions

Given that actuator constraints place fundamental limitations on the initial conditions that can be stabilized, it is important for the control system designer to explicitly characterize these limitations by identifying, for each control configuration, the set of admissible initial conditions starting from where the constrained closed-loop system is asymptotically stable. As discussed in step (d) below, this characterization is necessary for the design of an appropriate switching policy that ensures the fault-tolerance of the control system. The control law designed in step (a) provides such a characterization. Specifically, using a Lyapunov argument, one can show that the set

$$\Theta(u_k^{max}) = \{x \in \mathbb{R}^7 : L_{\bar{f}_k}^* V_k \leq u_k^{max} |L_{\bar{g}_k} V_k|\} \quad (5.8)$$

describes a region in the state space where the control action satisfies the constraints and the time-derivative of the corresponding Lyapunov function is negative-definite along the trajectories of the closed-loop system. Note that the size of this set depends, as expected, on the magnitude of the constraints. In particular, the set becomes smaller as the constraints become tighter (smaller  $u_k^{max}$ ). For a given control configuration, one can use the above inequality to estimate the stability region associated with this configuration. This can be done by constructing the largest invariant subset of  $\Theta$ , which we denote by  $\Omega(u_k^{max})$ . Confining the initial conditions within the set  $\Omega(u_k^{max})$  ensures that the closed-loop trajectory stays within the region defined by  $\Theta(u_k^{max})$ , and thereby  $V_k$  continues to decay monotonically, for all times that the  $k$ -th control configuration is active (see [46] for further discussion on this issue).

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  $V_{c_k} = V_k + V_{\eta_k}$ , where  $V_{\eta_k} = \eta^T P_{\eta_k} \eta$  and  $P_{\eta_k}$  is a positive definite matrix and choosing a level set of  $V_{c_k}$ ,  $\Omega_{c_k}$ , for which  $\dot{V}_{c_k} < 0$  for all  $x$  in  $\Omega_{c_k}$ .



**Remark 5.1** Note that the composite Lyapunov functions,  $V_{c_k}$ , used in implementing the switching rules, are in general different from the Lyapunov functions  $V_k$  used in designing the controllers. Owing to the ISS property of the  $\eta_k$ -subsystem of each mode, only a Lyapunov function for the  $e_k$  subsystem, namely  $V_k$ , is needed and used to design a controller that stabilizes the full  $e_k - \eta_k$  interconnection for each mode. However, when implementing the switching rules (constructing the  $\Omega_{c_k}$ ), we need to track the evolution of  $x$  (and hence the evolution of both  $e_k$  and  $\eta_k$ ). Therefore, the Lyapunov functions used in verifying the switching conditions at any given time,  $V_{c_k}$ , are based on  $x$ . From the asymptotic stability of each mode, the existence of these Lyapunov functions is guaranteed by converse Lyapunov theorems. Note also that the above controller design is only one example of a controller design that allows for an explicit characterization of the stability region and is used for the purpose of illustration. Other controller designs such as the hybrid predictive controller [55, 114, 54, 116] that enable implementation of predictive controllers with a well characterized stability region can also be used to achieve fault-tolerant control within the proposed framework.

**Remark 5.2** Note that in practical implementation when the state trajectory gets close to the desired equilibrium point, the first  $(|L_{\bar{g}_k} V_k|)^2$  term in the denominator of the control law of Equation 5.7 could cause chattering in the control action. To alleviate this chattering, a small positive number  $\nu_k$  may be added to the first  $(|L_{\bar{g}_k} V_k|)^2$  term in the denominator. The addition of  $\nu_k$  allows for achieving practical stability, with decrease in magnitude of  $\nu_k$  (while keeping it large enough to avoid chattering) resulting in the state trajectory going further closer to the desired equilibrium point.

### 5.3.3 Fault-Detection Filter Design

The next step in implementing fault-tolerant control is that of designing appropriate fault-detection filters that can detect the occurrence of a fault in the control actuator by observing the behavior of the closed-loop process. To this end, we design for a

given control configuration, a fault detection filter of the form:

$$\begin{aligned} \dot{w}(t) &= f_k(w(t)) + g_k(w(t))u_k(w) \\ |u_k| &\leq u_k^{max} \\ r_k(t) &= \|x(t) - w(t)\| \end{aligned} \tag{5.9}$$

where  $x(t) \in \mathbb{R}^7$  denotes the vector of state variables and  $u_k(t) \in [-u_k^{max}, u_k^{max}] \subset \mathbb{R}$  denotes the constrained manipulated input associated with the  $k$ -th control configuration,  $w(t) \in \mathbb{R}^7$  is the vector of filter states and  $r_k(t) \in \mathbb{R}$  is the residual that detects the occurrence of a fault. The filter states are initialized at the same value as the process states ( $w(0) = x(0)$ ) and essentially predict the evolution of the process in the absence of actuator faults. The residuals captures the difference between the predicted evolution of the states in the absence of faults and that of the process state, thereby detecting faults in the control actuators. Specifically, the value of  $r_k(t)$  becomes non-zero at the earliest time that a fault occurs (for a detailed analysis of the detection properties of the filter, see Chapter 4).

**Remark 5.3** Note that in the presence of measurement noise, the value of  $r(t)$  will be nonzero even in the absence of faults. To handle this problem, the filter should declare a fault only if the value of  $r(t)$  increases beyond some threshold,  $\delta$ , where  $\delta$  accounts for the deviation of the plant measurements from the nominal measurements in the absence of faults (see the simulation section for a demonstration). Note also, that plant model mismatch or unknown disturbances can also cause the value of  $r(t)$  to 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 redesigning the filter as well as the controllers for the individual control configuration. Specifically, as in the case of measurement noise, the filter should declare a fault only if the value of  $r(t)$  increases beyond some threshold,  $\delta$ , where  $\delta$  accounts for the deviation of the plant dynamics from the nominal dynamics in the absence of faults. The controllers for the individual control configurations need to be redesigned to mitigate the effect of disturbances on the process, in a way that allows the characterization

of the robust stability regions. The robust stability region can subsequently be used in deciding which backup control configuration should be implemented in the closed-loop. With regard to the fault-detection filter, the detection threshold provides a suitable handle that can be used to achieve early fault detection. In the presence of noise, however, having a small fault-detection threshold can lead to the triggering of false alarms (as demonstrated in the simulation example) and should be picked to achieve the desired tradeoff between avoiding false alarms and detecting faults.

### 5.3.4 Fault-Tolerant Switching Logic

Having designed the feedback control laws, characterized the stability region associated with each control configuration, and designed the fault-detection filter, the fourth step is to derive the switching policy that the supervisor needs to employ to activate/deactivate the appropriate control configurations in the event of faults. The key idea here is that, because of the limitations imposed by constraints on the stability region of each configuration, the supervisor can only activate the control configuration for which the closed-loop state is within the stability region at the time of control system fault. Without loss of generality, let the initial actuator configuration be  $k(0) = 1$ ,  $T_{fault}$  be the time when this configuration fails and let  $T_{detect}$  be the earliest time at which the value of  $r_1(t) > \delta_{r_1} > 0$  (where  $\delta_{r_1}$  is the detection threshold chosen based on the acceptable level of deviation of the actual closed-loop performance from the desired one), then the switching rule given by

$$k(t \geq T_{detect}) = 2 \text{ if } x(T_{detect}) \in \Omega_{c_2}(u_2^{max}) \quad (5.10)$$

guarantees asymptotic closed-loop stability. The implementation of the above switching law requires monitoring the closed-loop state trajectory with respect to the stability regions associated with the various actuator configurations. This idea of tying the switching logic to the stability regions was first proposed in [49] for the control

of switched nonlinear systems.

## 5.4 Simulation Results

Several simulation runs were carried out to evaluate the effectiveness of the proposed fault-detection and fault-tolerant control strategy. Figure 5.2 shows the evolution of the open-loop state profiles. Under the operating conditions listed in Table 5.2, the open-loop system behaves in an oscillatory fashion (i.e., the system possesses an open-loop unstable steady-state surrounded by a stable limit cycle).

First, process operation under primary control configuration was considered (i.e., the feed temperature,  $T_{feed}$ , was the manipulated input) and a bounded nonlinear controller was designed using the formula of Equations 5.6-5.7. Specifically, a quadratic function of the form  $V_1 = \frac{1}{2}(T - T_s)^2$  and  $\rho_1 = 0.01$  were used to design the controller and a composite Lyapunov function of the form  $V_{c_1} = 5 \times 10^{-3}(In - In_s)^4 + 5 \times 10^{-4}(M_1 - M_{1s})^2 + 5 \times 10^{-11}(Y_1 - Y_{1s})^2 + 5 \times 10^{-11}(Y_2 - Y_{2s})^2 + 5 \times 10^{-4}(T - T_s)^2 + 5 \times 10^{-2}(T_{w_1} - T_{w_{1s}})^2 + 5 \times 10^{-11}(T_{g_1} - T_{g_{1s}})^2$  was used to estimate the stability region of the primary control configuration yielding a  $c_1^{max} = 62$ .

The first  $(|L_{\bar{g}_k} V_k|)^2$  term in the denominator of the control law of Equation 5.7 was replaced by  $(|L_{\bar{g}_k} V_k|)^2 + \nu_k$  (as discussed in Remark 5.2), with  $\nu_1 = 1$  and  $\nu_2 = 5 \times 10^{-9}$ , to alleviate chattering of the control action close to the desired equilibrium point under configurations 1 and 2, respectively. Figure 5.3 shows the evolution of the closed-loop state profiles and Figure 5.4 shows the evolution of the manipulated inputs starting from the initial condition  $In(0) = 450 \frac{mol}{m^3}$ ,  $M_1(0) = 340 \frac{mol}{m^3}$ ,  $Y_1(0) = 4.6 mol$ ,  $Y_2(0) = 4.6 mol$ ,  $T(0) = 360 K$ ,  $T_{w_1}(0) = 300 K$ , and  $T_{g_1}(0) = 300 K$  for which  $V_{c_1} = 61.4$ . Since this initial state is within the stability region of the primary control configuration (i.e.,  $V_{c_1}(x(0)) \leq c_1^{max}$ ), the primary control configuration is able to

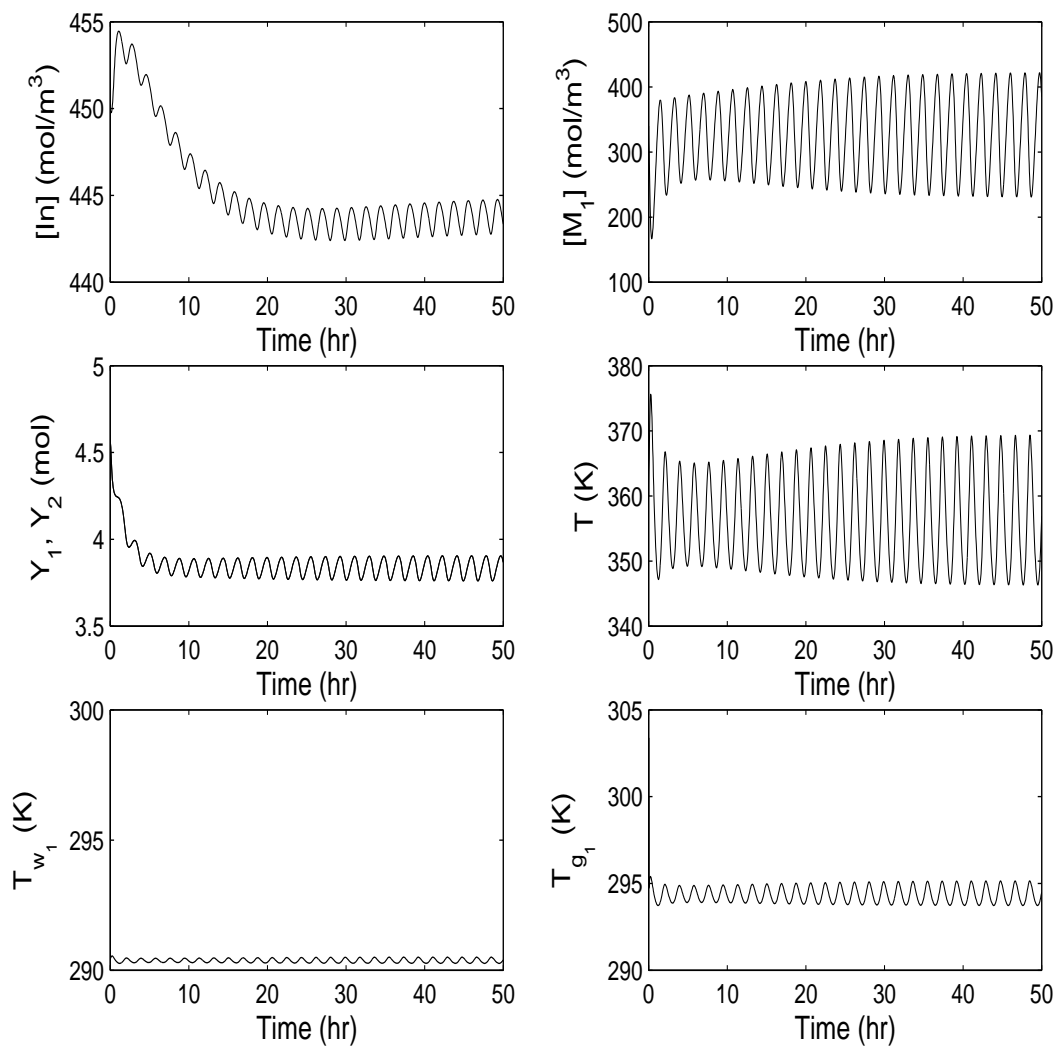


Figure 5.2: Evolution of the open-loop process states.

stabilize the system at the steady-state of interest.

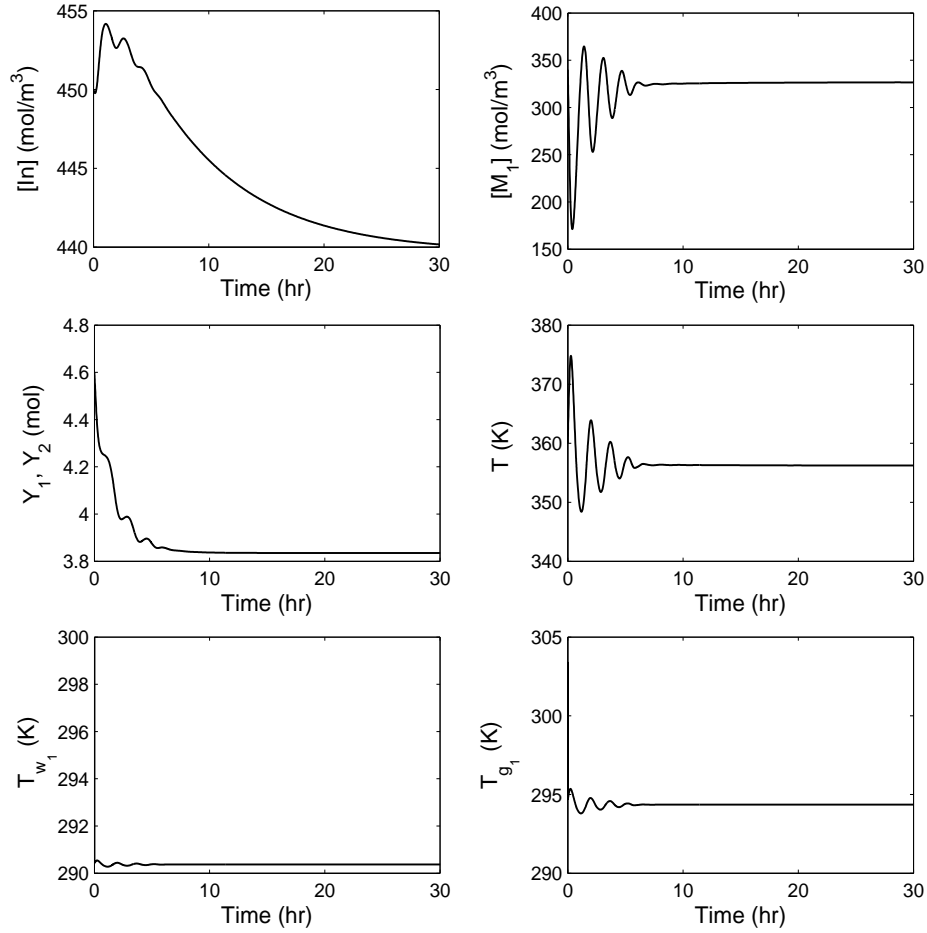


Figure 5.3: Closed-loop state profiles under the primary control configuration.

Next, we considered the case of having a fault in the primary control configuration. In this case, the supervisor had available a fall-back control configuration with the catalyst flowrate,  $F_c$ , as the manipulated input. A quadratic Lyapunov function of the form  $V_2 = e_2^T P_2 e_2$  and  $\rho_2 = 0.01$  was used to design the controller that used the fall-back control configuration and a composite Lyapunov function of the form  $V_{c_2} = 5 \times 10^{-3}(In - In_s)^4 + 5 \times 10^{-4}(M_1 - M_{1s})^2 + 5 \times 10^{-11}(Y_1 - Y_{1s})^2 + 5 \times$

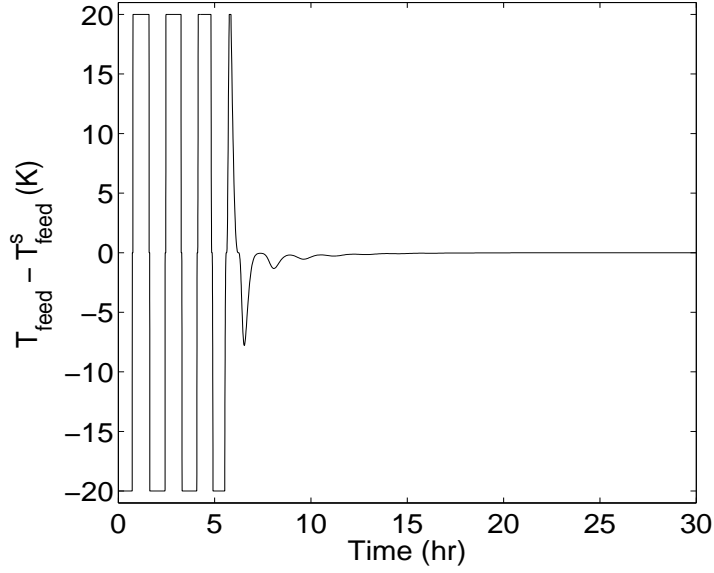


Figure 5.4: Manipulated input profile under primary control configuration.

$10^{-11}(Y_2 - Y_{2s})^2 + 5 \times 10^{-4}(T - T_s)^2 + 5 \times 10^{-11}(T_{w_1} - T_{w_{1s}})^2 + 5 \times 10^{-11}(T_{g_1} - T_{g_{1s}})^2$  was used to estimate the stability region of the fall-back control configuration yielding a  $c_2^{max} = 56.8$ .

To demonstrate that control loop reconfiguration results in fault-tolerant reactor control in the presence of input constraints, we carried out the following simulations: We first initialized the reactor at  $In(0) = 450 \frac{mol}{m^3}$ ,  $M_1(0) = 340 \frac{mol}{m^3}$ ,  $Y_1(0) = 4.6 mol$ ,  $Y_2(0) = 4.6 mol$ ,  $T(0) = 360 K$ ,  $T_{w_1}(0) = 300 K$ , and  $T_{g_1}(0) = 300 K$  resulting in  $V_{c_1} = 61.4$  which implied that this initial state was within the stability region of the primary control configuration. Consider now, a fault in the primary control configuration at time  $T_{fault} = 5 hrs 34 mins$  (see dashed lines in Figures 5.5-5.6). In the case of no switching to fall-back control configuration or no backup control configurations available, closed-loop stability is not achieved and the process behaves in an oscillatory fashion (solid line in Figure 5.5).

However, applying our fault-detection and fault-tolerant control strategy, the su-

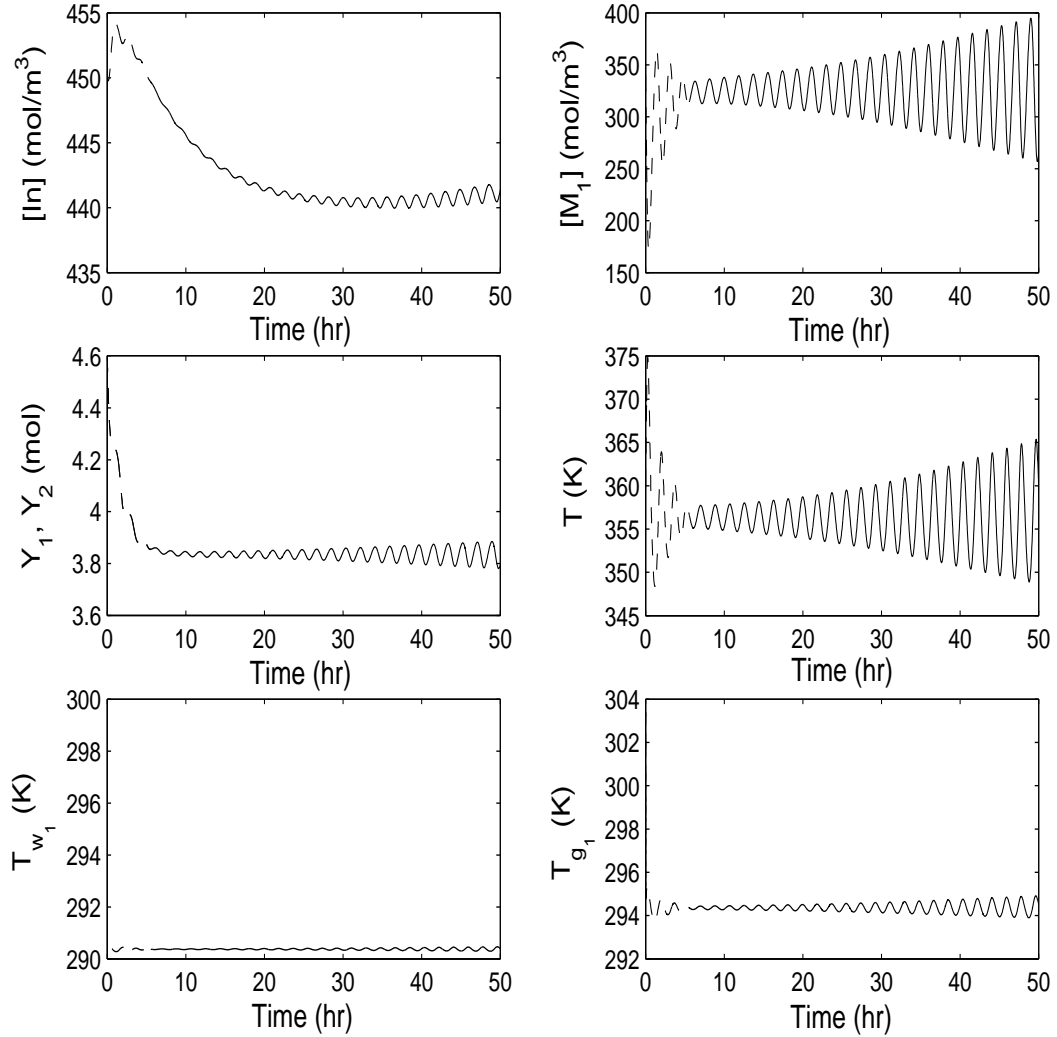


Figure 5.5: Evolution of the closed-loop state profiles under primary control configuration (dashed lines) and no fall-back control configuration available to switch to (or fall-back control configuration is not activated) resulting in open-loop oscillatory behavior (solid lines) after primary control configuration fails at  $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ .



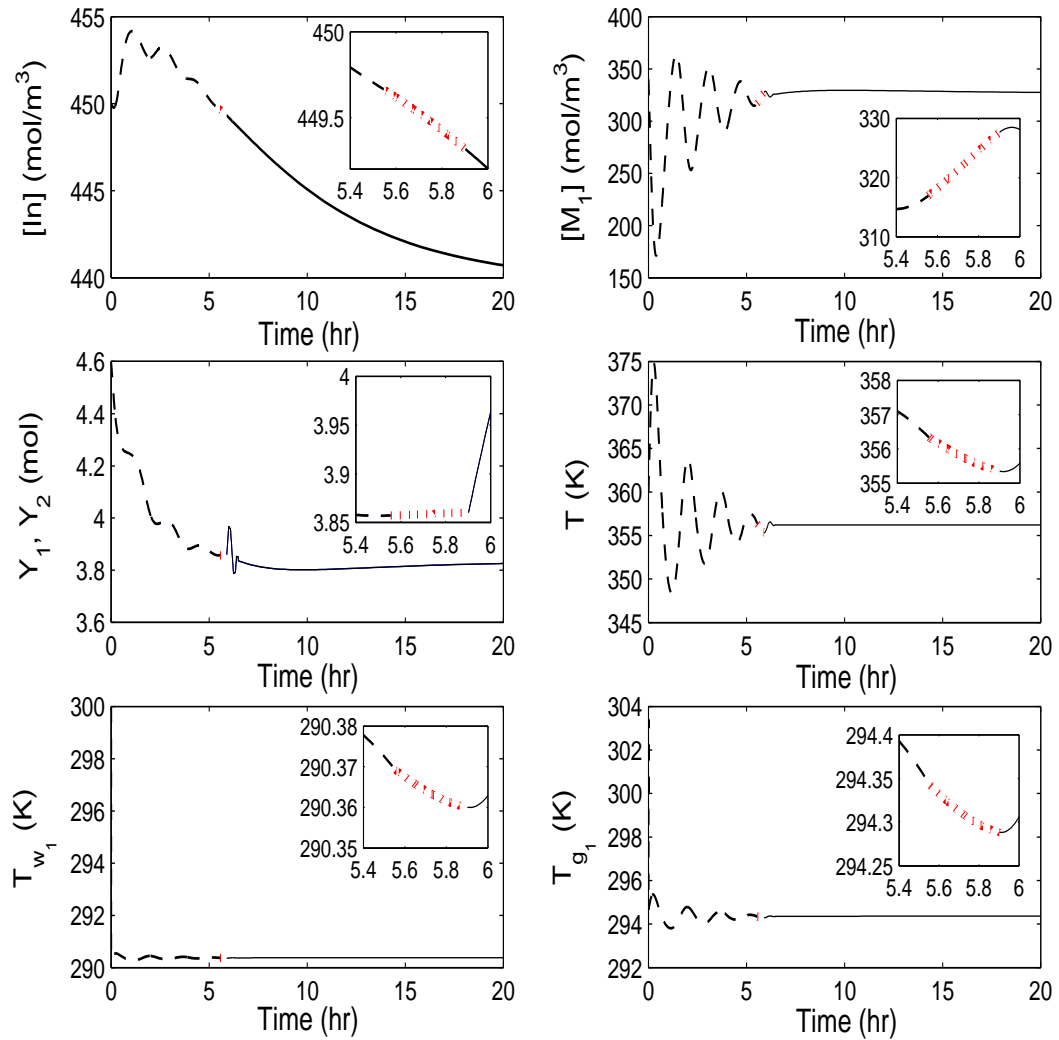


Figure 5.6: Evolution of the closed-loop state profiles under primary control configuration (dashed lines) which fails at  $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ . At this point, the process starts operating open-loop (dotted lines). At  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$ , the detection filter verifies that there is a fault on the primary control configuration and the control system switches to the fall-back control configuration (solid lines).

supervisor kept track of the residual value  $r_1$  (see dashed line in Figure 5.7) and observed the residual value  $r_1$  becoming non-zero at  $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ . At  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$ , the residual value  $r_1$  reached the detection threshold ( $\delta_{r_1} = 0.5$ ) and a fault on primary control configuration was declared. The supervisor, then, checked if switching to fall-back control configuration would preserve stability. This was done by evaluating the value of the composite Lyapunov function of the fall-back control configuration at  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$  where the states were of the following values:  $In(T_{detect}) = 449.7 \frac{\text{mol}}{\text{m}^3}$ ,  $M_1(T_{detect}) = 316.9 \frac{\text{mol}}{\text{m}^3}$ ,  $Y_1(T_{detect}) = 3.86 \text{ mol}$ ,  $Y_2(T_{detect}) = 3.86 \text{ mol}$ ,  $T(T_{detect}) = 356.3 \text{ K}$ ,  $T_{w_1}(T_{detect}) = 290.4 \text{ K}$ , and  $T_{g_1}(T_{detect}) = 294.3 \text{ K}$ . Since  $V_{c_2}(x(T_{detect})) = 49.6 \leq c_2^{max}$ , the state, at the time the filter detected the fault in the primary control configuration, was within the stability region of the fall-back control configuration. Therefore, switching to fall-back control configuration would preserve closed-loop stability (see solid lines in Figure 5.6).

Next, we also investigated the implementation of the fault-detection and fault-tolerant control strategy in the presence of measurement noise. Specifically, we considered Gaussian measurement noise of the following magnitude:  $In = 0.5 \frac{\text{mol}}{\text{m}^3}$ ,  $M_1 = 0.3 \frac{\text{mol}}{\text{m}^3}$ ,  $Y_1 = 0.04 \text{ mol}$ ,  $Y_2 = 0.04 \text{ mol}$ ,  $T = 0.8 \text{ K}$ ,  $T_{w_1} = 0.3 \text{ K}$ , and  $T_{g_1} = 0.3 \text{ K}$ . Note that in the presence of measurement noise, the value of the residual stayed non-zero even in the absence of actuator faults (for the measurement noise considered in this study, a detection threshold of  $\delta_{r_1} = 0.5$  was no longer appropriate and triggered false alarms). A new threshold that captured the effect of the measurement noise on the value of the residual was needed. A detection threshold of ( $\delta_{r_1} = 0.7$ ) was then picked. Consider once again a fault in the primary control configuration at time  $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ . This fault was detected by the fault-detection filter via the residual reaching the threshold at  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$  (see

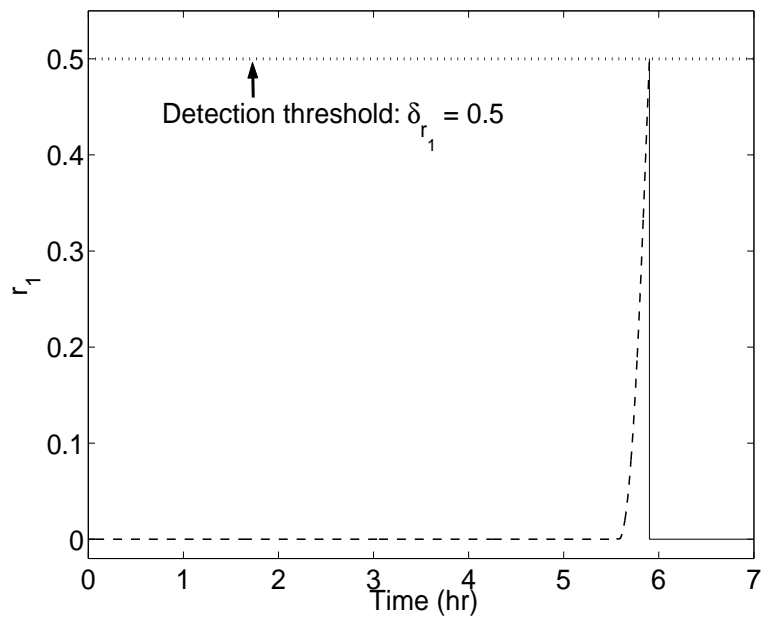


Figure 5.7: Evolution of the detection filter residual value under primary control configuration (dashed line). At  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$ , the detection filter residual value reaches the detection threshold of 0.5 which verifies that a fault on the primary control configuration occurs. A switch to the fall-back control configuration (solid line) resets the detection filter residual back to zero.

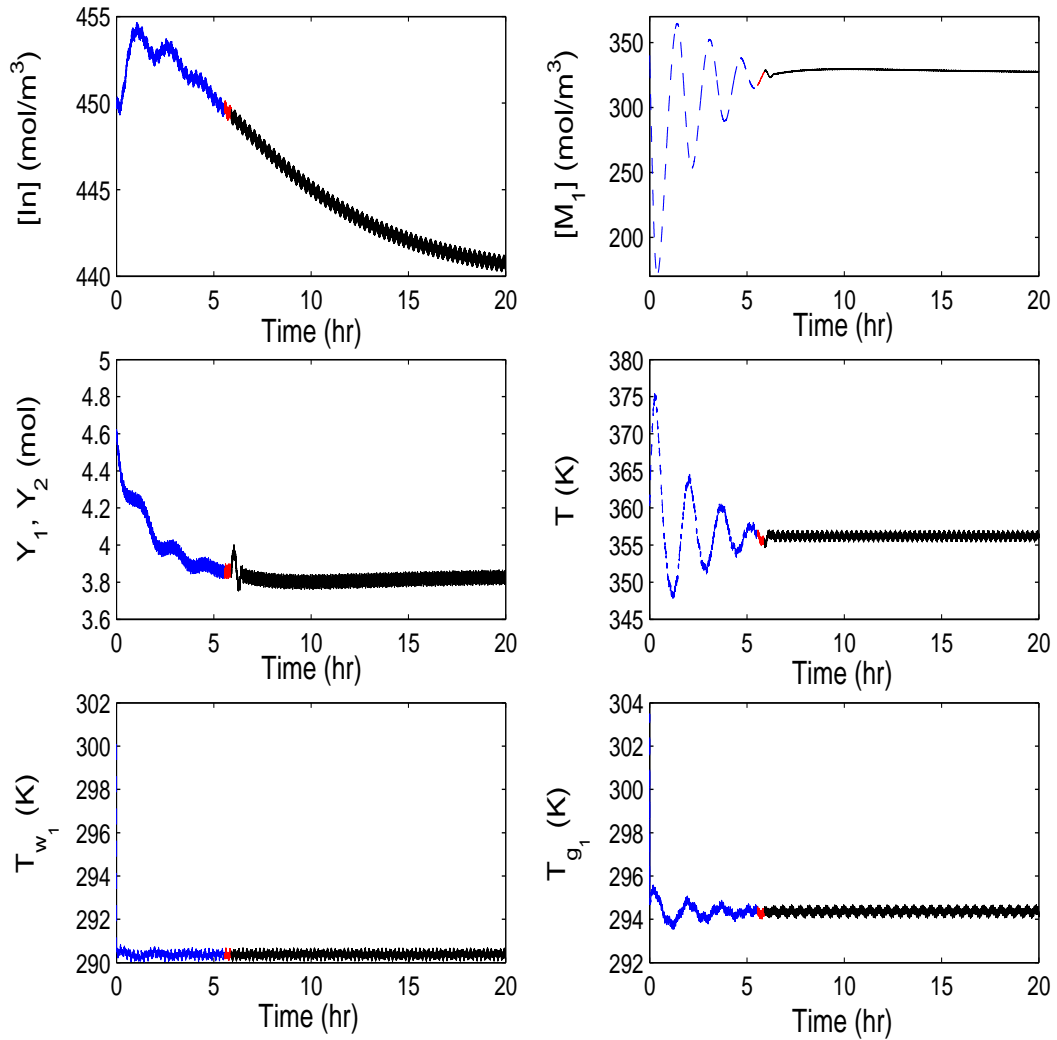


Figure 5.8: Evolution of the closed-loop state profiles in the case of measurement noise under primary control configuration (dashed lines) which fails at  $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ . At this point, the process starts operating open-loop (dotted lines). At  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$ , the detection filter verifies that there is a fault on the primary control configuration and the control system switches to the fall-back control configuration (solid lines).

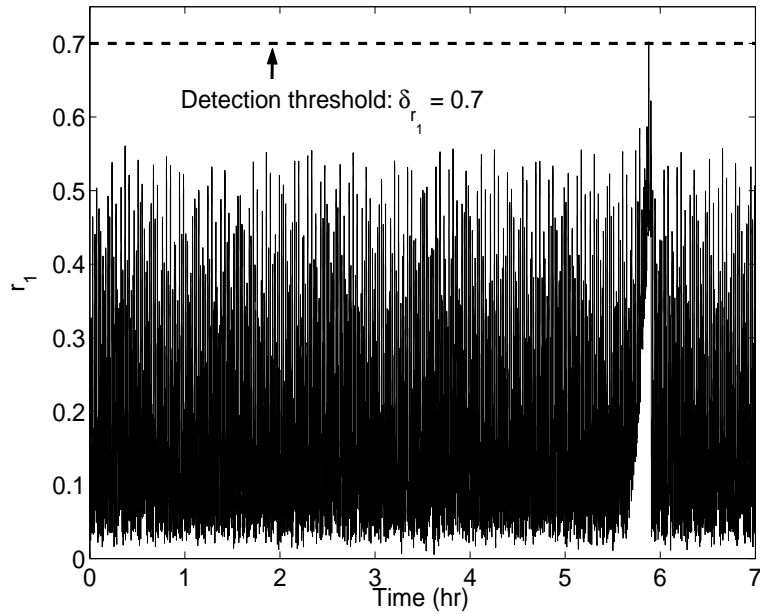


Figure 5.9: Evolution of the detection filter residual value in the case of measurement noise. A detection threshold of 0.5 triggers false alarm even before real fault on primary control configuration at  $T_{fault} = 5 \text{ hrs } 34 \text{ mins}$ . A new detection threshold of 0.7 is picked and implemented. At  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$ , the detection filter residual value reaches the detection threshold of 0.7 which verifies that a fault on the primary control configuration occurs. A switch to the fall-back control configuration resets the detection filter residual back to normal.

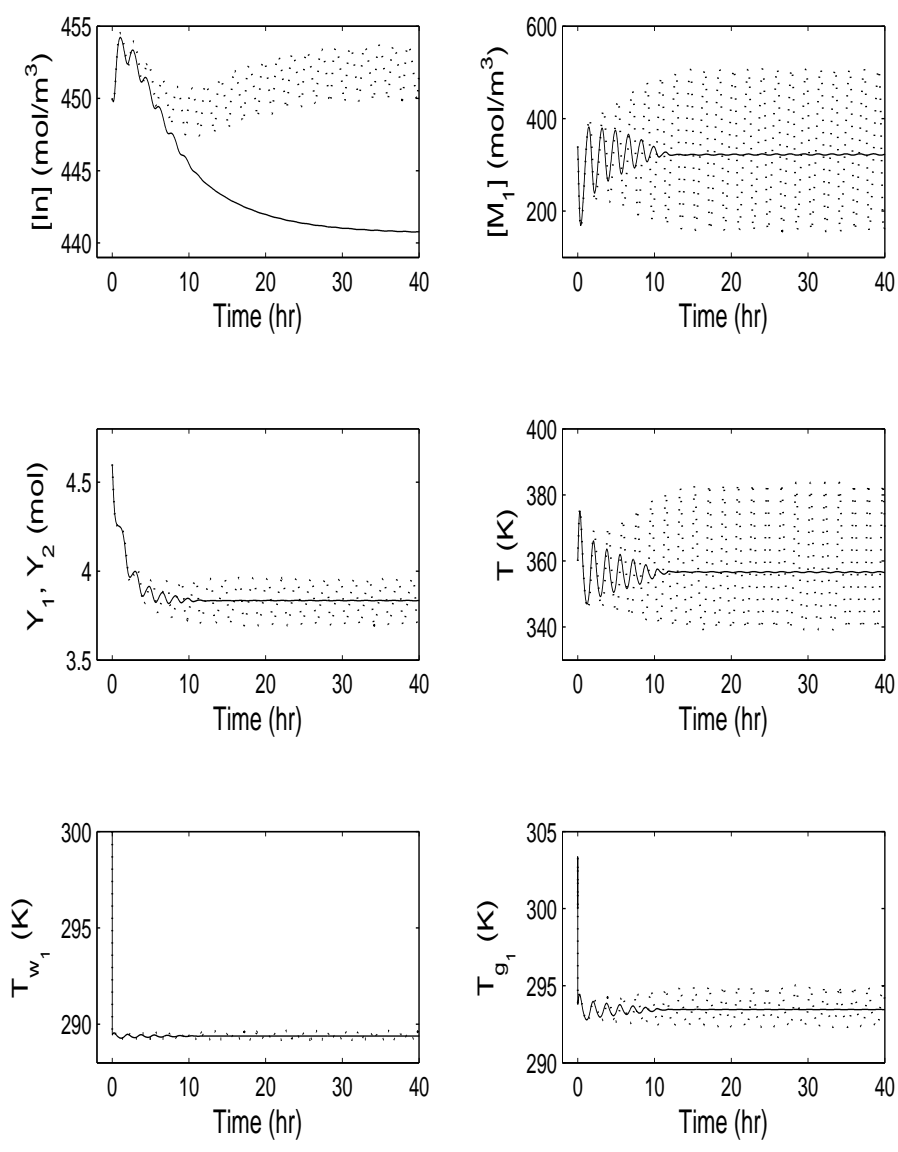


Figure 5.10: Evolution of the open-loop (dotted lines) and closed-loop (solid lines) state profiles under the primary control configuration in the presence of parametric model uncertainty and disturbances.

dashed line in Figure 5.9). The supervisor, then, checked if switching to fall-back control configuration would preserve stability. As before, this was done by evaluating the value of the composite Lyapunov function of the fall-back control configuration at  $T_{detect} = 5 \text{ hrs } 54 \text{ mins}$  where the states were of the following values:  $In(T_{detect}) = 449.3 \frac{\text{mol}}{\text{m}^3}$ ,  $M_1(T_{detect}) = 327.5 \frac{\text{mol}}{\text{m}^3}$ ,  $Y_1(T_{detect}) = 3.83 \text{ mol}$ ,  $Y_2(T_{detect}) = 3.83 \text{ mol}$ ,  $T(T_{detect}) = 355.6 \text{ K}$ ,  $T_{w_1}(T_{detect}) = 290.4 \text{ K}$ , and  $T_{g_1}(T_{detect}) = 294.4 \text{ K}$ . Since  $V_{c_2}(x(T_{detect})) = 42.7 \leq c_2^{max}$ , the state, at the time the filter detected the fault in the primary control configuration, was within the stability region of the fall-back control configuration. Subsequent switching to the fall-back control configuration once again resulted in closed-loop stability (see solid lines in Figure 5.8).

Finally, we also evaluated the robustness of the controller that is a vital component of the fault-tolerant control structure. We considered values of some of the process parameters being different from the ones used in the controller design, specifically,  $E_a = 38.058 \text{ kJ/mol}$  and  $H_{reac} = 3780.429 \text{ kJ/kg}$  and also in the presence of disturbance in the inlet coolant temperature, with  $T_{w_i} = 288.56 \text{ K}$ . The dotted lines in Figure 5.10 show the open-loop profiles illustrating the effect of the presence of disturbances and uncertainty in the parameters on the process states. In contrast, when the primary control configuration is implemented, the controller is able to reject the disturbances and stabilize the process at the desired equilibrium point (see solid lines in Figure 5.10).

## 5.5 Conclusions

In this chapter, we focused on fault-tolerant control of an industrial gas phase polyethylene reactor. Initially, a family of candidate control configurations, characterized by different manipulated inputs, were identified. For each control configuration, a

bounded nonlinear feedback controller, that enforced asymptotic closed-loop stability in the presence of constraints, was designed, and the constrained stability region associated with it was explicitly characterized using Lyapunov-based tools. A fault-detection filter was designed to detect the occurrence of a fault in the control actuator by observing the deviation of the process states from the expected closed-loop behavior. A switching policy was then derived, on the basis of the stability regions, to orchestrate the activation/deactivation of the constituent control configurations in a way that guaranteed closed-loop stability in the event of control system faults. Closed-loop simulations were carried out to implement the fault-tolerant control strategy on the gas phase polyethylene reactor and to demonstrate the implementation of the fault-tolerant control method in the presence of measurement noise.



## Chapter 6

# Fault-Tolerant Control of Nonlinear Process Systems Subject to Sensor Faults

### 6.1 Introduction

The ability to implement fault-tolerant control relies on some degree of redundancy in the control configurations (availability of sets of sensor/actuator combinations that can be used to implement controllers), that can either be used all at one time (the reliable control approach, for example, [178]), or activated when the need arises (the reconfiguration approach). The use of only as many control loops as required at a time is motivated by economic considerations (to save on unnecessary control action), and has been employed in the context of chemical processes; however, the available results are mostly based on the assumption of a linear system description (for example, [10, 173]), and do not account for complexities such as control constraints.

In implementing fault-tolerant control (as well as feedback control), the importance of sensors is well-recognized and several researchers have focused on the prob-

lem of efficient sensing and measurement for well-functioning sensors and networks of sensors [23, 7, 125]. In [15, 157, 68, 150] the problem of measurements arriving at different known rates and its implication on simulation and control (multi-rate control) is addressed. In chemical processes, sensor data losses arising due to sampling, measurement or communication irregularities are more likely to be manifested as intermittent availability of measurements (asynchronous measurements), where only an average rate of availability of measurements is known, but not the exact times when the measurements will be available.

When explicitly considered, irregular measurements can be analyzed as a robustness problem. Specifically, for a given stabilizing control law, a bound on the sensor data loss rate (defined as the ratio of the time during which measurements are available over the total time) can be computed such that if the sensor data loss rate is within this bound, closed-loop stability is preserved. The difference in the nature of sensor irregularities (measurements arriving at different known rates as opposed to asynchronously) has important implications in the robustness of a given system to sensor data losses. Furthermore, for unconstrained systems, such a bound for the data loss rate (defined over an infinite time interval) can be computed (for example, see [75, 186] and the references therein). For constrained systems, however, for such a bound on the data loss rate to exist, it has to be defined over a finite time interval where the derived bound accounts for the limitations imposed by the presence of constraints.

The extensive work in the area of nonlinear process control can be utilized toward computing such a bound, and in choosing the appropriate feedback laws (for excellent reviews of results in the area of nonlinear process control see [95, 14, 180, 109]; for a more recent review see [29]). These approaches have recently been utilized to

address the problem of fault-tolerant control of nonlinear processes subject to constraints and faults in the control actuators. In Chapter 2, sensor faults arising due to communication losses were modeled as delays in implementing the control action and a reconfiguration strategy was devised to achieve fault-tolerance subject to faults in the control actuators. In Chapter 4, a reconfiguration based approach was utilized for the purpose of achieving tolerance to actuator faults under the assumptions that the measurements were continuously available. The results of Chapter 2 and 4 however, do not take the presence of intermittent sensor data losses into account either in the implementation of individual control configurations, or in the reconfiguration strategies. The fault-tolerant (or even stabilizability in the absence of faults) capabilities of the results of Chapter 2 and 4 therefore do not hold in the presence of sensor data losses. Furthermore, outside of these recent results as well the problem of fault-tolerant control for handling sensor faults for nonlinear systems subject to constraints in the control actuators has received limited attention.

Motivated by the above, in this chapter, we consider the problem of fault-tolerant control of nonlinear process systems subject to input constraints and sensor faults (both complete failures and asynchronous measurements) [120]. We employ a reconfiguration approach, wherein, for a given process, a set of candidate control configurations are first identified, and in the event of a fault an appropriate backup configuration is activated to maintain stability. To illustrate the importance of accounting for the presence of constraints, we first consider sensor faults manifested as complete loss of measurements (faults that necessitate taking corrective action to repair the sensors). We address the problem of determining which candidate control configuration should be implemented in the closed-loop system to achieve stability after the sensor is recovered (this analysis is carried out under the assumption of continuous

availability of measurements when the sensor is functioning). We then consider the problem in the presence of intermittent sensor data losses. We define the sensor data loss rate to account for the presence of constraints (specifically, we define the data loss rate over a finite time interval) and analyze the stability properties in the presence of input constraints and sensor data losses. We characterize the stability region (that is, the set of initial conditions starting from where closed-loop stabilization under continuous availability of measurements is guaranteed) and the maximum allowable data loss rate that a given control configuration can tolerate. If the data loss rate goes above the allowable data loss rate, reconfiguration is triggered and a candidate backup configuration is activated for which the state of the closed-loop system resides in the stability region of the candidate configuration and the data loss rate is less than the allowable data loss rate for the candidate control configuration. We use a chemical reactor to illustrate our method and then demonstrate an application to a polyethylene reactor.

## 6.2 Preliminaries

We consider nonlinear processes with input constraints, described by:

$$\begin{aligned} \dot{x} &= f(x) + G_{k(t)}(x)u_{k(t)}(y(t)) \\ y(t) &= \begin{cases} x(t) & t \in [t_{2i}, t_{2i+1}) \\ x(t_{2i+1}) & t \in [t_{2i+1}, t_{2i+2}) \end{cases} \\ u_k &\in \mathbf{U}_k, k(t) \in \mathcal{K} = \{1, \dots, N\}, \quad N < \infty \end{aligned} \tag{6.1}$$

where  $x \in \mathbb{R}^n$  denotes the vector of state variables,  $y \in \mathbb{R}^m$  denotes the vector of measured variables,  $[t_{2i}, t_{2i+1})$  and  $[t_{2i+1}, t_{2i+2})$  denote the time intervals during which measurements of the state variables are available, and are lost, respectively, with  $t_0 = 0$  (that is, measurement being initially available),  $u_{k(t)}(x) \in \mathbb{R}^m$  denotes the manipulated inputs under the  $k$ th configuration taking values in a nonempty

convex subset  $\mathbf{U}_k$  of  $\mathbb{R}^m$ , where  $\mathbf{U}_k = \{u \in \mathbb{R}^m : \|u\| \leq u_k^{max}\}$ ,  $\|\cdot\|$  is the Euclidean norm of a vector,  $u_k^{max} > 0$  is the magnitude of input constraints and  $f(0) = 0$ . The vector function  $f(x)$  and the matrix  $G_k(x) = [g_{1,k}(x) \cdots g_{m,k}(x)]$  are assumed to be sufficiently smooth on their domains of definition.  $k(t)$ , which takes values in the finite index set  $\mathcal{K}$ , represents a discrete state that indexes the matrix  $G_k(\cdot)$  as well as the manipulated input  $u_k(\cdot)$ . For each value that  $k$  assumes in  $\mathcal{K}$ , the system is controlled via a different set of manipulated inputs which defines a given control configuration. 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 left, that is,  $x(T^-) = \lim_{t \rightarrow T^-} x(t)$ . Throughout the manuscript, we assume that for any  $u_k \in \mathbf{U}_k$  the solution of the system of Equation 6.1 exists and is continuous for all  $t$ .

We next review one example of a state feedback controller [46, 48] (inspired by the results on bounded control in [103]) that, under the assumption of continuous availability of measurements, provides an explicit estimate of the stability region for the closed-loop system subject to constraints (for more details on the controller design, see [46, 48]).

**Theorem 6.1** *Consider the nonlinear system of Equation 6.1 under state feedback (that is,  $x(t)$  is available for all  $t \geq 0$ ) for a configuration  $k$ , for which a Control Lyapunov Function  $V_k$  exists, under the following bounded nonlinear feedback controller:*

$$u_k = -w_k(x, u_k^{max})(L_{G_k} V_k(x))^T \quad (6.2)$$

where  $w_k(x, u_k^{max}) =$

$$\begin{cases} \frac{\alpha_k(x) + \sqrt{\alpha_k^2(x) + (u_k^{max} \|b_k^T(x)\|)^4}}{\|b_k^T(x)\|^2 \left[1 + \sqrt{1 + (u_k^{max} \|b_k^T(x)\|)^2}\right]}, & b_k^T(x) \neq 0 \\ 0, & b_k^T(x) = 0 \end{cases} \quad (6.3)$$

with  $\alpha_k(x) = L_{f_k} V_k(x) + \rho_k V_k(x)$ ,  $\rho_k > 0$  and  $b_k(x) = L_{G_k} V_k(x)$ . Assume that the set  $\Phi_k(u_k^{max})$  of  $x$  satisfying

$$L_{f_k} V_k(x) + \rho_k V_k(x) \leq u_k^{max} \|(L_{G_k} V_k(x))^T\| \quad (6.4)$$

contains the origin and a neighborhood of the origin. Also, let  $\Omega_k(u_k^{max}) := \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\}$  be a level set of  $V_k$ , completely contained in  $\Phi_k$ , for some  $c_k^{max} > 0$ . Then for all  $x(0) \in \Omega_k(u_k^{max})$  the control law of Equations 6.2-6.4 guarantees that the origin of the closed-loop system is asymptotically stable [48].

**Proof of Theorem 6.1** Please refer to [46, 48] for proof of Theorem 6.1.

**Remark 6.1** The problems caused by input constraints have motivated numerous studies on the dynamics and control of systems subject to input constraints. Important contributions include results on optimization-based methods such as model predictive control (for example, [66, 164, 109]) and Lyapunov-based control (for example, [103, 158, 85, 92]). 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 [92] for a survey of results in this area. Recently, we developed a hybrid predictive control structure that employs switching between bounded control and MPC for stabilization of nonlinear systems [54], and nonlinear systems with uncertainty [116], subject to input constraints via using Lyapunov-based controllers [46, 48] as fall-back controllers. More recently Lyapunov-based model predictive controllers were designed that guarantee stabilization from an explicitly characterized set of initial conditions in the presence of input [115] and input and state [117] constraints. The controller of Equation 6.3 is one example of a controller design

that provides an explicit characterization of the stability region in the presence of input constraints, and is only used to illustrate the main ideas behind the proposed approach. The results in this chapter are not limited to this particular controller design, and any other controller design that provides an explicit characterization of the stability region can be used instead (for example, the hybrid predictive controller [54, 116] or the Lyapunov-based predictive controller [115, 117]; for further details and references, see [29]).

### 6.2.1 A Chemical Reactor Example

In this section, we re-visit the chemical reactor in Section 2.4.1 that we will use to illustrate the key features of our proposed method. The mathematical model of the process described in Equation 2.13. The values of the process parameters and the corresponding steady-state values can be found in Section 2.4.1. It was verified that under these conditions, the system of Equation 2.13 has three steady-states (two locally asymptotically stable and one unstable at  $(T_s, C_{As}) = (388 \text{ K}, 3.59 \text{ mol/L})$ ).

The control objective considered here is that of stabilizing the reactor at the (open-loop) unstable steady-state using the measurements of concentration and temperature. The following manipulated input candidates are assumed to be available (see Figure 2.3):

1. Configuration 1: Rate of heat input,  $u_1 = Q$ , subject to the constraints  $|Q| \leq u_{max}^1 = 748 \text{ KJ/s}$ .
2. Configuration 2: Inlet stream temperature,  $u_2 = T_{A0} - T_{A0s}$ , subject to the constraints  $|u_2| \leq u_{max}^2 = 100 \text{ K}$ .
3. Configuration 3: Inlet reactant concentration,  $u_3 = C_{A0} - C_{A0s}$ , subject to the constraints  $|u_3| \leq u_{max}^3 = 4 \text{ mol/L}$ .

where configuration 2 will be used as the primary manipulated input.

To this end, we consider the chemical reactor operating under a given control configuration. At a certain time, one of the sensors fails in a way that it is imperative to recover the sensor to implement feedback control. The problem that we analyze is whether reactivating the original control configuration (after sensor recovery) guarantees closed-loop stability. We will next consider the problem where the sensors do not fail, however, the process experiences intermittent loss of measurements (and this rate increases at a certain time due to sampling/measurement/communication errors. In this case, how much measurement data loss can be tolerated by the currently active control configuration, before it becomes necessary to reconfigure, and, if necessary, which backup configuration should be activated in the closed-loop system. Note that while we use the simple chemical reactor example only to motivate our results, the scenarios that we describe are relevant to all process operations. We also include an application to a more realistic process example, a polyethylene reactor, on the second example.

## **6.3 Stabilization Subject to Sensor Failures**

In this section, we consider the problem arising out of sensor failures that lead to the failure of the control loop and necessitate recovery. In analyzing this problem and in devising the fault-tolerant control strategy, we account for the presence of nonlinearity and constraints and show how they impact the reconfiguration logic.

### **6.3.1 Reconfiguration Law**

Consider the closed-loop system of Equations 6.1-6.4 for which candidate control configurations have been identified and the stability region under each candidate con-



figuration has been explicitly characterized. Let the closed-loop system of Equations 6.1-6.4 be initialized under a configuration  $k$  with  $x_0 \in \Omega_k$ . Let  $T^f$  be the time at which the sensor fails and  $T^r$  be the time at which the sensor recovers. In the absence of measurements, the process runs open loop from the time  $T^f$  to  $T^r$ . Consequently, during this time the process state may drift further away from the desired operating condition. When the measurements become available again, switching to the original control configuration may not achieve closed-loop stability. The key consideration in devising the reconfiguration logic is the limitation imposed on the stability region under a given control configuration by the presence of input constraints and is formalized below:

**Theorem 6.2** *Let  $k(0) = i$  for some  $i \in \mathcal{K}$  and  $x(0) := x_0 \in \Omega_i$ . Let  $T^f$  be the time that the sensor measurements become unavailable and let  $T^r$  be the earliest time that they become available again. Then, the following switching rule:*

$$k(t) = \left\{ \begin{array}{ll} i, & 0 \leq t < T^f \\ l, & t \geq T^r, x(T^r) \in \Omega_l \end{array} \right\} \quad (6.5)$$

*guarantees asymptotically stabilization of the origin of the closed-loop system.*

**Proof of Theorem 6.2** We consider the two possible cases; first if no sensor failure occurs ( $T^f = \infty$ ), and second if a failure occurs at some finite time  $T^f$  and the sensors are recovered at time  $T^r$ .

*Case 1:* The absence of a failure implies  $k(t) = i \forall t \geq 0$ . Furthermore, since  $x(0) \in \Omega_i$ , and control configuration  $i$  is implemented for all times in this case, asymptotic stability follows from Theorem 6.1.

*Case 2:* At time  $T^r$ , the supervisor switches to a control configuration  $l$  for which  $x(T^r) \in \Omega_l$ . From this time onwards, since configuration  $l$  is implemented in the closed-loop system for all times, and since  $x(T^r) \in \Omega_l$ , once again, asymptotic stability follows from Theorem 6.1.

This completes the proof of Theorem 6.2.

**Remark 6.2** Theorem 6.2 accounts for the presence of constraints in the reconfiguration logic via the consideration of the stability region of candidate control configurations. Note that the problem that we consider here are sensor failures that result in loss of controllability. For the sake of illustration, consider a linear system of the form  $\dot{x} = Ax + Bu; y = Cx$ , where  $x$  is the state vector,  $y$  is the vector of measured variables and  $u$  is the vector of manipulated variables, with  $A$ ,  $B$  and  $C$  being matrices of appropriate dimensions. Consider the case when all state variables are being measured ( $C = I$ ), and a state feedback law of the form  $u = Ky = Kx$  is used to stabilize the system. Further let some of the sensors fail at some time, resulting in a new  $C$  matrix denoted by  $\bar{C}$ . The same feedback gain matrix  $K$  may no longer be stabilizing. If  $\bar{C}$  is such that it can be used to reconstruct (estimate) the unstable states of the system (that is, all the unstable states remain observable) then feedback control (with an observer, and with a different feedback gain matrix) can still be used to stabilize the system. However if  $\bar{C}$  is such that some of the unstable states of the system become unobservable, then the system simply cannot be stabilized using feedback control, and fixing the sensors becomes imperative. In other words, it is when measurements become unavailable (due to individual sensor malfunction, or loss of communication lines) that result in loss of controllability, that it becomes imperative to detect, isolate and correct the problem. Due to the open-loop behavior of the process during this intermediate time, the process states may drift and go out of the stability region of the currently active control configuration. Reactivating the original control configuration may therefore not stabilize the closed-loop system making it necessary to ascertain the suitability of a candidate control configuration by using Theorem 6.2 (see the simulation example for a demonstration).

**Remark 6.3** While in this chapter we do not focus on the problem of fault-detection and isolation (considering instead the problem of determining the corrective action that needs to be taken once the fault information is available), this problem has been

approached using a data-based or a model-based strategy. Statistical and pattern recognition techniques for data analysis and interpretation (for example, [144, 74, 5, 4, 112]), use past plant data to construct indicators that identify deviations from normal operation, and help in isolating faults. The problem of using fundamental process models for the purpose of detecting faults has been studied extensively in the context of linear systems [108, 60, 67]; and recently, some existential results in the context of nonlinear systems have been derived [129, 146, 38].

In Chapter 4 we proposed an integrated fault-detection and fault-tolerant control structure that handles faults in the control actuators under the assumption of continuous availability of state or output measurements. The fault-detection and isolation filter in Chapter 4 relies on the measurements to observe deviations of the process behavior from the expected closed-loop behavior to detect faults, and needs to be redesigned if required to detect and isolate faults in the sensors. While the problem of designing sensor fault-detection and isolation filter remains outside the scope of the work in this chapter, we note that the proposed fault-tolerant controller allows the use of any data- or model-based fault-detection and isolation filter to provide information about the occurrence of the fault (leading to its recovery). In this chapter we focus instead on determining what corrective action needs to be taken after a fault has been reported and how the time that it takes to recover the fault impacts on the reconfiguration logic. Specifically, the reconfiguration logic points to the necessity of recovering the sensor sufficiently fast to avoid the situation where the process state, by the time of recovery, has escaped the stability region of the backup configurations. Alternatively, the proposed method can also be used for the purpose of designing the control configurations in a way that maximizes the region in state space covered by the backup configurations to increase the chances that the process state at the time of recovery lies in the stability region of at least one backup configuration.

### 6.3.2 Application to Chemical Reactor

In this section, we illustrate the utility of the reconfiguration law of Equation 6.5. To this end, consider the chemical reactor of Equation 2.13 with the three candidate control configurations available. The first step in implementing the reconfiguration law of Equation 6.5 is that of determining the stability regions of the individual control configurations under the control law of Equations 6.2-6.4. An explicit characterization of the stability regions is obtained and is shown in Figure 6.1. The area indicated by I, II and III indicates the set of initial conditions starting from where all three configurations can stabilize the closed-loop system, I, II starting from where only configurations 1 and 2 can achieve stability and I, III indicate the set of initial conditions starting from where only configurations 1 and 3 can stabilize the closed-loop system.

The closed-loop system is initialized under configuration 2 from an initial condition belonging to the stability region of configuration 2. At  $t = 200$  min, however, a sensor failure occurs resulting in open-loop operation, and the process state begins to drift away from the desired equilibrium point (see dotted line in Figure 6.1). Recognizing that it is imperative to rectify this fault, the sensors are recovered (alternatively, redundant sensors are activated) at  $t = 220$  min. With the state information again available, if the original control configuration (configuration 2) is reactivated, closed-loop stability is not achieved (see dash-dotted lines in Figure 6.1). This happens because during the time that the process was running open-loop, the states of the closed-loop system moved away from the desired equilibrium point and out of the stability region of configuration 2. In contrast, if the reconfiguration law of Equation 6.5 is used, the law dictates activation of configuration 1 (since the process state, when state information becomes available again, lies in the stability region of configuration 1). Closed-loop stability is subsequently achieved (solid line in Figure 6.1).

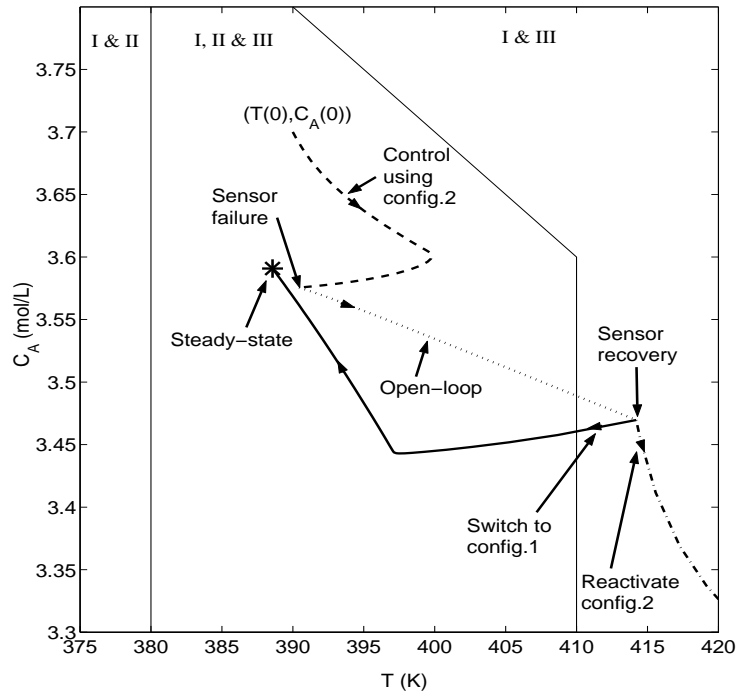


Figure 6.1: Evolution of the state profile under configuration 2 (dashed line) followed by loss of measurements (dotted line) and upon recovery reactivating configuration 2 (dash-dotted line), closed-loop stability is not preserved; however, switching to configuration 1 (solid line) preserves closed-loop stability.

Note that at the time the state information became available again, the state was also in the stability region of configuration 3, and switching to either configuration 1 or 3 would guarantee closed-loop stability. In such cases (when more than one control configurations satisfy the stability criteria), additional performance criteria, such as ease/cost of use can be used to decide which control configuration should be implemented in the closed-loop system (see Chapter 3).

## 6.4 Stabilization Subject to Sensor Data Losses

In the previous section, we considered the problem of devising the reconfiguration law in a way that accounts for the presence of constraints on the manipulated inputs under the available control configurations. We now consider the problem of intermittent sensor data losses (not complete failures) and develop a reconfiguration law that achieves fault-tolerant in the presence of sensor data-losses. As evidenced in the previous section, a prerequisite to implementing fault-tolerant control is the characterization of the stability properties under the available control configurations, which we undertake in this section, and in the next section present the reconfiguration law. We consider the closed-loop system of Equations 6.1-6.4 under a configuration  $k$  and drop the subscript  $k$  in the remaining of this section with the understanding that the robustness of the closed-loop system under control configuration  $k$  is being analyzed.

### 6.4.1 Modeling Sensor Data Loss

Preparatory to the analysis of the stability properties of the closed-loop system under sensor data losses, we describe how we model the occurrence of sensor data losses. Specifically, sensor data availability is modeled as a random Poisson process. At a given time  $t$  an ‘event’ takes place that determines whether the system will be

closed-loop or open-loop (see Figure 6.2). For a given rate of data loss  $0 \leq r \leq 1$ , a random variable  $P$  is chosen from a uniform probability distribution between 0 and 1. If  $P \leq r$ , the event is deemed to be ‘measurement loss’, while if  $P > r$ , the event is understood to be ‘measurement available’. Furthermore, with  $W$  defined as the number of events per unit time, another random variable  $\chi$  with uniform probability distribution between 0 and 1 determines the time for which the current event will last, given by  $\Delta = \frac{-\ln\chi}{W}$ . At  $t + \Delta$  another event takes place and whether it represents a measurement or loss of measurement, as well as its duration, is similarly determined. Note that in the presence of constraints, prolonged duration of measurement loss may land the system states at a point starting from where stabilization may not be achievable (even with continuous measurement); in characterizing the stability properties of constrained systems, we therefore need to define data loss rates over a finite time interval as stated in Assumption 6.1 below.

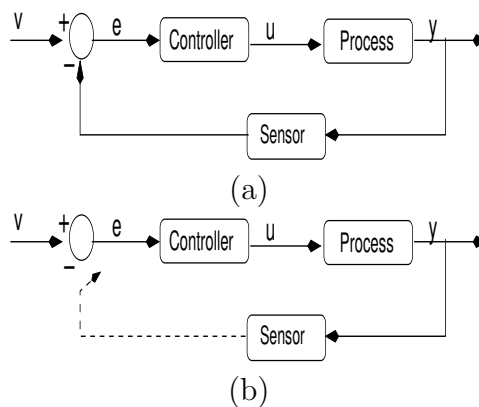


Figure 6.2: Closed-loop system in the (a) absence, and (b) presence of sensor data losses.

**Assumption 6.1** For a positive real number  $T^*$ , defining  $r \in [0, 1]$  as the sensor data loss rate implies that over every successive finite time interval  $T^*$ , the measurements are available for a total time of  $T^* \times (1 - r)$ .

Note that Assumption 6.1 does not impose any restrictions on the distribution of sequences of measurement loss and availability over the time interval  $T^*$ . Furthermore, the assumption does not need to hold for *any* finite interval  $T^*$  but only successive time intervals  $T^*$ . To illustrate the difference, consider the case where the assumption requires the data loss rate to hold over any finite time interval  $T^*$ , and that one such interval is  $\tau, \tau + T^*$ . Requiring the data loss rate to hold over any interval  $T^*$  would mean that the same data loss rate should also hold over the interval  $\tau + \epsilon_t, \tau + T^* + \epsilon_t$ , for *any* positive real number  $\epsilon_t$ , which can only be true if the data loss and measurement events are periodic with a period  $T^*$ . The requirement that the data loss rate hold over successive intervals  $T^*$  only says that over the time interval  $T^*$ , if the duration of all the measurement loss events is summed up, then that sum is equal to  $T^* \times r$ , and the data loss events could be distributed arbitrarily during this time interval. In simulating data losses, Assumption 6.1 can be practically realized by picking  $W$  to be sufficiently large; the reasoning behind this is as follows: a larger value of  $W$  increases the number of events per unit time, and when  $W$  is sufficiently large, we can get a sufficiently large number of events over every finite time interval  $T^*$  such that the rate of data loss is sufficiently close to  $r$ .

#### 6.4.2 Analyzing Closed-Loop Stability

In this section, we consider the closed-loop system subject to sensor data losses as defined in previous section, and analyze the stability properties (robustness) with respect to sensor data losses. Specifically, the objective is to establish, for convergence to a desired neighborhood of the origin, a data loss rate  $r^*$ , defined over a finite time interval  $T$ , such that if  $r \leq r^*$  then convergence to a desired neighborhood is achieved in the presence of data losses. Note that implicit in this analysis (also in the



formulation of Equation 6.1) is the understanding that during the time that sensor measurements are unavailable, the values of the measured variables (in computing the control action) are ‘frozen’ at the last available measurement. This results in the value of the manipulated variable being frozen at the last computed value. The implications of this intuitive assumption on the stabilizing properties under a given control configuration is discussed in Remark 6.5.

We first consider the closed-loop system under the controller of Equation 6.3, where the control action is computed in an implement and hold fashion with a hold time  $\Delta$ . We establish that for convergence to a desired neighborhood of the origin, there exists a bound on the implement and hold time  $\Delta^*$ , such that if the hold time is less than  $\Delta^*$ , then during the entire hold time, we get (outside of the desired neighborhood of the origin) that  $\dot{V} < 0$  (by virtue of the fact that the control action is ‘held’ at the value computed using the last available measurement) and eventual convergence to the desired neighborhood can be achieved. This analysis reveals that anytime the control action is ‘updated’ by using the current state value, the closed-loop Lyapunov-function decreases during the next  $\Delta$  (for  $\Delta \leq \Delta^*$ ) time. In essence, it reveals that the worst distribution of the measurement loss events, or the most destabilizing that they can be, would be if they were to occur consecutively. The sum of the duration of all the measurement loss events not being greater than  $r \times T^*$  over a finite time interval  $T^*$  can be exploited to yield the desired result which is formalized in Theorem 6.3 below.

**Theorem 6.3** *Consider the constrained system of Equation 6.1 under the bounded control law of Equations 6.2-6.4 designed using the Lyapunov function  $V$  and  $\rho > 0$ , and the stability region estimate  $\Omega$  under continuous implementation. Then, given any positive real number  $d$  such that  $\|x\| \leq d$  implies  $x \in \Omega$  and  $T^*$  over which a data loss rate  $r$  is defined, there exists a positive real number  $r^*$  such that if  $x(0) := x_0 \in \Omega$*

and is known, and  $r \in (0, r^*]$ , then  $x(t) \in \Omega \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$ .

**Proof of Theorem 6.3** The proof consists of two parts. In the first part, we assume that the measurement loss events occur consecutively, and show the existence of a bound on the data loss rate  $r^*$  below which convergence to the desired neighborhood is achieved. In part 2, we show that this result also holds for any distribution of the open loop events over the time interval  $T^*$ .

*Part 1:* Substituting the control law of Equations 6.2-6.4 into the system of Equation 6.1 it can be shown that:

$$\dot{V}(x) = -\rho^*V(x) \quad (6.6)$$

for all  $x \in \Omega$ , where  $\Omega$  was defined in Equation 6.4. Note that since  $V(\cdot)$  is a continuous function of the state, one can find a finite, positive real number,  $\delta'$ , such that  $V(x) \leq \delta'$  implies  $\|x\| \leq d$ . Consider now evolution of the states between the time 0 to  $T^*$ , where  $T^*$  is the time interval over which the data loss rate is defined, and for a given data loss rate  $r$ , denote the duration of open-loop operation as  $\Delta$ . In the rest of the proof, we show the existence of a positive real number  $\Delta^*$  such that all state trajectories originating in  $\Omega$  converge to the level set of  $V$  ( $V(x) \leq \delta'$ ) for any value of  $\Delta \in (0, \Delta^*]$ . Hence we have that  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$ . We then use the definition of the data loss rate to come up with an  $r^*$  to show that the result holds for any  $r \leq r^*$ .

To this end, consider a “ring” close to the boundary of the stability region, described by  $\mathcal{M} := \{x \in \mathbb{R}^n : (c^{max} - \delta) \leq V(x) \leq c^{max}\}$ , for a  $0 \leq \delta < c^{max}$ . Let the control action be computed for some  $x(0) := x_0 \in \mathcal{M}$  and, upon unavailability of subsequent measurements, held constant until a time  $\Delta^{**}$ , where  $\Delta^{**}$  is a positive real number ( $u(t) = u(x_0) := u_0 \forall t \in [0, \Delta^{**}]$ ) to be determined. Then,  $\forall t \in [0, \Delta^{**}]$ ,

$$\begin{aligned} \dot{V}(x(t)) &= L_f V(x(t)) + L_G V(x(t))u_0 \\ &= L_f V(x_0) + L_G V(x_0)u_0 \\ &\quad + (L_f V(x(t)) - L_f V(x_0)) \\ &\quad + (L_G V(x(t))u_0 - L_G V(x_0)u_0) \end{aligned} \quad (6.7)$$

Since the control action is computed based on the states in  $\mathcal{M} \subseteq \Omega$ ,  $L_f V(x_0) + L_G V(x_0)u_0 \leq -\rho^* V(x_0)$ . By definition, for all  $x_0 \in \mathcal{M}$ ,  $V(x_0) \geq c^{max} - \delta$ , therefore  $L_f V(x_0) + L_G V(x_0)u_0 \leq -\rho^*(c^{max} - \delta)$ .

Since the function  $f(\cdot)$  and the elements of the matrix  $G(\cdot)$  are continuous,  $\|u\| \leq u^{max}$ ,  $\mathcal{M}$  is bounded and  $L_f V(\cdot)$ ,  $L_G V(\cdot)$  are Lipschitz, then one can find, for all  $x_0 \in \mathcal{M}$ , positive real numbers  $\Delta^{**}$ ,  $K^1$ ,  $K^2$  and  $K^3$  such that  $\|x(\tau) - x_0\| \leq K^1 \Delta^{**}$  for all  $\tau \leq \Delta^{**}$ ,  $\|L_f V(x(\tau)) - L_f V(x_0)\| \leq K^3 K^1 \Delta^{**}$ ,  $\|L_G V(x(\tau))u_0 - L_G V(x_0)u_0\| \leq K^2 K^1 \Delta^{**}$  for all  $\tau \leq \Delta^{**}$ , and  $\Delta^{**} < \frac{\rho^*(c^{max} - \delta) - \epsilon}{(K^1 K^2 + K^1 K^3)}$  where  $\epsilon$  is a positive real number such that

$$\epsilon < \rho^*(c^{max} - \delta) \quad (6.8)$$

Using these inequalities in Equation 6.7, we get

$$\dot{V}(x(\tau)) \leq -\epsilon < 0 \quad \forall 0 \leq \tau \leq \Delta^{**} \quad (6.9)$$

This implies that, given  $\delta'$ , if we pick  $\delta$  such that  $c^{max} - \delta < \delta'$  then if the control action is computed for any  $x \in \mathcal{M}$ , and the measurement loss time is less than  $\Delta^{**}$ , we get that  $\dot{V}$  remains negative during this time, and therefore the state of the closed-loop system cannot escape  $\Omega$  (since  $\Omega$  is a level set of  $V$ ). We now show the existence of  $\Delta'$  such that for all  $x_0 \in \Omega^f := \{x \in \mathbb{R}^n : V(x_0) \leq c^{max} - \delta\}$ , we have that  $x(\Delta) \in \Omega^u := \{x_0 \in \mathbb{R}^n : V(x_0) \leq \delta'\}$ , where  $\delta' < c^{max}$ , for any  $\Delta \in (0, \Delta']$ .

Consider  $\Delta'$  such that

$$\delta' = \max_{V(x_0) \leq c^{max} - \delta, u \in \mathcal{U}, t \in [0, \Delta']} V(x(t)) \quad (6.10)$$

Since  $V$  is a continuous function of  $x$ , and  $x$  evolves continuously in time, then for any value of  $\delta < c^{max}$ , one can choose a sufficiently small  $\Delta'$  such that Equation 6.10 holds. Let  $\Delta^* = \min\{\Delta^{**}, \Delta'\}$ . We now show that for all  $x_0 \in \Omega^u$  and  $\Delta \in (0, \Delta^*]$ ,  $x(t) \in \Omega^u$  for all  $t \geq 0$ .

For all  $x_0 \in \Omega^u \cap \Omega^f$ , by definition  $x(t) \in \Omega^u$  for  $0 \leq t \leq \Delta$  (since  $\Delta \leq \Delta'$ ). For all  $x_0 \in \Omega^u \setminus \Omega^f$  (and therefore  $x_0 \in \mathcal{M}$ ),  $\dot{V} < 0$  for  $0 \leq t \leq \Delta$  (since  $\Delta \leq \Delta^{**}$ ). Since  $\Omega^u$  is a level set of  $V$ , then  $x(t) \in \Omega^u$  for  $0 \leq t \leq \Delta$ .

We note that for  $x$  such that  $x \in \Omega \setminus \Omega^u$ , negative definiteness of  $\dot{V}$  is guaranteed for  $\Delta \leq \Delta^* \leq \Delta^{**}$ . Finally, for all  $\Delta^* \leq t \leq T^*$ , negative definiteness of  $\dot{V}$  is guaranteed by the control law of Equation 6.3. Now for a given value of  $T^*$ , the worst case scenario (that is, the maximum time over which the system may run open-loop) involves loss of measurements for the last  $\Delta$  time for a given interval, followed by consecutive loss of measurements for the first  $\Delta$  time of the next interval. Therefore, continued negative definiteness of  $V$  (and convergence to the desired neighborhood) can be guaranteed if the measurement loss time in each interval  $\Delta \leq \frac{\Delta^*}{2}$ . An  $r^* = \frac{\Delta^*}{2T^*}$  will ensure that the maximum duration of measurement loss over the interval  $T^*$  is less than  $\Delta^*/2$ , and also maximum loss of measurement between two successive intervals is less than  $\Delta^*$  ( If  $\frac{\Delta^*}{2} > T^*$ , then we have to restrict  $r^*$  to 1 to ensure that  $r < 1$  and that we get at least one measurement over the entire interval  $T^*$ ). Therefore, for all  $x(0) \in \Omega$ , there exists an  $r^*$  such that if  $r \leq r^*$ ,  $\limsup_{t \rightarrow \infty} V(x(t)) \leq \delta'$ . Finally, since  $V(x) \leq \delta'$  implies  $\|x\| \leq d$ , therefore we have that  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d$ .

*Part 2:* Consider now the finite time interval  $T^*$ , such that for convergence to a desired neighborhood of the origin, the bound on the data loss rate  $r^*$ , under the assumption that the data-loss events all occur consecutively, has been computed. Consider now that the data-loss events do not occur continuously, but occur in  $N$  intervals, each of duration  $\Delta_i$  with  $\sum_{i=1}^N \Delta_i = T^* \times r^*$ . From part 1 above, for each of these durations  $\Delta_i$ , negative definiteness of  $\dot{V}$  can be established. For the duration during which the measurements are available,  $\dot{V} < 0$  is achieved by virtue of the control law. In summary, having established the bound  $r^*$  under consecutive loss of measurement, the same bound  $r^*$  continues to guarantee practical stability irrespective of the distribution of the measurement loss events.

This completes the proof of Theorem 6.3.

**Remark 6.4** Note that one can easily remove the assumption that  $x_0$  is known by ‘stepping back’ from the boundary of the stability region enough to ensure that during the time  $r^*T^*$ , the state trajectory cannot escape the boundary of the stability region. By the definition of rate of data loss, the first measurement is guaranteed to be available by  $(r^*T^*)^+$ . Any time during the interval  $T^*$  that a measurement is received with the state still residing in the stability region (due to the ‘stepping back’) Theorem 6.3 can be used to establish practical stability. Note also, that the value of  $r^*$  depends on the interval  $T^*$  over which it is defined (see the simulation example in section 6.4.3 for a demonstration). To understand this more clearly, let us revisit the proof of Theorem 6.3. It can be seen that for convergence to a desired neighborhood of the origin, one can come up with a value  $\Delta^*$  such that if only one measurement was received every  $\Delta^*$ , then convergence to the desired neighborhood would be achieved. Theorem 6.3 exploits this fact together with the definition of the data loss rate, to ensure that over a  $\Delta^*$  duration within  $T^*$  (and across two time intervals), at least one measurement is received. In summary,  $\Delta^*$  is fixed by the given size of the neighborhood to the origin where convergence is desired ( $\delta'$ ); given a  $T^*$  over which the data loss rate is defined,  $r^*$  can then in turn be picked such that the maximum duration of open-loop behavior across intervals stays less than  $\Delta^*$ .

**Remark 6.5** In our results, no bound on the open-loop instability is assumed to be known, leading to practical (and not asymptotic) stability to the desired equilibrium point. If additional assumptions are made on the open-loop growth of the Lyapunov-function (locally) around the desired equilibrium point, asymptotic stability can be shown using the same line of reasoning as in [75]. Specifically, during the time that the measurements are not available, the value of  $V$  is allowed to increase during  $T^*$ , so long as the increase in  $V$  can be ‘countered’ by the decrease in  $V$  during the rest of the time (which relies on assuming a known measure of open-loop instability). The limitations imposed by the presence of constraints, however, would still need to be accounted for, with the data loss rate having to be defined over a finite interval. Furthermore, the set of stabilizable initial conditions will only be a subset of  $\Omega$  such that starting from this

subset, the closed-loop state can not escape  $\Omega$  during the time of open-loop evolution  $r^*T^*$ . In our results, with  $x_0$  known,  $r^*$  is picked so that  $\dot{V}$  stays negative during the entire duration of  $T^*$  (until convergence to the desired neighborhood is achieved), thereby obviating the need to restrict the set of initial conditions to a subset of  $\Omega$ . Note also that  $V$  being allowed to increase during  $T^*$  (as long as it decays by the end of  $T^*$ ) could possibly lead to a larger allowable  $r^*$ . The tradeoff would be that the Lyapunov function would not be guaranteed to decay all the time but only to decay in value at steps of  $T^*$ , and it could take longer to reach the desired neighborhood of the origin. Note that the problem considered in this chapter is not that of ascertaining finite-time stability (ensuring convergence to the desired equilibrium point in finite time, see, for example, [17]) under continuous availability of measurement but rather that of analyzing preservation of stability under asynchronous measurements. Note that for the case when sensor measurements are lost but it is possible to change the value of the manipulated input, statistical (for example, [127]) or first principles model based methods designed to ‘fill-in’ the unavailable state measurement can very well be included within the proposed framework, and can serve to improve the data-loss handling capabilities of the control designs (depending upon the accuracy of the data prediction). The proposed fault-tolerant control structure, however, addresses a more general problem, that of intermittent loss of communication between the controller and the process, including asynchronous measurements as well as the inability to change the manipulated input value during the communication lapses.

**Remark 6.6** The proof of theorem 6.3 relies on the stabilizing properties of the controller during the time that measurements are not available to ensure that even during that time,  $\dot{V} < 0$ . Note that the rate of decay of the Lyapunov function that is achieved under continuous measurements is closely related to how much data loss can be tolerated in the system in the sense that for a given process and constraints on the manipulated inputs, if one control law achieves greater decay of the Lyapunov function over the other, then it can tolerate greater sensor data loss compared to the other (note that the tradeoff could be a smaller stability region estimate). The

continued decay of the Lyapunov function, however, can only be achieved over a finite time, and in turn, requires the data loss rate to be defined over a finite time. Even if one were to use the approach discussed in Remark 6.4 to come up with an alternate bound, the limitations imposed by the constraints on the definition of the rate of data loss (specifically, the need to define it over a finite time interval) would be present and can be understood as follows: If there were no constraints,  $\dot{V} < 0$  under continuous measurement could possibly be achieved over the entire state space. No matter how ‘far’ the states go during the unavailability of measurements, when (over the infinite time duration) the measurements do become available, one could require them to be available for a large enough time (compared to the time during which they were not available) to achieve an overall reduction in the value of the Lyapunov function. Constraints, however, limit the set of initial conditions (estimated using the stability region  $\Omega$ ) starting from where  $\dot{V} < 0$  is achievable. If the measurements are not available for a large duration, the states may go too ‘far’ (that is, out of the stability region) and then even if measurements were available for all time after that,  $\dot{V} < 0$  could not be achieved simply due to limited available control action (see the simulation example for a demonstration). In contrast, defining the data loss rate over a finite time interval enables restricting the states to stay within the region from where  $\dot{V} < 0$  and hence closed-loop stability is achievable.

**Remark 6.7** Note that the specific problem that this chapter considers yields a solution that is essentially different from, and cannot be handled by simply using adaptive or other robust control approaches. These approaches, however, can very well be integrated within the proposed framework. The key requirement being that the controller design (whether it be an adaptive control design or another robust controller design) for the individual control configuration allow for an explicit characterization of its stability properties in the presence of input constraints and asynchronous data losses. It is this characterization that can be subsequently used in fault-tolerant reconfiguration strategies. Note also that multi-rate data loss problems, where data is available at predetermined (but different) times for the different measurements can be analyzed

as special cases for the problem considered in this chapter which does not assume data availability at predetermined rates.

### 6.4.3 Control of a Chemical Reactor Subject to Sensor Data Loss

Consider the chemical reactor of Equation 2.13 again with the inlet stream temperature, as the manipulated input  $u_2 = T_{A0} - T_{A0s}$ , subject to the constraints  $|u_2| \leq u_{max}^2 = 100$  K, and subject to measurement data losses. We first design the bounded controller and estimate the stability region (see Figure 6.3). For a given value of  $T^* = 10$  minutes, we pick a value of  $W = 10$  events per minute (the simulations are run as discussed in Section 6.4.1); which yields an overall event rate of  $1/W$  that is, about one event every six seconds (or about 100 events in 10 minutes). It was verified that with this value of  $W$ , the rate of data loss, as defined, was approximately achieved over the duration of every ten minutes, in other words, that  $W = 10$  is a sufficiently large value of  $W$ . Starting from an initial condition within the stability region of the first configuration, the closed-loop system is unstable with a data loss rate  $r = 0.4$  (dashed lines in Figure 6.3; the corresponding manipulate input profile can be seen in Figure 6.4). However, if the data loss rate is kept at 0.1, closed-loop stability is achieved (see solid lines in Figures 6.3-6.4), demonstrating the need for the data loss to be sufficiently small.

The next simulation run demonstrates the dependence of  $r^*$  on the time interval over which it is defined (as discussed in Remark 6.6). Specifically, we now run the same simulation with an even smaller data loss rate ( $r = 0.05$ ), however, with the data rate defined over the duration of the simulation of 68 minutes. A scenario where measurements are received continuously for the first five minutes, lost consecutively for the next 3.6 minutes, and received thereafter results in an overall rate of data loss



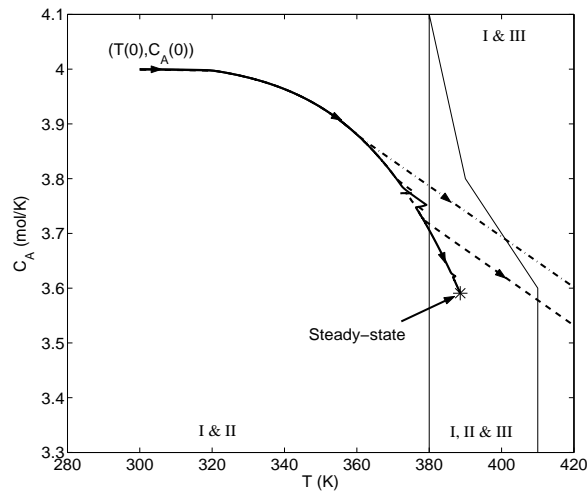


Figure 6.3: Evolution of the state trajectory under control configuration 2 in the presence of sensor data loss (defined over a finite interval) at a rate of 0.4 (dashed line), sensor data loss (defined over an infinite interval) at a rate of 0.05 (dash-dotted line) and sensor data loss (defined over a finite interval) at a rate of 0.1 (solid line).

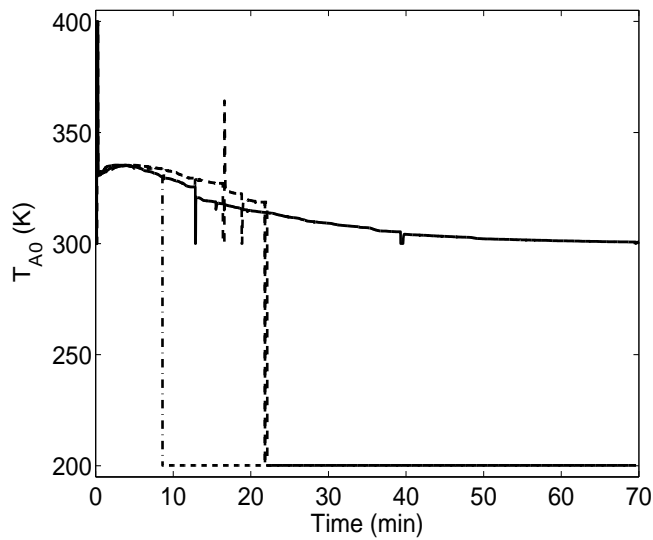


Figure 6.4: Manipulated input profile under control configuration 2 in the presence of sensor data loss (defined over a finite interval) at a rate of 0.4 (dashed line), sensor data loss (defined over an infinite interval) at a rate of 0.05 (dash-dotted line) and sensor data loss (defined over a finite interval) at a rate of 0.1 (solid line).

of only 0.05. We see however, that closed-loop stability is not achieved (dash-dotted lines in Figures 6.3-6.4). This is so because with this larger value of  $T^*$ , the acceptable bound on the rate of data loss decreases, and illustrates the interconnection between the maximum allowable data loss rate and the interval over which it is defined. In summary, the above simulations demonstrate the need for the data loss rate to be less than what the system can tolerate (that is, for  $r \leq r^*$ ), with  $r^*$  appropriately computed for a given time interval  $T^*$  over which the rate is defined.

## 6.5 Fault-Tolerant Control Subject to Sensor Data Losses

Having analyzed the stability properties of the individual control configurations subject to sensor data losses, in this section we present a fault-tolerant controller that maintains closed-loop stability in the presence of sensor data losses.

### 6.5.1 Reconfiguration law

Fault-tolerance is achieved via switching to a backup configuration for which the state of the closed-loop system is within the stability region, and the sensor data loss rate is less than the bound on the data loss rate required for closed-loop stability. To formalize this idea, consider the constrained nonlinear system of Equation 6.1 for which the bounded controllers of the form of Equation 6.3 have been designed and the stability regions  $\Omega_j$ ,  $j = 1, \dots, N$  have been explicitly characterized under each control configuration, and the bounds on the data loss rate  $r_j^*$ ,  $j = 1, \dots, N$  have been computed. Let  $d_{max} = \max_{j=1, \dots, N} d_j$ , where  $d_j$  was defined in Theorem 6.3 and let  $\Omega_U = \bigcup_{j=1}^N \Omega_j$ . We consider the problem where the process starts operating under configuration  $i$  with a data loss rate of  $r_i(0)$ , and at some point in time the data loss rate  $r(t)$  possibly becomes greater than  $r_i^*$ .

**Theorem 6.4** Let  $k(0) = i$  for some  $i \in \mathcal{K}$  and  $x(0) := x_0 \in \Omega_i$ . Let  $T^f$  be the earliest time such that  $r(t) > r_i^*$  with  $x(T^f)$  measured. Then, the following switching rule:

$$k(t) = \left\{ \begin{array}{ll} i, & 0 \leq t < T^f \\ l, & t \geq T^f, x(T^f) \in \Omega_l, r(T^f) \leq r_l^* \end{array} \right\} \quad (6.11)$$

and  $r(t) \leq r_i^* \forall t \geq T^f$  guarantees that  $x(t) \in \Omega_U \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_{max}$ .

**Proof of Theorem 6.4** We consider the two possible cases; first if the data loss rate  $r$  stays less than or equal to  $r_i^*$  for all times, and second if  $r > r_i^*$  at some time  $T^f$ .

*Case 1:* The absence of a switch implies  $k(t) = i \forall t \geq 0$ . Furthermore, since  $x(0) \in \Omega_i$ ,  $r(t) \leq r_i^*$  and control configuration  $i$  is implemented for all times in this case, we have that  $x(t) \in \Omega_i \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_i$ . Finally, since  $\Omega_i \subseteq \Omega_U$  and  $d_i \leq d_{max}$ , we have that  $x(t) \in \Omega_U \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_{max}$ .

*Case 2:* At time  $T^f$ , the supervisor switches to a control configuration  $l$  for which  $x(T^f) \in \Omega_l$  and  $r \leq r_l^*$ . From this time onwards, since configuration  $l$  is implemented in the closed-loop system for all times, and since  $x(T^f) \in \Omega_l$  and  $r(t) \leq r_l^*$ , we have that  $x(t) \in \Omega_l \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_l$ . As in case 1, since  $\Omega_l \subseteq \Omega_U$  and  $d_l \leq d_{max}$ , we have that  $x(t) \in \Omega_U \forall t \geq 0$  and  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq d_{max}$ .

This completes the proof of Theorem 6.4.

**Remark 6.8** Theorem 6.4 explicitly takes into consideration the constraints in the manipulated inputs and the measurement losses in deciding which backup configuration to implement in the closed-loop system, and therefore requires that a backup configuration is implemented for which the state resides in its stability region *and* the data loss rate is less than the data loss rate that the backup configuration can tolerate. Disregarding either of these factors could lead to instability (see the simulation example for a demonstration).

**Remark 6.9** Note that the result of Theorem 6.4 assumes explicit knowledge of the current data loss rate to not only identify the appropriate backup configuration

but also to trigger reconfiguration. In this sense, the reconfiguration logic has an in-built fault detection mechanism, with faults being defined as data loss rate exceeding the allowable data loss rate. In practice, the data loss rate can only be estimated over finite intervals of time, and this estimate can be used in deciding which backup configuration should be activated according the reconfiguration rule of Theorem 6.4. Note also, that other than the data loss rate (estimate) going over the allowable bound, other means of detecting instability like behavior (such as the state trajectory going close to the boundary of the stability region under the currently-active control configuration) can be used to trigger the reconfiguration. It is worth pointing out, however, that this fault-detection capability is only limited to the rate of data loss exceeding the tolerable value. As discussed in Remark 6.3, explicit fault detection mechanisms which detect faults in the sensors (such as sensors reporting incorrect values) can be used within the proposed approach to tackle sensor faults manifested as erroneous measurements.

**Remark 6.10** While we assume the availability of measurements of all the state variables, the same approach can be used to analyze the case where each control configuration is comprised of a set of sensors and actuators with the sensors (measurements) different in different control configurations. Specifically, under each control configuration, an estimation scheme, coupled with the feedback controller, will have to be implemented and the output feedback stability region, subject to constraints and sensor data losses characterized. Subsequently, the reconfiguration rule will have to be modified to account for the fact that the reconfiguration decision is made on the basis of state estimates (which may contain errors); for a switching scheme that addresses these issues in the context of switched nonlinear systems under continuous output feedback control, see [56].

### 6.5.2 Fault-Tolerant Control of a Chemical Reactor

Consider, once again the chemical reactor of section 6.4.3 in the presence of sensor data losses. As seen in section 6.4.3, the closed-loop system using configuration 2 experiences instability when the data loss rate becomes 0.4. In the event of such data losses, one of the backup control configurations need to be activated and this choice cannot be made only by looking at the states with respect to the stability region. In this section we demonstrate the application of the switching rule of Theorem 6.4 that achieves fault-tolerance. To this end, we first characterize the stability region under each backup configuration. Figure 6.5 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 configurations 1, 2 and 3, the bound on the data loss rate is estimated at  $r_1^* = 0.35$ ,  $r_2^* = 0.3$  and  $r_3^* = 0.15$ , respectively.

We consider an initial condition,  $T(0) = 300\text{ K}$ ,  $C_A(0) = 4.0\text{ mol/L}$ ,  $C_B(0) = 0.0\text{ mol/L}$ , using the  $T_{A0}$ -control configuration within the stability region of configuration 2, and consider a case where the rate of sensor data loss increases from an initial value of 0.1 to 0.35. As shown by the solid line in Figure 6.5, the controller proceeds to drive the closed-loop trajectory towards the desired steady-state, up until time 13.5 minutes of reactor startup when the sensor data loss rate increases to 0.35. If the supervisor does not use the result of Theorem 6.4 to trigger reconfiguration, but persists with using configuration 2, stability is not achieved (see dotted lines in Figures 6.5-6.6). Note that at this time, the state of the closed-loop system resides in the stability region of both backup configurations 1 and 3. If the supervisor does implement reconfiguration, but in a way that does not account for the presence of sensor data loss and activates configuration 3, the state trajectory does not converge

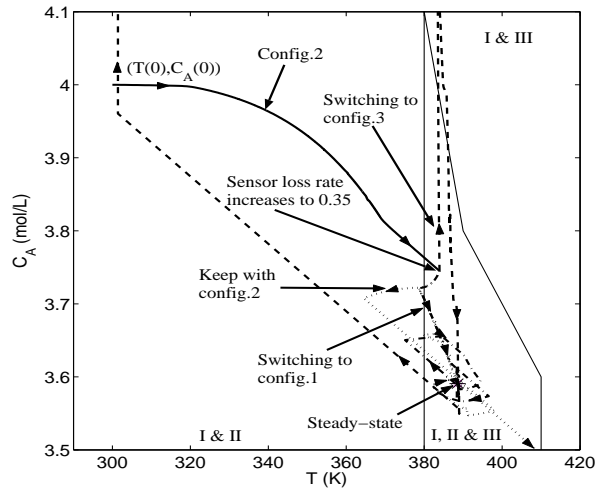


Figure 6.5: Evolution of the state trajectory: At  $t = 13.5$  minutes the data loss rate goes up to 0.35 under configuration 2 (solid line). Keeping with configuration 2 (dotted line) or switching to configuration 3 (dashed line) does not preserve stability, while switching to configuration 1 (dash-dotted line) preserves stability.

to the desired steady-state (see dashed line in Figure 6.5) even though the state at the switching time is within stability region of control configuration 3. This happens because the rate of data loss is not within the tolerable bound for configuration 3. In contrast, if the reconfiguration rule of Equation 6.11 is implemented, and the supervisor activates configuration 1, the state trajectory converges to the desired steady-state (see dashed-dotted line in Figure 6.5). The corresponding manipulated input profiles are shown in Figure 6.6.

### 6.5.3 Fault-Tolerant Control of a Polyethylene Reactor Subject to Sensor Data Loss

Having demonstrated the application of the proposed fault-tolerant controller on the illustrative example, we next consider a more complex process, specifically, an indus-

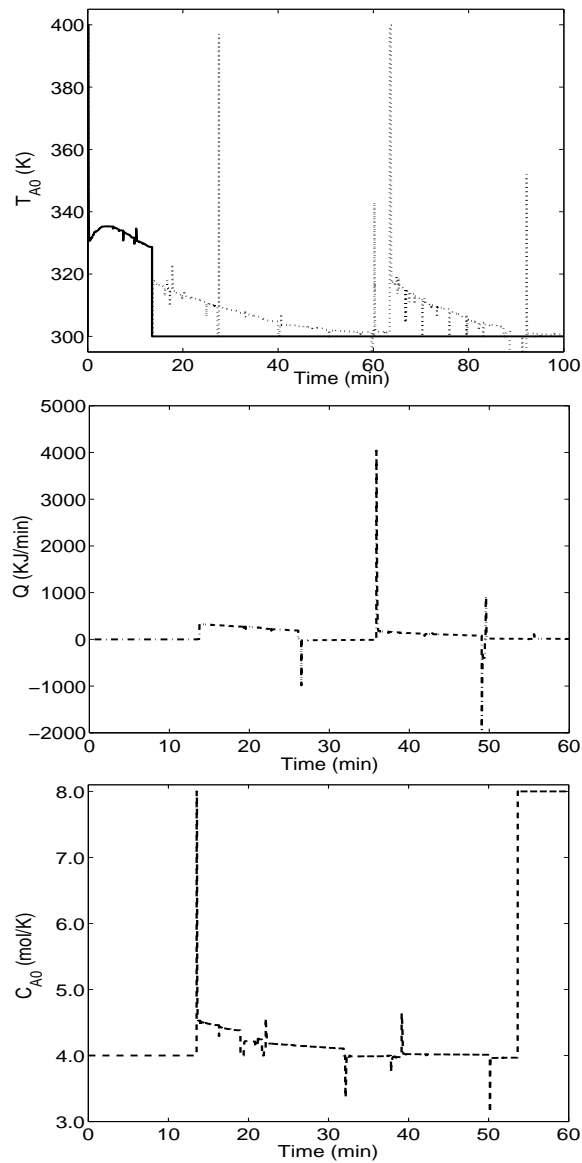


Figure 6.6: Manipulate input profiles: At  $t = 13.5$  minutes the data loss rate goes up to 0.35 under configuration 2 (solid line), switching to configuration 3 does not preserve stability (dashed line), while switching to configuration 1 (dash-dotted line) preserves stability.

trial gas phase polyethylene reactor system (see Figure 5.1). This reactor was also studied in Chapter 5 in the context of faults in the control actuator (under assumption of continuous availability of process measurements). Mathematical model for this reactor is described in Equations 5.1-5.2. For the definition of all the variables and the values of the process parameters, please refer to Chapter 5. The open-loop system at the nominal operating condition exhibits an unstable equilibrium point surrounded by a limit cycle. The control objective is to stabilize the reactor using measurements of the state variables. To accomplish this objective we consider the following manipulated input candidates:

1. Catalyst flowrate,  $u_1 = (F_c - F_c^s)a_c$ , subject to the constraint  $|u_1| \leq u_{max}^1 = (\frac{2}{3600})a_c \frac{mol}{s}$ .
2. Feed temperature,  $u_2 = \frac{F_{M_1}C_{pm1}+F_{In}C_{pIn}}{M_rC_{pr}+B_wC_{ppol}}(T_{feed} - T_{feed}^s)$ , subject to the constraint  $|u_2| \leq u_{max}^2 = \frac{F_{M_1}C_{pm1}+F_{In}C_{pIn}}{M_rC_{pr}+B_wC_{ppol}}(20) \frac{K}{s}$ .

First, process operation under primary control configuration was considered (that is, the catalyst flowrate,  $F_c$ , was the manipulated input) and a bounded nonlinear controller was designed using the formula of Equations 6.2-6.4. Specifically, a quadratic function of the form  $V_1 = e_1^T P_1 e_1$  and  $\rho_1 = 0.01$  were used to design the controller and a composite Lyapunov function of the form  $V_{c_1} = 5 \times 10^{-3}(In - In_s)^4 + 5 \times 10^{-4}(M_1 - M_{1s})^2 + 5 \times 10^{-11}(Y_1 - Y_{1s})^2 + 5 \times 10^{-11}(Y_2 - Y_{2s})^2 + 5 \times 10^{-4}(T - T_s)^2 + 5 \times 10^{-11}(T_{w_1} - T_{w_{1s}})^2 + 5 \times 10^{-11}(T_{g_1} - T_{g_{1s}})^2$  was used to estimate the stability region of the primary control configuration yielding a  $c_1^{max} = 56.8$ . A quadratic Lyapunov function of the form  $V_2 = \frac{1}{2}(T - T_s)^2$  and  $\rho_2 = 0.01$  were used to design the controller that used the fall-back control configuration (that is, the feed temperature,  $T_{feed}$ , was the manipulated input) and a composite Lyapunov function of the form  $V_{c_2} = 5 \times 10^{-3}(In - In_s)^4 + 5 \times 10^{-4}(M_1 - M_{1s})^2 + 5 \times 10^{-11}(Y_1 - Y_{1s})^2 + 5 \times$



$10^{-11}(Y_2 - Y_{2s})^2 + 5 \times 10^{-4}(T - T_s)^2 + 5 \times 10^{-2}(T_{w_1} - T_{w_{1s}})^2 + 5 \times 10^{-11}(T_{g_1} - T_{g_{1s}})^2$  was used to estimate the stability region of the fall-back control configuration yielding a  $c_2^{max} = 62$ .

Figure 6.7 shows the evolution of the closed-loop state profiles under continuous measurement (solid lines) starting from the initial condition  $In(0) = 450 \frac{mol}{m^3}$ ,  $M_1(0) = 340 \frac{mol}{m^3}$ ,  $Y_1(0) = 4.6 mol$ ,  $Y_2(0) = 4.6 mol$ ,  $T(0) = 360 K$ ,  $T_{w_1}(0) = 300 K$ , and  $T_{g_1}(0) = 300 K$  for which  $V_{c_1} = 56.78$ . Since this initial state is within the stability region of the primary control configuration (that is,  $V_{c_1}(x(0)) \leq c_1^{max}$ ), the primary control configuration is able to stabilize the system at the steady-state of interest. The corresponding manipulated inputs are shown on Figures 6.8-6.9. The dynamics of the process also reveal an important feature regarding tolerance to sensor data losses. Specifically, for this particular process, even under no control (equivalent to complete data loss), the process goes to a limit cycle which is within the stability region for the closed-loop system under continuous availability of measurements. This characteristic impacts positively on the tolerance of the closed-loop system to data losses, and a high sensor data loss rate of 0.75 ends up being tolerable (see dotted lines in Figures 6.7 and 6.9), even with the value of the manipulated input variable set to the nominal value during the time that the measurements are unavailable (equivalent to open-loop operation).

Consider now a case where the rate of sensor data loss increases from an initial value of 0.75 to 0.80 at 0.97 hour of reactor startup. As shown by the dashed lines in Figure 6.10, the controller proceeds to drive the closed-loop trajectory towards the desired steady-state up until 0.97 hours. If the supervisor does not account for the increase of sensor data loss and continues utilizing the primary control configuration to control the reactor, the state trajectory does not converge to the desired steady-state

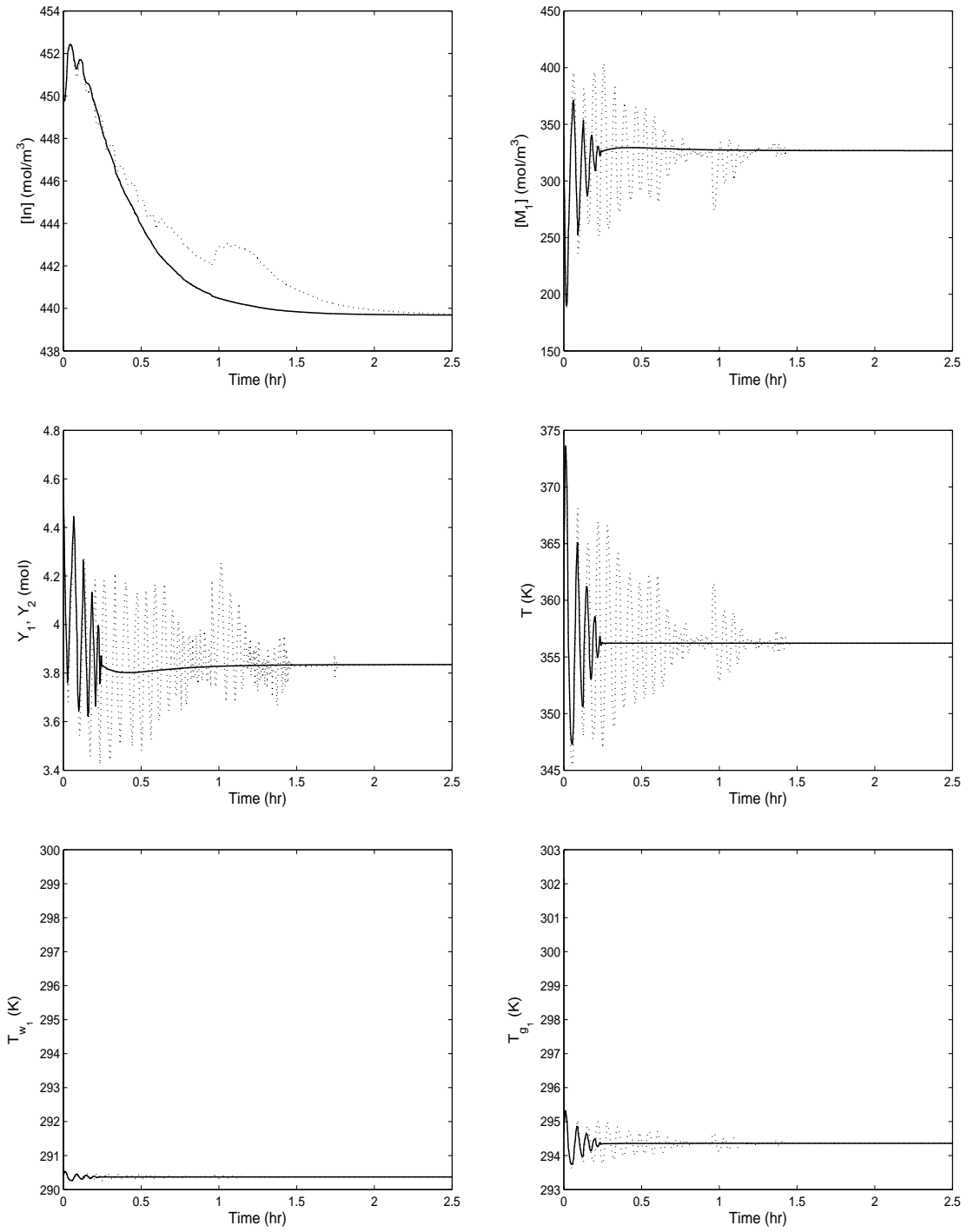


Figure 6.7: Evolution of the closed-loop state profiles under primary control configuration under continuous measurements (solid lines) and sensor data loss rate of 0.75 (dotted lines).

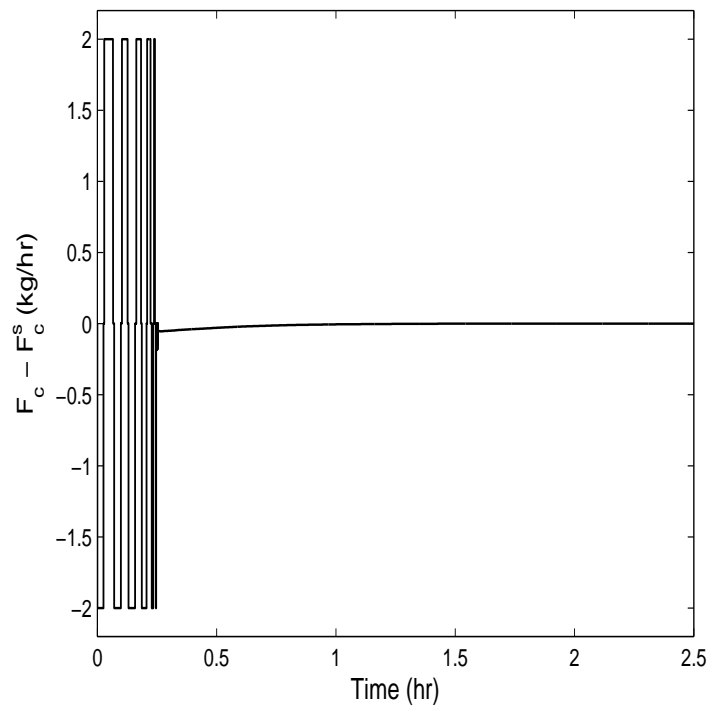


Figure 6.8: Evolution of the manipulated input profiles under primary control configuration under continuous measurements.

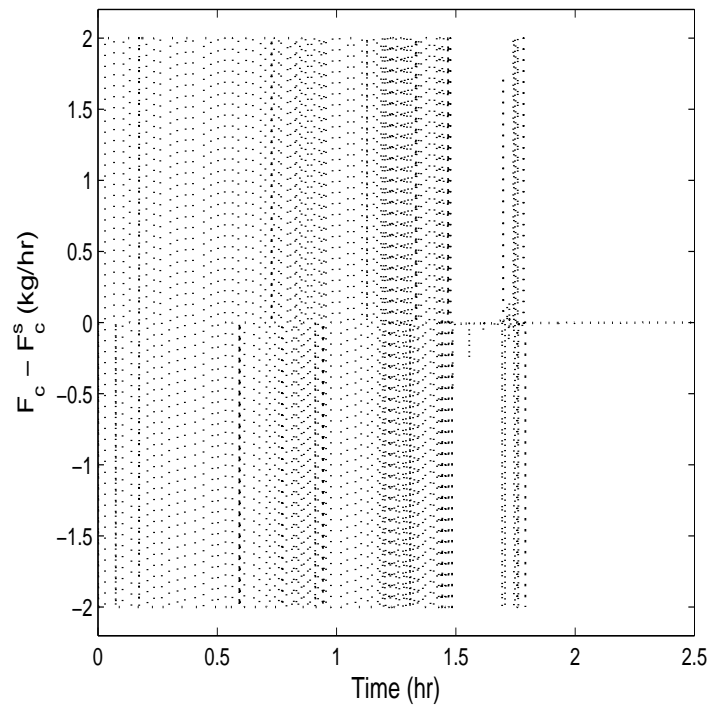


Figure 6.9: Evolution of the manipulated input profiles under primary control configuration with sensor data loss rate of 0.75.

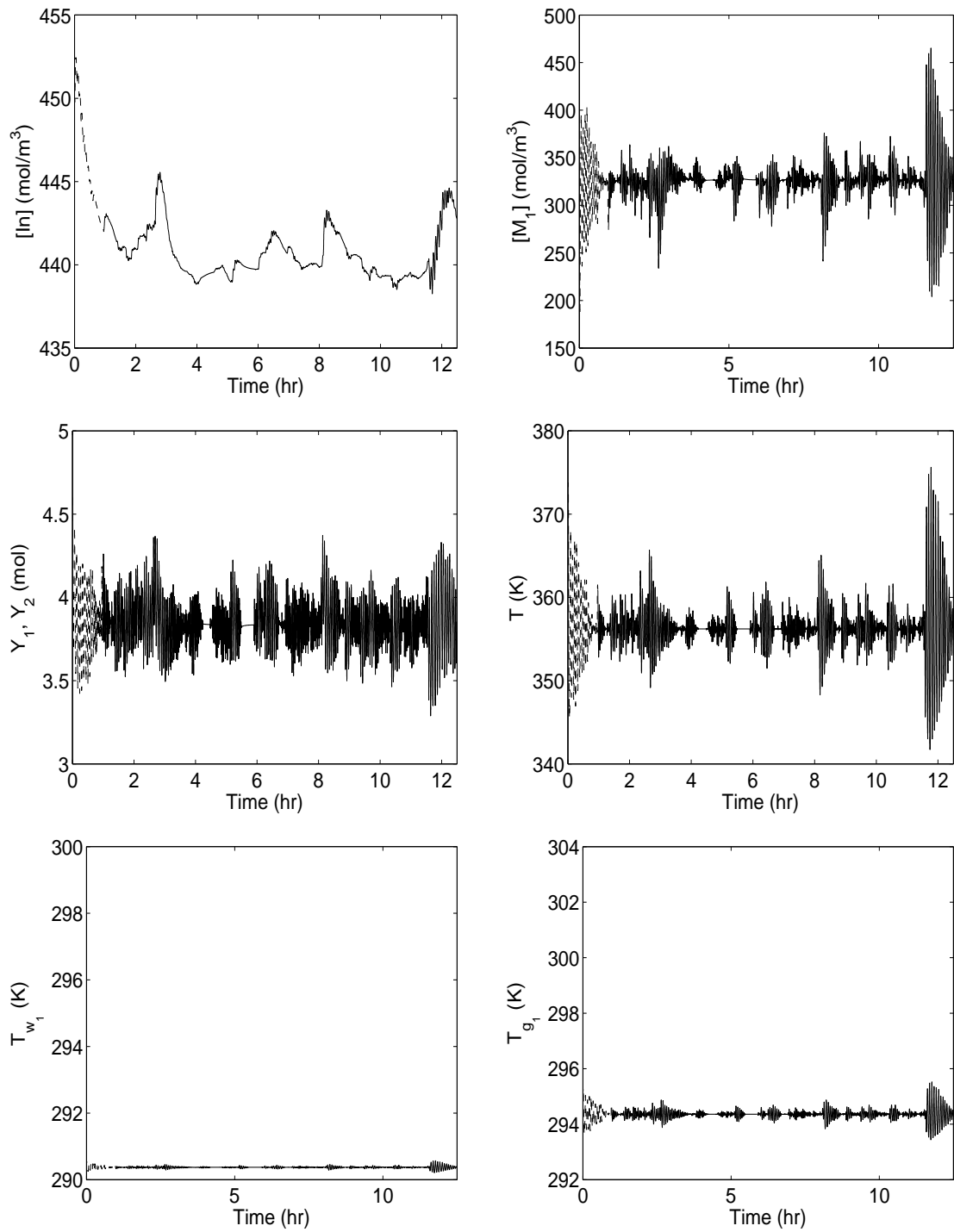


Figure 6.10: Evolution of the closed-loop state profiles under the primary configuration with the data loss rate increasing from 0.75 to 0.80 at 0.97 hours.

(see Figure 6.10) even though the state at the time that the data loss rate increases is within the stability region of the primary configuration ( $V_{c_1}(x(t = 0.97hour)) = 1.6380 \leq c_1^{max}$ ). This happens because the rate of data loss is not within the tolerable bound for primary control configuration ( $r > r_1^* = 0.75$ ).

In this case, the supervisor had available a fall-back control configuration with the feed temperature as the manipulated input. At time 0.97 hour when sensor data loss rate increases from 0.75 to 0.80,  $V_{c_2} = 1.6382$  implying that the state of the closed-loop system resides in the stability region of the fall-back configuration (that is,  $V_{c_2}(x(t = 0.97hour)) \leq c_2^{max}$ ) as well as  $r \leq r_2^* = 0.95$ . If the reconfiguration rule of Equation 6.11 is implemented, and the supervisor activates the fall-back configuration, the state trajectory converges to the desired steady-state (see Figure 6.11). The corresponding manipulated input profiles are shown in Figure 6.12.

## 6.6 Conclusions

In this chapter, we considered the problem of designing a fault-tolerant controller for nonlinear process systems subject to constraints and sensor data losses. Having identified candidate control configurations for a given system, we first explicitly characterized the stability properties that is, the set of initial conditions starting from where closed-loop stabilization under continuous availability of measurements is guaranteed as well as derived a bound on the maximum allowable data loss rate which preserves closed-loop stability. This characterization was utilized in designing a reconfiguration logic that was shown to achieve practical stability in the presence of sensor data losses. The application of the proposed method was illustrated using a chemical process example and demonstrated on a polyethylene reactor.

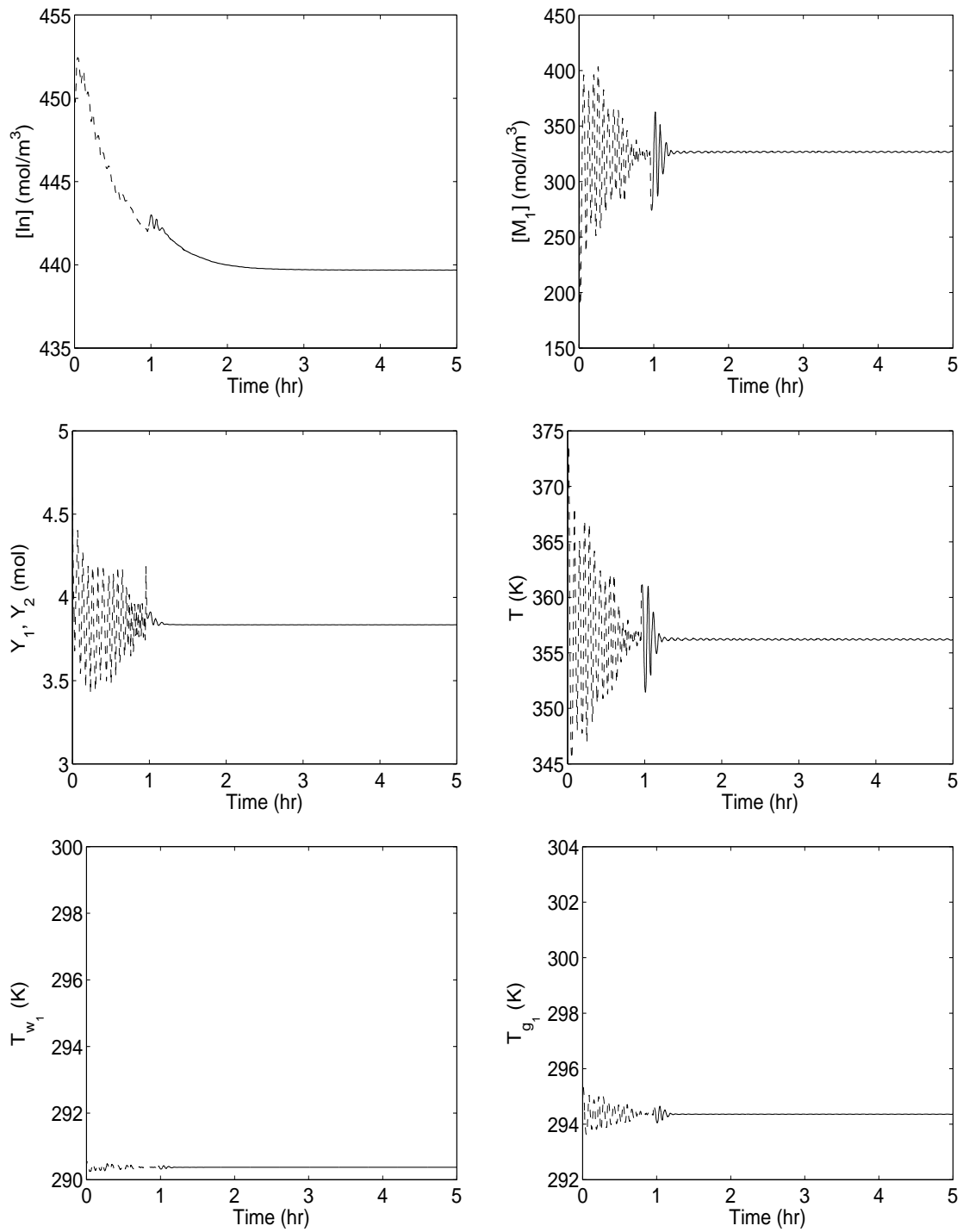


Figure 6.11: Evolution of the closed-loop state profiles under the reconfiguration law of Equation 6.11 with the data loss rate increasing from 0.75 to 0.80 at 0.97 hours.

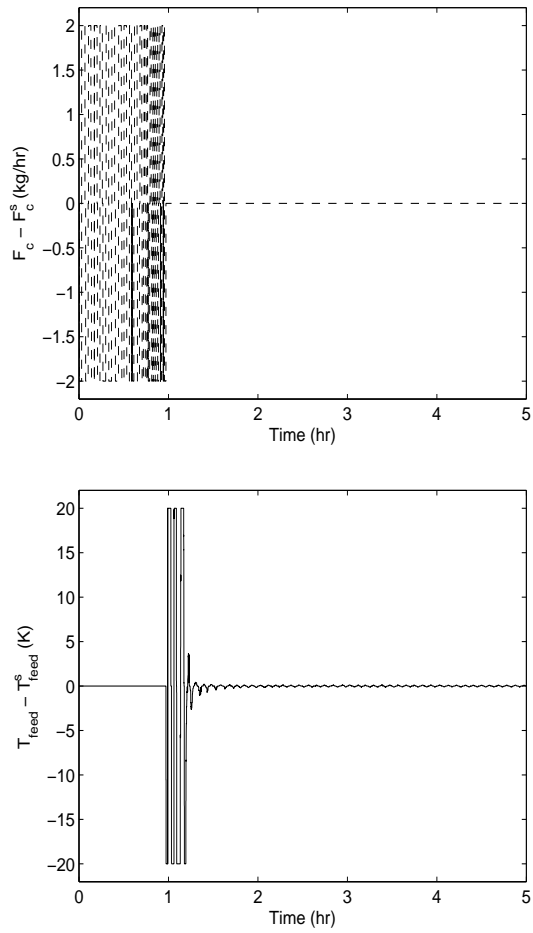


Figure 6.12: Evolution of the closed-loop input profiles under the reconfiguration law of Equation 6.11 with the data loss rate increasing from 0.75 to 0.80 at 0.97 hours.



## Chapter 7

# Handling Sensor Malfunctions in Control of Particulate Processes

### 7.1 Introduction

Particulate processes play a key role in a broad range of process industries ranging from chemical, materials and minerals to agricultural, food and pharmaceutical. These areas of manufacturing have a current value exceeding, according to some estimates, two trillion dollars and a growth factor of five to ten over the next decade. Examples include the crystallization of proteins for pharmaceutical applications, the emulsion polymerization for the production of latex, the fluidized bed production of solar-grade silicon particles through thermal decomposition of silane gas and the aerosol synthesis of titania powder used in the production of white pigments. Particulate processes are widely recognized as presenting a number of processing challenges which are not encountered in gas or liquid processes. One of these challenges is to operate the particulate process in a way that it consistently makes products with a desired particle size distribution. For example, in crystallization processes, the shape of the crystal size distribution is an important quality index which strongly affects

crystal function and downstream processing such as filtration, centrifugation, and milling [142].

Population balances have provided a natural framework for the mathematical modeling of particle size distributions (PSDs) (see, for example, the tutorial article [81] and the review article [140]), and have been successfully used to describe PSDs in many particulate processes. Population balance modeling of particulate processes typically leads to systems of nonlinear partial integro-differential equations that describe the rate of change of the PSD. The population balance models (PBMs) are also coupled with the material, momentum and energy balances that describe the rate of change of the state variables of the continuous phase, leading to complete particulate process models. In the context of PBM-based control of particulate processes, the main difficulty in synthesizing practically implementable nonlinear feedback controllers is the distributed parameter nature of the PBMs which does not allow their direct use for the synthesis of low-order (and therefore, practically implementable) nonlinear output feedback controllers. To overcome this problem, we took advantage of the property that the dominant dynamic behavior of many particulate process models is low-dimensional and proposed [24] a model reduction procedure, based on a combination of the method of weighted residuals and the concept of approximate inertial manifold, which leads to the construction of low-order ordinary differential equation (ODE) systems that accurately reproduce the dominant dynamics of broad classes of particulate process models. These ODE systems were subsequently used for the synthesis of nonlinear [24, 84, 28], robust [25, 45], and predictive [151, 152] controllers that enforce desired stability, performance, robustness and constraint handling properties in the closed-loop system. Owing to the low-dimensional structure of the controllers, the computation of the control action involves the solution of a small set

of ODEs, and thus, the developed controllers can be readily implemented in real-time with reasonable computing power. In addition to these results, an on-line optimal control methodology including various performance objectives was developed for a seeded batch cooling crystallizer in [175, 183]. The reader may refer to [33, 41, 20, 30] for reviews of results on simulation and control of particulate processes.

Despite this progress on the design of advanced feedback control systems for particulate processes, the problem of investigating controller stability, performance and robustness in the presence of sensor data losses has received no attention. Sensor data losses may arise due to a host of reasons including measurement sample loss, intermittent failures associated with measurement techniques, as well as those induced via data packet losses over transmission lines. Previous work on control subject to actuator/sensor faults has exclusively focused on lumped parameter systems. Specifically, in Chapter 2, communication losses were modeled as delays in implementing the control action and in Chapter 4 the problem of unavailability of some of the states for measurement was considered and reconfiguration-based strategies were devised to achieve fault-tolerance subject to faults in the control actuators. Furthermore, in Chapter 6, a theoretical framework was developed for the modeling, analysis and reconfiguration-based fault-tolerant control of nonlinear processes subject to asynchronous sensor data losses (intermittent unavailability of measurements). Specifically, for each control configuration, the stability region (i.e., the set of initial conditions starting from where closed-loop stabilization under continuous availability of measurements is guaranteed) as well as the maximum allowable data loss rate which preserves closed-loop stability was computed and this characterization was utilized in taking preventive action, i.e., to trigger reconfiguration, as well as in making the decision as to which backup configuration should be employed in the closed-loop sys-

tem to maintain stability. The method was applied to a lumped polyethylene reactor model.

This chapter focuses on the problem of feedback control of particulate processes in the presence of sensor data losses [65]. Two typical particulate process examples, a continuous crystallizer and a batch protein crystallizer, are considered and are modeled by population balance models. In the case of the continuous crystallizer, a Lyapunov-based nonlinear output feedback controller is first designed on the basis of an approximate moment model and is shown to stabilize an open-loop unstable steady state of the population balance model in the presence of input constraints. Then, the robustness of the nonlinear controller with respect to data losses is extensively investigated through simulations. In the case of the batch crystallizer, a predictive controller is first designed to obtain a crystal size distribution at the end of the batch that has desired shape while satisfying state and input constraints. Subsequently, we point out how the constraints in the predictive controller can be modified as a means of achieving constraint satisfaction in the closed-loop system in the presence of sensor data losses. Extensive simulations are presented to demonstrate the effect of sensor data losses on closed-loop stability and performance in both examples.

## **7.2 Handling Sensor Malfunctions: Continuous Crystallizer**

In the present section, we consider a standard model of a continuous crystallizer and address the problem of stabilization of its open-loop unstable steady-state using both state feedback and output feedback control in the presence of sensor data losses. We begin with the presentation of the crystallizer model, continue with the controller design and modeling of sensor data losses and conclude with extensive simulation results and discussion.

### 7.2.1 Population Balance Model of a Continuous Crystallizer

We consider a continuous crystallizer which is fed by a stream of solute at concentration  $c_0$ . 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 [98, 83]:

$$\begin{aligned}\frac{\partial n}{\partial \bar{t}} &= -\frac{\partial(R(\bar{t})n)}{\partial r} - \frac{n}{\tau} + \delta(r-0)Q(\bar{t}) \\ \frac{dc}{d\bar{t}} &= \frac{(c_0 - \rho)}{\bar{\epsilon}\tau} + \frac{(\rho - c)}{\tau} + \frac{(\rho - c)}{\bar{\epsilon}} \frac{d\bar{\epsilon}}{d\bar{t}}\end{aligned}\quad (7.1)$$

where  $n(r, \bar{t})$  is the density of crystals of radius  $r \in [0, \infty)$  at time  $\bar{t}$  in the suspension,  $\tau$  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:

$$\begin{aligned}R(\bar{t}) &= k_1(c - c_s) \\ Q(\bar{t}) &= \bar{\epsilon}k_2 \exp\left[-\frac{k_3}{\left(\frac{c}{c_s} - 1\right)^2}\right]\end{aligned}\quad (7.2)$$

where  $k_1$ ,  $k_2$ , and  $k_3$  are constants and  $c_s$  is the concentration of solute at saturation.

A second-order accurate finite-difference spatial discretization scheme with 1,000 discretization points was used to obtain the solution of the system of Equations 7.1-7.2 (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 Table 7.1. The crystallizer exhibits highly oscillatory behavior, which is the result of the

Table 7.1: Process parameters of the continuous crystallizer.

$c_0$	=	1000.0	$kg\ m^{-3}$
$c_s$	=	980.2	$kg\ m^{-3}$
$c_{0s}$	=	999.943	$kg\ m^{-3}$
$\rho$	=	1770.0	$kg\ m^{-3}$
$\tau$	=	1.0	$hr$
$k_1$	=	$5.065 \times 10^{-2}$	$mm\ m^3\ kg^{-1}\ hr^{-1}$
$k_2$	=	7.958	$mm^{-3}\ hr^{-1}$
$k_3$	=	$1.217 \times 10^{-3}$	

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 Equation 7.2). 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 Equations 7.1-7.2 to derive an approximate ODE model. Specifically, the  $j$ th moment of  $n(r, \bar{t})$  is defined as:

$$\mu_j = \int_0^\infty r^j n(r, \bar{t}) dr, \quad j = 0, \dots, \quad (7.3)$$

and upon multiplying the population balance in Equation 7.1 by  $r^j$ , integrating over all particle sizes, and introducing the following set of dimensionless variables and parameters:

$$\begin{aligned} \tilde{x}_0 &= 8\pi\sigma^3\mu_0, \quad \tilde{x}_1 = 8\pi\sigma^2\mu_1, \quad \tilde{x}_2 = 4\pi\sigma\mu_2, \quad \tilde{x}_3 = \frac{4}{3}\pi\mu_3, \dots, \\ t &= \frac{\bar{t}}{\tau}, \quad \sigma = k_1\tau(c_{0s} - c_s), \quad Da = 8\pi\sigma^3k_2\tau, \\ F &= \frac{k_3c_s^2}{(c_{0s} - c_s)^2}, \quad \alpha = \frac{(\rho - c_s)}{(c_{0s} - c_s)}, \quad \tilde{y} = \frac{(c - c_s)}{(c_{0s} - c_s)}, \quad u = \frac{(c_0 - c_{0s})}{(c_{0s} - c_s)} \end{aligned} \quad (7.4)$$

where  $c_{0s}$  is the steady-state solute concentration in the feed, the dominant dynamics of the process of Equation 7.1 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 in the following form:

$$\begin{aligned}
\frac{d\tilde{x}_0}{dt} &= -\tilde{x}_0 + (1 - \tilde{x}_3)Da e^{\frac{-F}{\tilde{y}^2}} \\
\frac{d\tilde{x}_1}{dt} &= -\tilde{x}_1 + \tilde{y}\tilde{x}_0 \\
\frac{d\tilde{x}_2}{dt} &= -\tilde{x}_2 + \tilde{y}\tilde{x}_1 \\
\frac{d\tilde{x}_3}{dt} &= -\tilde{x}_3 + \tilde{y}\tilde{x}_2 \\
\frac{d\tilde{y}}{dt} &= \frac{1 - \tilde{y} - (\alpha - \tilde{y})\tilde{y}\tilde{x}_2}{1 - \tilde{x}_3} + \frac{u}{1 - \tilde{x}_3}
\end{aligned} \tag{7.5}$$

where  $\tilde{x}_\nu$ ,  $\nu = 0, 1, 2, 3$ , are dimensionless moments of the crystal size distribution,  $\tilde{y}$  is dimensionless concentration of the solute in the crystallizer, and  $u$  is a dimensionless concentration of the solute in the feed. Note 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  $\tilde{x}_3$  and  $\tilde{y}$  are bounded, and it converges to a globally exponentially stable equilibrium point when  $\lim_{t \rightarrow \infty} \tilde{x}_3 = c_1$  and  $\lim_{t \rightarrow \infty} \tilde{y} = c_2$ , where  $c_1, c_2$  are constants. The reader may refer to [45] for a detailed derivation of the moments model, and to [28] for further results and references in this area. The stability properties of the fifth-order model of Equation 7.5 have been also studied and it has been shown [83] 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 Equation 7.1 around the unstable steady-state includes two isolated complex conjugate eigenvalues with a positive real part.

### 7.2.2 Bounded Lyapunov-Based Control

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 of Equation 7.5. The control objective

is to stabilize the crystallizer at an unstable steady-state (which corresponds to a desired PSD) using constrained control action. To this end, we initially re-write the moments model of Equation 7.5 in a more compact form:

$$\begin{aligned}\dot{\tilde{x}}(t) &= f(\tilde{x}(t)) + g(\tilde{x}(t))\tilde{u}(t) \\ |u| &\leq u_{max} \\ \tilde{z}(t) &= h(\tilde{x}(t))\end{aligned}\tag{7.6}$$

where  $\tilde{x} = [\tilde{x}_0 \ \tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3 \ \tilde{y}]'$ ,  $\tilde{x}_\nu = x_\nu - x_\nu^s$ ,  $\nu = 0, 1, 2, 3$ ,  $\tilde{z} = z - z^s$ ,  $\tilde{u} = u - u^s$ ,  $u_{max} > 0$  denotes the bound on the manipulated input, the superscript at  $x_\nu^s$  refers to the unstable steady-state at which we would like to asymptotically stabilize the system,  $h(\tilde{x}(t)) = \tilde{x}_0$  and  $z$  denotes the measured output. In the system of Equation 7.6,  $f$  and  $g$  have the following form:

$$f(\tilde{x}) = \begin{bmatrix} -\tilde{x}_0 + (1 - \tilde{x}_3)Da e^{\frac{-F}{\tilde{y}^2}} \\ -\tilde{x}_1 + \tilde{y}\tilde{x}_0 \\ -\tilde{x}_2 + \tilde{y}\tilde{x}_1 \\ -\tilde{x}_3 + \tilde{y}\tilde{x}_2 \\ \frac{1 - \tilde{y} - (\alpha - \tilde{y})\tilde{y}\tilde{x}_2}{1 - \tilde{x}_3} \end{bmatrix}, \quad g(\tilde{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{1 - \tilde{x}_3} \end{bmatrix}$$

Next we will review the design procedure of the bounded controller through state feedback and output feedback approaches. In the state feedback problem, measurements of  $\tilde{x}_\nu(t)$  and  $\tilde{y}(t)$  are assumed to be available for all  $t$ . In the output feedback problem, with the measurements of only  $\tilde{z} = \tilde{x}_0$  available, the controller is constructed through a standard combination of a state feedback controller with a state observer. The state feedback controller is synthesized via Lyapunov techniques and the state observer is an extended Luenberger-type observer.

### State Feedback Control

Consider the system of Equation 7.6, for which a control Lyapunov function (CLF),  $V(\tilde{x})$ , is available. Using the control Lyapunov function, we construct, using the



results in [103] (see also [46, 48]), the following continuous bounded control law:

$$u(\tilde{x}) = -k(\tilde{x})L_gV(\tilde{x}) \quad (7.7)$$

where

$$k(\tilde{x}) = \begin{cases} \frac{L_fV(\tilde{x}) + \sqrt{(L_fV(\tilde{x}))^2 + (u_{max}L_gV(\tilde{x}))^4}}{(L_gV(\tilde{x}))^2 \left[ 1 + \sqrt{1 + (u_{max}L_gV(\tilde{x}))^2} \right]}, & L_gV(\tilde{x}) \neq 0 \\ 0, & L_gV(\tilde{x}) = 0 \end{cases} \quad (7.8)$$

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.,

$$\Omega = \{\tilde{x} \in \mathbb{R}^n : V(\tilde{x}) \leq c^{max}\} \quad (7.9)$$

where  $c^{max} > 0$  is the largest number for which every nonzero element of  $\Omega$  is fully contained in the set:

$$\Phi = \{\tilde{x} \in \mathbb{R}^n : L_fV(\tilde{x}) < u_{max}|L_gV(\tilde{x})|\}. \quad (7.10)$$

### Output Feedback Control

Under the hypothesis that the system of Equation 7.6 is locally observable (that is, its linearization around the desired operating steady-state is observable), the practical implementation of a nonlinear state feedback controller of the form of Equation 7.7 will be achieved by employing the following nonlinear state observer:

$$\frac{d\omega}{dt} = f(\omega) + g(\omega)u + L(\tilde{z} - h(\omega)) \quad (7.11)$$

where  $\omega$  denotes the observer state vector (the dimension of the vector  $\omega$  is equal to the dimension of  $\tilde{x}$  in the system of Equation 7.6),  $\tilde{z}$  is the measured output, and  $L$  is a matrix chosen so that the eigenvalues of the matrix  $C_L = \frac{\partial f}{\partial \omega}|_{(\omega=\omega_s)} - L\frac{\partial h}{\partial \omega}|_{(\omega=\omega_s)}$ ,

where  $\omega_s$  is the operating steady-state, lie in the open left-half of the complex plane. The state observer of Equation 7.11 consists of a replica of the system of Equation 7.6 plus a linear gain multiplying the discrepancy between the actual and the estimated value of the output, and therefore, it is an extended Luenberger-type observer. The combination of the state observer of Equation 7.11 with the state feedback controller of Equation 7.7 leads to the following nonlinear output feedback controller:

$$\begin{aligned}
\frac{d\omega_0}{dt} &= -\omega_0 + (1 - \omega_3)Dae \frac{-F}{\omega_4^2} + L_0(h(\tilde{x}) - h(\omega)) \\
\frac{d\omega_1}{dt} &= -\omega_1 + \omega_4\omega_0 + L_1(h(\tilde{x}) - h(\omega)) \\
\frac{d\omega_2}{dt} &= -\omega_2 + \omega_4\omega_1 + L_2(h(\tilde{x}) - h(\omega)) \\
\frac{d\omega_3}{dt} &= -\omega_3 + \omega_4\omega_2 + L_3(h(\tilde{x}) - h(\omega)) \\
\frac{d\omega_4}{dt} &= \frac{1 - \omega_4 - (\alpha - \omega_4)\omega_4\omega_2}{1 - \omega_3} + L_4(h(\tilde{x}) - h(\omega)) \\
u &= -k(\omega)L_gV(\omega)
\end{aligned} \tag{7.12}$$

where  $L = [L_0 \ L_1 \ L_2 \ L_3 \ L_4]^T$  are the observer parameters and  $h(\omega) = \omega_0$ . The practical implementation of the nonlinear controller of Equation 7.12 requires online measurements of the controlled output  $\tilde{x}_0$ ; in practice, such measurements can be obtained by using, for example, light scattering [19, 142] and FBRM (focused-beam reflectance measurement) [11].

### 7.2.3 Modeling Sensor Data Loss

Following the approach presented in Chapter 6, sensor data losses are modeled within the framework of random Poisson processes. Specifically, at a given time  $t$  an ‘event’ takes place that determines whether the system will be closed-loop or open-loop (see Figure 7.1). For a given rate of data loss  $0 \leq r \leq 1$ , a random variable  $P$  is chosen from a uniform probability distribution between 0 and 1. If  $P \leq r$ , the event is

deemed to be ‘measurement loss’ (which implies that the process operates in open-loop), while if  $P > r$ , the event is understood to be ‘measurement available’ (which implies that the process operates in closed-loop). Furthermore, with  $W$  defined as the number of events per unit time, another random variable  $\chi$  with uniform probability distribution between 0 and 1 determines the time for which the current event will last, given by  $\Delta = \frac{-\ln\chi}{W}$ . At  $t + \Delta$  another event takes place and whether it represents a measurement or loss of measurement, as well as its duration, is similarly determined.

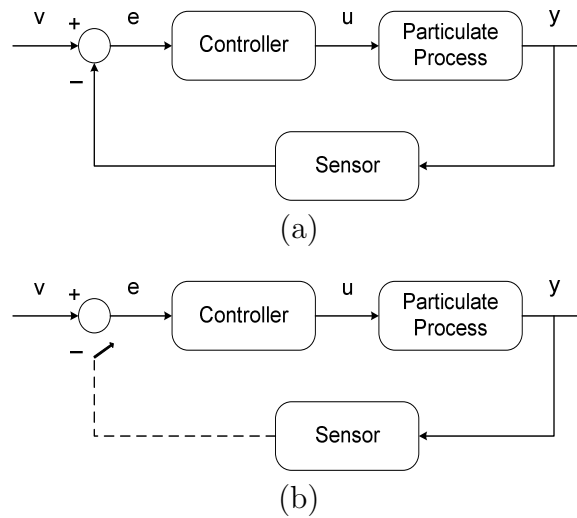


Figure 7.1: Closed-loop system in the (a) absence, and (b) presence of sensor data losses.

Note that in the presence of constraints in the manipulated input, prolonged duration of measurement loss may land the system states at a point starting from where stabilization may not be achievable (even with continuous measurement). Therefore, the presence of manipulated input constraints implies that the sensor data loss rate should be defined over a finite time interval. Specifically, for a positive real number  $T^*$ , we define  $r \in [0, 1]$  as sensor data loss rates over the finite time interval of duration  $T^*$ . This implies that over every successive finite time interval  $T^*$ , the mea-

measurements are available for a total time of  $T^* \times (1 - r)$ . Note that this definition does not impose any restrictions on the distribution of sequences of measurement loss and availability over the time interval  $T^*$  and does not need to hold for *any* finite interval  $T^*$  but only successive time intervals  $T^*$  (requiring the data loss rate to hold over *any* fixed finite time interval  $T^*$  would be equivalent to require it hold over infinitesimal time intervals). All it says is that over the time interval  $T^*$ , if the duration of all the measurement loss events is summed up, then that sum is equal to  $T^* \times r$ . In simulating data losses, this definition can be practically realized by picking  $W$  to be sufficiently large; the reasoning behind this is as follows: a larger value of  $W$  increases the number of events per unit time, and when  $W$  is sufficiently large, we can get a sufficiently large number of events over every finite time interval  $T^*$  such that the rate of data loss is sufficiently close to  $r$ .

#### 7.2.4 Simulation Results

In this section, we apply the state feedback controller of Equation 7.7 and output feedback controller of Equation 7.12 to the crystallizer process model and evaluate their robustness in the presence of sensor data losses. Specifically, the objective is to compute a data loss rate  $r^*$ , defined over a finite time interval  $T^*$ , such that if  $r < r^*$  then convergence to a desired neighborhood is achieved in the presence of data losses. Note that implicit in this analysis is the understanding that during the time that sensor measurements are unavailable, the values of the measured variables (in computing the control action) are ‘frozen’ at the last available measurement. This results in the value of the manipulated variable being frozen at the last computed value.

Note also, that the value of  $r^*$  is expected to depend on the interval  $T^*$  over which

it is defined (see the simulation example in [119] for a demonstration). To understand this more clearly, note that for convergence to a desired neighborhood of the origin, one can come up with a value  $\Delta^*$  such that if only one measurement was received every  $\Delta^*$ , then convergence to the desired neighborhood would be achieved. The robustness analysis in [119] exploits this fact together with the definition of the data loss rate, to ensure that over a  $\Delta^*$  duration within  $T^*$  (and across two time intervals), at least one measurement is received. In summary,  $\Delta^*$  is fixed by the given size of the neighborhood to the origin where convergence is desired ( $\delta'$ ); a given a  $T^*$  over which the data loss rate is defined,  $r^*$  can then in turn be picked such that the maximum duration of open-loop behavior across intervals stays less than  $\Delta^*$ .

Following the proposed methodology, we first use the reduced moments model of Equation 7.5 to design the controllers. 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 solute feed concentration. The values of the dimensionless model parameters in Equation 7.5 can be found in Table 7.2. The dimensionless solute feed concentration,  $u$ , is subject to the constraints:  $-u_{max} \leq u \leq u_{max}$ . For  $u_{max} = 2$ , the constraint on the inlet solute concentration corresponds to  $960 \text{ kg/m}^3 \leq c_0 \leq 1040 \text{ kg/m}^3$  and for  $u_{max} = 4$ , the constraint on the inlet solute concentration corresponds to  $920 \text{ kg/m}^3 \leq c_0 \leq 1080 \text{ kg/m}^3$ . The desired steady-state is  $\tilde{x}^s = [\tilde{x}_0^s \quad \tilde{x}_1^s \quad \tilde{x}_2^s \quad \tilde{x}_3^s \quad \tilde{y}^s]' = [0.0652 \quad 0.0399 \quad 0.0244 \quad 0.0149 \quad 0.6118]'$ , 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 Equation 7.5 in deviation variable form – thus translating the steady-state to the origin – to obtain the system of Equation 7.6 which we transform into the normal form. We introduce the invertible coordinate

Table 7.2: Dimensionless parameter values of the continuous crystallizer.

$\sigma$	$=$	$k_1\tau(c_{0s} - c_s)$	$=$	1.0	<i>mm</i>
$Da$	$=$	$8\pi\sigma^3k_2\tau$	$=$	200.0	
$F$	$=$	$k_3c_s^2/(c_{0s} - c_s)^2$	$=$	3.0	
$\alpha$	$=$	$(\rho - c_s)/(c_{0s} - c_s)$	$=$	40.0	

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:

$$\begin{aligned}\dot{\xi} &= A\xi + bl(\xi, \eta) + b\alpha(\xi, \eta)u \\ \dot{\eta} &= \Psi(\eta, \xi)\end{aligned}\tag{7.13}$$

where  $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ ,  $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ,  $l(\xi, \eta) = L_f^2 h(\Pi^{-1}(\xi, \eta))$  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 Equation 7.5 in the form of Equation 7.6, and are omitted for brevity.

The partially-linear  $\xi$ -subsystem in Equation 7.13 is used to design a bounded controller that stabilizes the full interconnected system of Equation 7.13 and, consequently, the original system of Equation 7.5. 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 Equations 7.7-7.8, where the positive-definite matrix,  $P = \begin{bmatrix} 1.7321 & 1.0000 \\ 1.0000 & 1.7321 \end{bmatrix}$ , is chosen to satisfy the Riccati matrix equality:  $A'P + PA - Pbb'P = -\bar{Q}$  where  $\bar{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  is a positive-definite matrix. The stability region estimate for the system is obtained as a level set of the Lyapunov function. For details on how to construct estimate of the stability regions for this system, see [151]. We initialize the crystallizer model at the following initial conditions:

$$[\tilde{x}_0(0) \ \tilde{x}_1(0) \ \tilde{x}_2(0) \ \tilde{x}_3(0) \ \tilde{y}(0)]' = [0.066 \ 0.041 \ 0.025 \ 0.015 \ 0.560]'$$

and initialize the observer at:

$$[\omega_0(0) \ \omega_1(0) \ \omega_2(0) \ \omega_3(0) \ \omega_4(0)]' = [0.047 \ 0.028 \ 0.017 \ 0.010 \ 0.5996]'$$

The matrix  $L$  was chosen as  $L = [L_0 \ L_1 \ L_2 \ L_3 \ L_4]' = [1 \ 0 \ 0 \ 0 \ 0]'$  to satisfy the requirement that the eigenvalues of the matrix  $C_L = \frac{\partial f}{\partial \omega} \Big|_{(\omega=\omega_s)} - L \frac{\partial h}{\partial \omega} \Big|_{(\omega=\omega_s)}$ , where  $\omega_s$  is the operating steady-state, lie in the open left-half of the complex plane.

We pick the value of number of events to be  $W = 400$  events per hour. Figure 7.2 shows the evolution of the closed-loop state and input profiles with  $u_{max} = 2$  and no sensor losses under both state feedback control (solid lines) and output feedback control (dashed lines). In both cases, we observe that the states of the closed-loop system converge to the desired steady-state. Figure 7.3 shows the evolution of the closed-loop state and input profiles with  $u_{max} = 2$  and 90% probability of sensor losses. Even though the amount of losses is very significant, both the state feedback controller (solid lines) and the output feedback controller (dashed lines) achieve stabilization of the process at the desired steady-state. However, if the sensor data loss rate is 95% closed-loop stability under both state feedback (solid lines) and output feedback control cannot be achieved (dashed lines), see Figure 7.4. We did not observe a significant difference between state feedback control and output feedback control in the sensor data loss rate for which closed-loop stability is preserved. This is expected due to the nature of the system dynamics. Specifically, we observed that even with continued open-loop operation, the process stays in a region such that if continuous measurements are received from a certain point in time onwards, closed-loop stability is achieved. The output feedback problem, up until the time that the state has not converged to the true values can be thought of as open-loop operation, however once

the state estimates converge then the problem ‘reverts’ to the state feedback problem and the preservation of closed-loop stability depends only on the data loss rate.

We also investigated the effect of different magnitude of manipulated input constraints on the sensor data loss rate that ensures closed-loop stability. Figure 7.5 shows the evolution of the state and input profiles with  $u_{max} = 4$  and no sensor data losses. The states of the closed-loop system under both state feedback control (solid lines) and output feedback control (dashed lines) converge to the steady-state. Figure 7.6 shows the evolution of the state and input profiles with  $u_{max} = 4$  and 70% probability of sensor data losses. In this case, closed-loop stability is maintained. However, when the data loss rate increases to 75% closed-loop stability is not achieved under both state feedback control (solid lines) and output feedback control (dashed lines), see Figure 7.7.

The reduced data loss rate under larger input constraints is expected because larger input constraints means that the input has a stronger effect on the process which implies that large time intervals of open-loop behavior of the manipulated input (when data losses occur) have an increased destabilizing effect on the closed-loop system. Table 7.3 summarizes  $r^*$  values for different  $u_{max}$ ; the larger the manipulated input constraint, the more sensitive the system toward sensor data loss. This observation also suggests that if excessive data loss rate occurs, the value of  $u_{max}$  can be artificially reduced to accommodate the data loss and if the current state resides in the stability region with the reduced  $u_{max}$ , closed-loop stability can be preserved.

### 7.3 Handling Sensor Malfunctions: Batch Crystallizer

In this section, we consider a batch particulate process and address the problem of producing PSD at the end of the batch that has a desired characteristics while



Table 7.3: Summary of  $r^*$  values for different  $u_{max}$  for the continuous crystallizer example.

$u_{max}$	$r^*$
1	0.05 (95% sensor data loss)
2	0.10 (90% sensor data loss)
3	0.20 (80% sensor data loss)
4	0.30 (70% sensor data loss)
6	0.35 (65% sensor data loss)

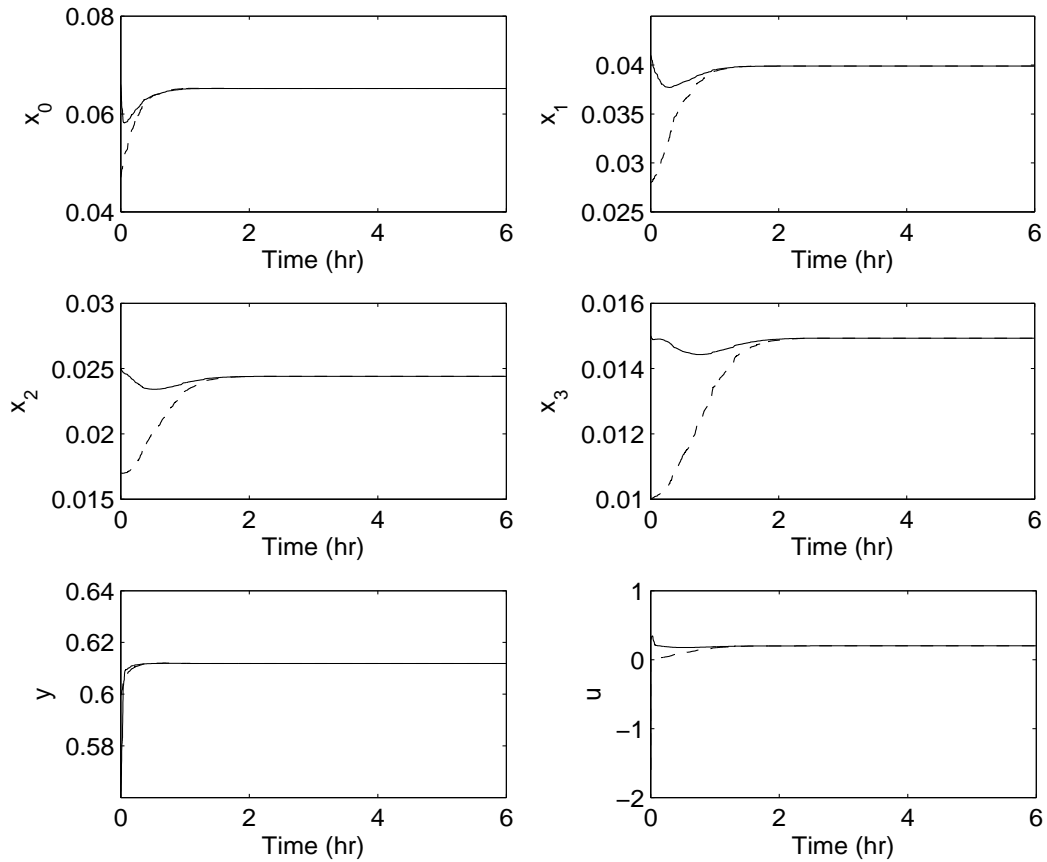


Figure 7.2: Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for  $u_{max} = 2$  and no sensor data losses.

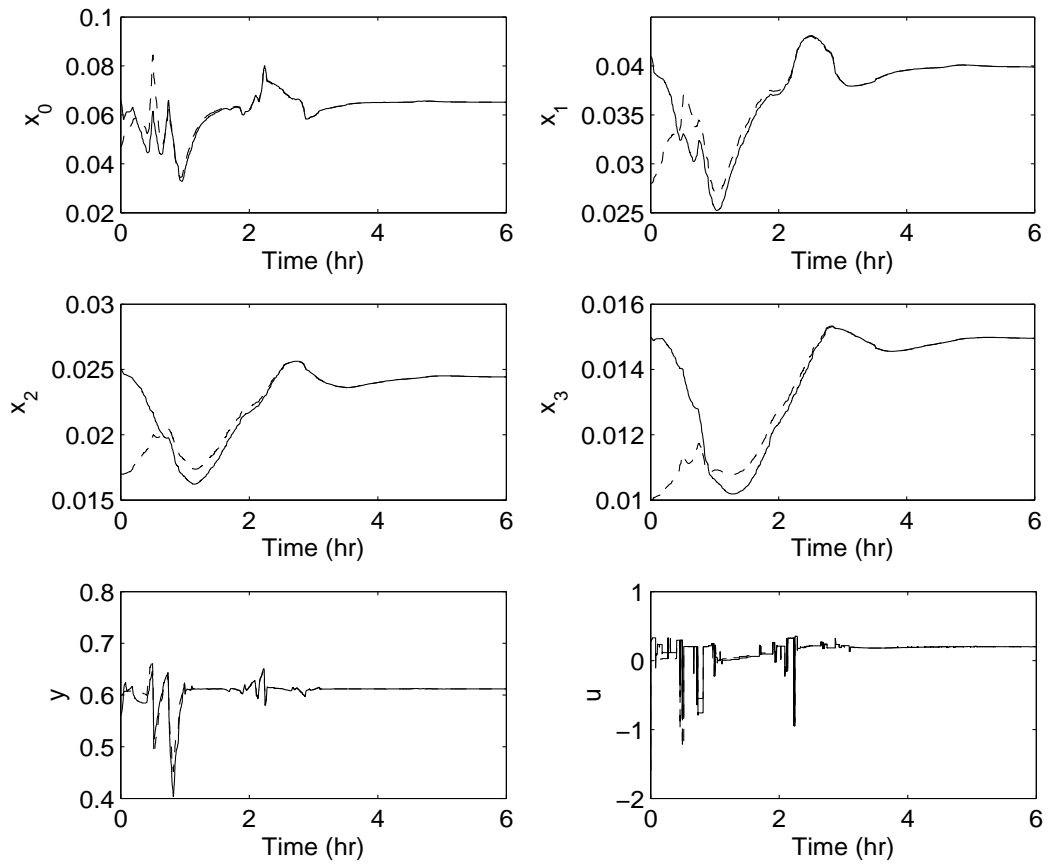


Figure 7.3: Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for  $u_{max} = 2$  and 90% probability of sensor data losses.

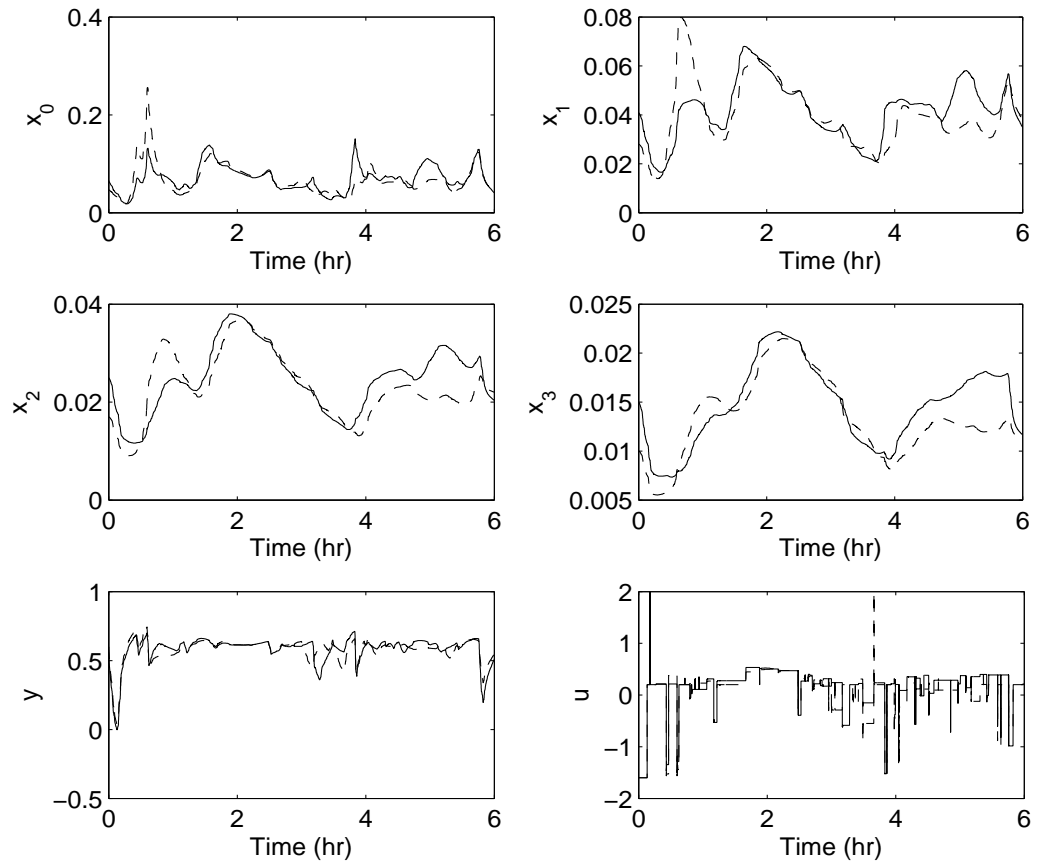


Figure 7.4: Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for  $u_{max} = 2$  and 95% probability of sensor data losses.

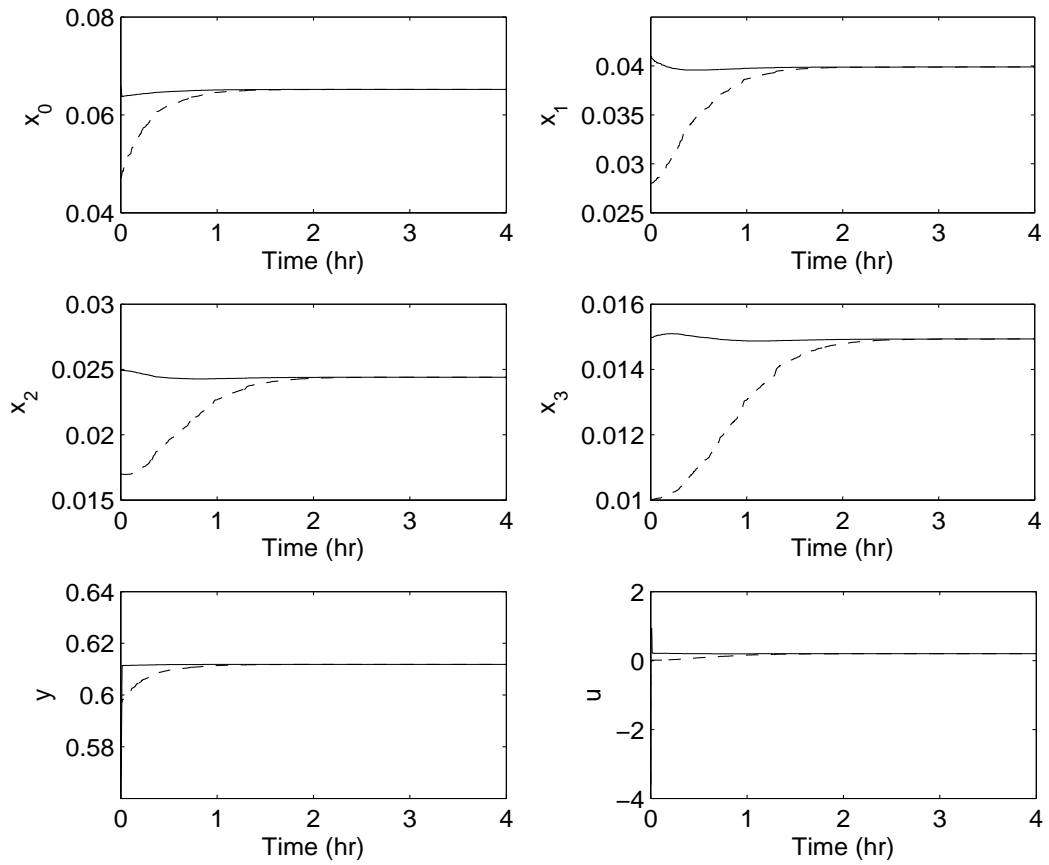


Figure 7.5: Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for  $u_{max} = 4$  and no sensor data losses.

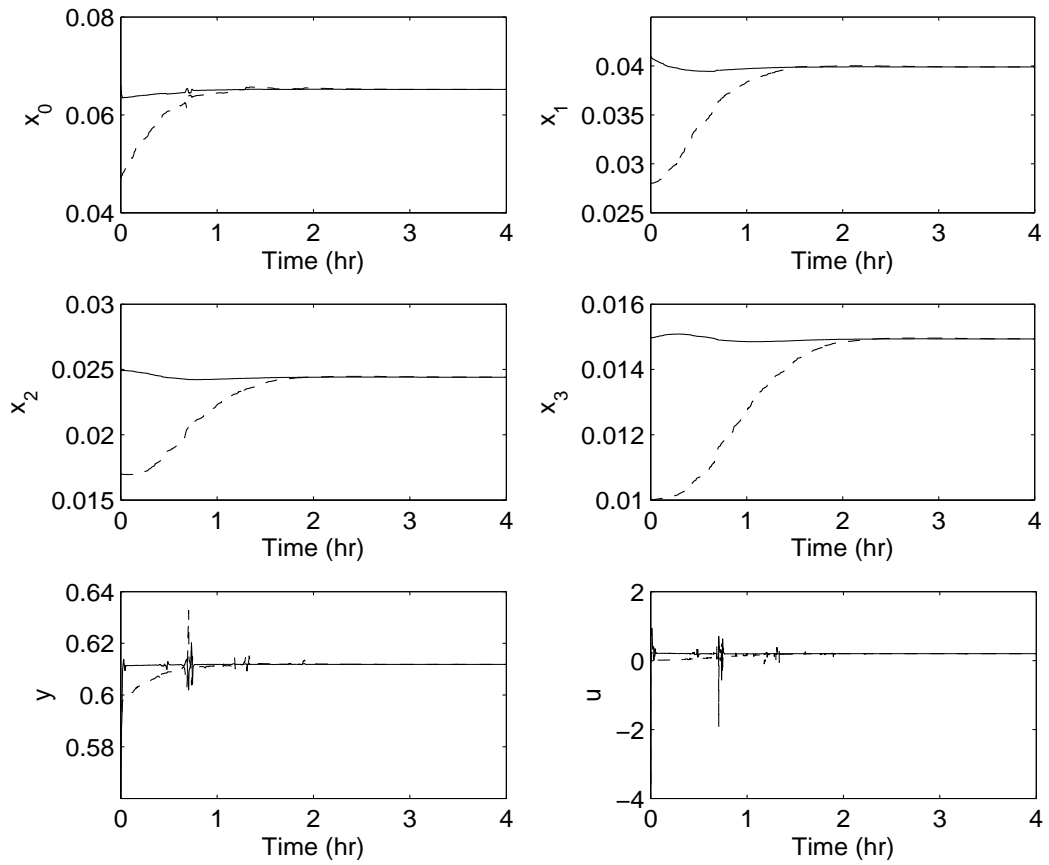


Figure 7.6: Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for  $u_{max} = 4$  and 70% probability of sensor data losses.

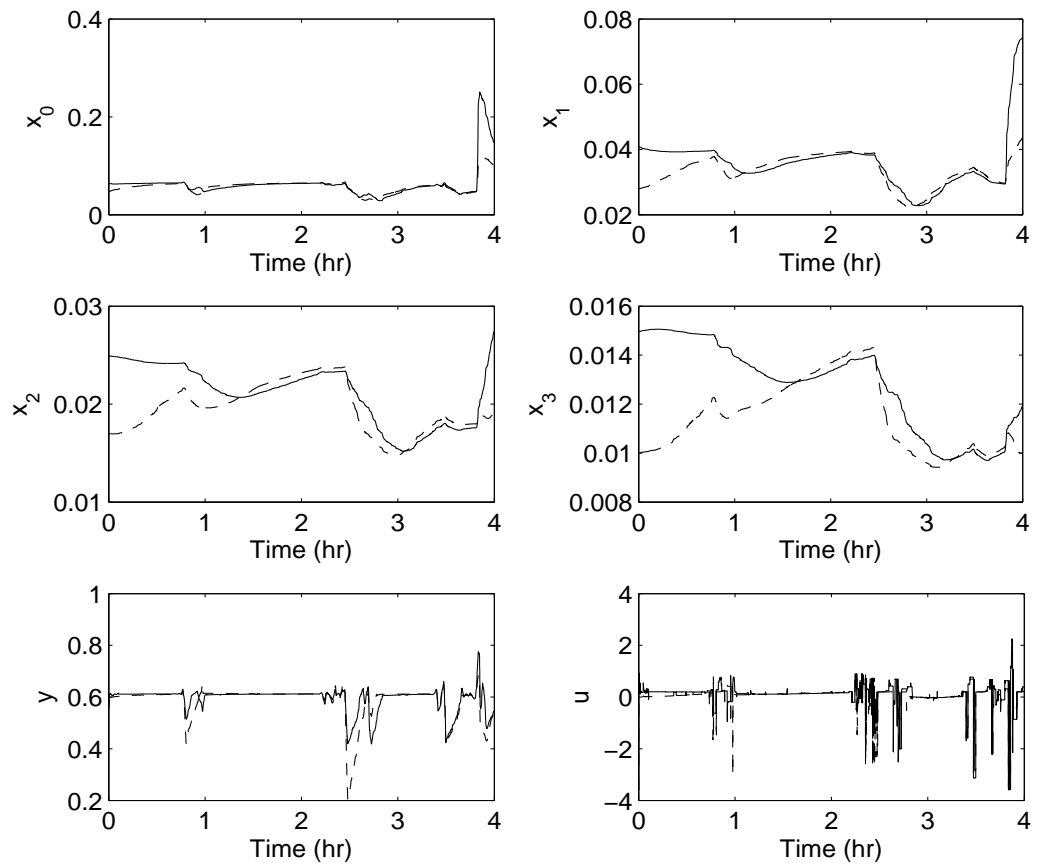


Figure 7.7: Evolution of the closed-loop state and input profiles under state feedback control (solid lines) and output feedback control (dashed lines) for  $u_{max} = 4$  and 75% probability of sensor data losses.

satisfying state and control constraints during the batch and handling sensor data losses.

### 7.3.1 Population Balance Model of a Protein Batch Crystallizer

The batch crystallizer considered in this chapter is taken from [152]. A population balance model is 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:

$$\begin{aligned} \frac{\partial n(r, t)}{\partial t} + G(t) \frac{\partial n(r, t)}{\partial r} &= 0, \quad n(0, t) = \frac{B(t)}{G(t)} \\ \frac{dC}{dt} &= -24\rho k_v G(t) \mu_2(t) \\ \frac{dT}{dt} &= -\frac{UA}{MC_p} (T - T_j) \end{aligned} \quad (7.14)$$

where  $G(t)$  is the growth rate,  $B(t)$  is the nucleation rate,  $\rho$  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 and  $\mu_2 = \int_0^\infty r^2 n(r, t) dr$  is the second moment of the CSD. The crystal nucleation rate  $B(t)$  [63, 16] is given by an equation of the following form:

$$B(t) = k_a C \exp\left(-\frac{k_b}{\sigma^2}\right) \quad (7.15)$$

where  $k_a$  and  $k_b$  are parameters that are obtained using experimental results. The supersaturation,  $\sigma$ , is the concentration of the solution in excess of the saturation concentration (solubility) and is understood to be the driving force for the crystal nucleation and growth. The supersaturation is defined as:

$$\sigma = \ln(C/C_s) \quad (7.16)$$

where  $C$  is the solute concentration and  $C_s$  is the solubility and expressed as follows:

$$C_s(T) = 1.0036 \times 10^{-3}T^3 + 1.4059 \times 10^{-2}T^2 - 0.12835T + 3.4613 \quad (7.17)$$

as a result of third-order polynomial data fitting based on solubility data in [148]. Equation 7.17 exhibits trends similar to the experimental solubility data of being low at low temperature and increasing significantly with increasing temperature. The crystal growth rate  $G(t)$  is derived based on an empirical model to describe the growth rate of the tetragonal HEW lysozyme crystals as a function of supersaturation in the following form:

$$G(t) = k_g \sigma^g \quad (7.18)$$

where  $k_g$  is the pre-exponential factor of the growth rate. Parameter values for this model are given in Table 7.4. Note that because of the tetragonal form of the crystals and the existence of about 46% of solvent in each crystal [99], the volumetric shape factor,  $k_v$ , is set equal to 0.54.

Table 7.4: Parameter values for the batch crystallizer model of Equations 7.14-7.18.

$k_a$	1044.4/(min cm <sup>3</sup> )	$k_g$	$3.1451 \times 10^{-9}$ cm/min
$k_b$	51.33	$g$	5.169
$k_v$	0.54	$\rho$	$1.40 \times 10^3$ mg/cm <sup>3</sup>
$U$	1800 kJ/m <sup>2</sup> · hr · K	$A$	0.25 m <sup>2</sup>
$M$	10 kg	$C_p$	4.13 kJ/K · kg

The fact that the dominant dynamics of the crystallizer are characterized by a small number of degrees of freedom [24], method of moments [81] (see also [28, 151, 107]) is applied to the system of Equation 7.14 to derive an approximate ODE model. Defining the  $i$ th moment of  $n(r, t)$  as:

$$\mu_i = \int_0^\infty r^i n(r, t) dr, \quad i = 0, 1, \dots, \infty \quad (7.19)$$



multiplying the population balance in Equation 7.14 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:

$$\begin{aligned}
\frac{d\mu_0}{dt} &= B(t) \\
\frac{d\mu_i}{dt} &= iG(t)\mu_{i-1}(t), \quad i = 1, 2, \dots, \infty \\
\frac{dC}{dt} &= -24\rho k_v G(t)\mu_2(t) \\
\frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j)
\end{aligned} \tag{7.20}$$

Note that in Equation 7.20, 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.

As will be seen in section 7.3.3, the control objective will require computation of  $\mu_3$  and  $\mu_4$ , hence  $N$  is chosen as 4 and the following reduced-order model is used for the purpose of controller design:

$$\begin{aligned}
\frac{d\mu_0}{dt} &= B(t) \\
\frac{d\mu_i}{dt} &= iG(t)\mu_{i-1}(t), \quad i = 1, 2, 3, 4 \\
\frac{dC}{dt} &= -24\rho k_v G(t)\mu_2(t) \\
\frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j)
\end{aligned} \tag{7.21}$$

### 7.3.2 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. Similar to the continuous crystallizer example, an extended Luenberger-type observer is used to estimate the values of the moments of the CSD and takes the following form:

$$\begin{aligned}
 \frac{d\hat{\mu}_0}{dt} &= \hat{B}(t) + L_0(C_m - \hat{C}) \\
 \frac{d\hat{\mu}_i}{dt} &= i\hat{G}(t)\hat{\mu}_{i-1}(t) + L_i(C_m - \hat{C}), \quad i = 1, \dots, 4 \\
 \frac{d\hat{C}}{dt} &= -24\rho k_v \hat{G}(t)\hat{\mu}_2(t) + L_5(C_m - \hat{C})
 \end{aligned} \tag{7.22}$$

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 7.5.

Table 7.5: Parameter values for the Luenberger-type observer of Equation 7.22.

$L_0$	-0.4	$L_1$	0.05
$L_2$	0.001	$L_3$	$1.7 \times 10^{-5}$
$L_4$	$3 \times 10^{-7}$	$L_5$	-0.1

Note that since the states  $\mu_3$  and  $\mu_4$  do not effect the evolution of the concentration, these states are therefore not observable from the concentration measurements. 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 there are

no crystals initially inside the crystallizer. In the case of a perfect model, therefore, the state estimates are naturally initialized at the true values and would continue to track the true values. In the case of plant-model mismatch, the observer continues to generate satisfactory estimates of the observable states.

### **7.3.3 Predictive Controller Formulation and Closed-Loop Results**

In the case of continuous crystallizer operation, the overriding objective is often stabilization, and the presence of constraints on the manipulated input limits the set of initial conditions starting from where stabilization can be achieved. For batch processes, in contrast, the expression of performance considerations in the form of appropriate constraints or through the objective function, and the achievement of a desired particle size distribution, is an important issue. Based on these considerations, we present in the remainder of this section a predictive controller formulation where, at time  $t_i$ , the control trajectory is computed by solving an optimization problem of

the form:

$$\begin{aligned}
\min \quad & -\frac{\mu_4(t_f)}{\mu_3(t_f)} \\
s.t. \quad & \frac{d\mu_0}{dt} = k_a C \exp\left(-\frac{k_b}{\sigma^2}\right) \\
& \frac{d\mu_i}{dt} = ik_g \sigma^g \mu_{i-1}(t), \quad i = 1, \dots, 4 \\
& \frac{dC}{dt} = -24\rho k_v k_g \sigma^g \mu_2(t) \\
& \frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) \\
& \mu_i(t_i) = \hat{\mu}_i(t_i) \\
& C(t_i) = \hat{C}(t_i) \\
& t_i \leq t \leq t_f \\
& T_{min} \leq T \leq T_{max} \\
& T_{j \ min} \leq T_j \leq T_{j \ max} \\
& \sigma_{min} + \epsilon \leq \sigma \leq \sigma_{max} - \epsilon \\
& \left| \frac{dC_s}{dt} \right| \leq k_1
\end{aligned} \tag{7.23}$$

$$\frac{B(t)}{G(t)} \leq n_{fine}, \forall t \geq t_f/2 \tag{7.24}$$

where  $\mu_4/\mu_3$  is the volume-averaged crystal size,  $T_{min}$  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  $\sigma$  are  $\sigma_{min} = 1.72$  and  $\sigma_{max} = 2.89$ . The constant,  $k_1$  (chosen to be 0.065 mg/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 5/ $\mu\text{m.ml}$ . The parameter  $\epsilon$  is used to allow for tightening of the constraints in the controller to enable constraint satisfaction for the system in the presence of sensor data losses and plant model mismatch. In the context of batch

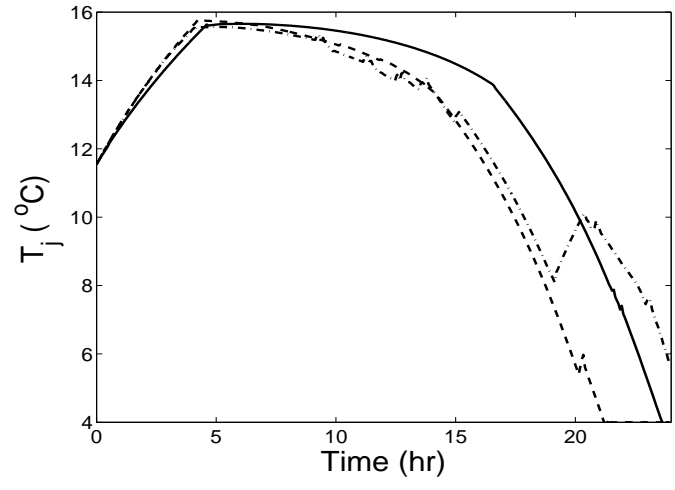
crystallizer control, previous work has shown that the objective of maximizing the volume-averaged crystal size can result in a large number of fines in the final product [104]. Therefore, the constraint of Equation 7.24 restricts the number of nuclei formed at any time instant during the second half of the batch run in order to limit the fines in the final product. Measurements of the solute concentration and the crystallizer temperature are assumed to be available;  $\Delta_m$ , the maximum possible delay between two successive measurements, is taken as five minutes. The measurements are used by the Luenberger-type observer to generate estimates of the moments, which are used as initial conditions of the states in the moments model.  $t_f$ , the total batch time, is chosen as 24 hours. The optimization problem is solved using sequential quadratic programming (SQP). A second-order accurate finite difference scheme with 3000 discretization points is used to obtain the solution of the population balance model of Equation 7.14.

We apply the control action computed by the low-order predictive controller of Equation 7.23 on the population balance model and study the problem of constraint satisfaction in the presence of sensor data losses and model uncertainty. Specifically, we consider a case of process-model mismatch by changing the value of the parameter  $g$  (the exponent relating growth rate to supersaturation) from its nominal value of 5.169 to 4.652 (a 10% change) in the predictive controller. We first show simulation results with maximum duration between successive measurement of  $\Delta_m = 5$  minutes and  $\epsilon$  in Equation 7.23 is 0. Solid lines in Figure 7.8 depict the implementation of the predictive controller where the state constraints are satisfied for the entire batch run and the performance objective is achieved (the supersaturation is within the lower and upper bound,  $1.72 \leq \sigma \leq 2.89$ ). Note that the possible errors in the values of the unobservable states has an impact on the achievement of the product properties

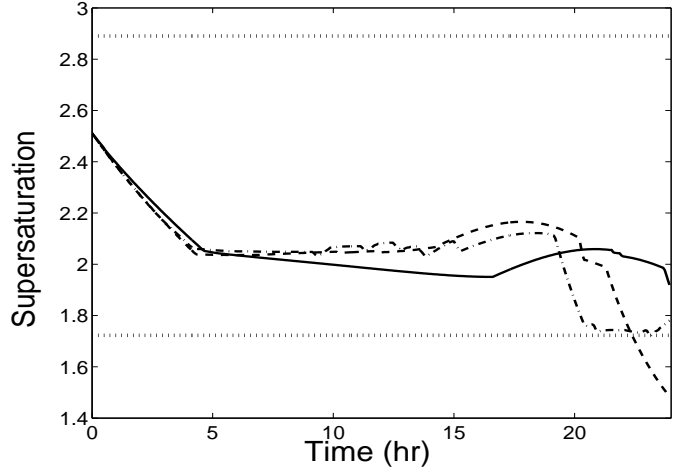
as described by the objective function. However, the satisfactory estimation of the observable states and the expression of the performance considerations as constraints on the observable states allows the achievement of most of the desired properties at the end of the batch run. Consider now the case where, due to sensor data losses, the maximum duration between successive measurement increases from  $\Delta_m = 5$  minutes to  $\Delta_m = 10$  minutes. Implementation of the predictive controller of Equation 7.23, with the same values of controller parameters as before ( $1.72 \leq \sigma \leq 2.89$ ), leads to violation of the state constraints (see dashed lines in Figure 7.8). To alleviate the problem of state constraint violation in the presence of sensor data losses, we implement the controller of Equation 7.23 with a tightened constraint on the supersaturation  $\epsilon = 0.24$  ( $1.96 \leq \sigma \leq 2.65$ ). As can be seen by the dotted lines in Figure 7.8, the predictive controller is able to successfully achieve the performance objective (the supersaturation is within the lower and upper bound,  $1.72 \leq \sigma \leq 2.89$ ), while at the same time respecting the state and input constraints in the presence of sensor data losses.

## 7.4 Conclusions

In this chapter, we investigated the problem of preserving closed-loop stability and performance of feedback control of particulate processes in the presence of sensor data losses. To demonstrate the issue of sensor data losses in the context of specific process applications, two typical particulate process examples, a continuous crystallizer and a batch protein crystallizer, were considered and modeled by population balance models. In both examples, feedback control systems was first designed on the basis of low-order models and applied to the population balance models to enforce closed-loop stability and constraint satisfaction. Subsequently, the robustness of the control



(a)



(b)

Figure 7.8: (a) Jacket temperature and (b) supersaturation profiles under output feedback control; sampling time of 5 minutes (solid lines), sampling time of 10 minutes without constraint modification (dashed lines) and sampling time of 10 minutes under the predictive controller with tightened constraints (dash-dotted lines).

systems in the presence of sensor data losses was investigated. Specifically, in the case of the continuous crystallizer, a Lyapunov-based nonlinear output feedback controller was designed and was shown to stabilize an open-loop unstable steady state of the population balance model in the presence of input constraints. It was demonstrated that this controller is robust with respect to significant sensor data losses but, as expected, it cannot maintain closed-loop stability when the sensor data losses exceed a certain threshold. In the case of the batch crystallizer, a predictive controller was first designed to obtain a desired crystal size distribution at the end of the batch while satisfying state and input constraints. In the presence of sensor data losses, we pointed out how the constraints in the predictive controller can be modified as a means of achieving constraint satisfaction in the closed-loop system in the presence of data losses.



## Chapter 8

# Analysis and Control of Mode Transitions in Biological Networks

### 8.1 Introduction

In a biological cell, cellular functions, such as metabolism, DNA synthesis, movement and information processing, are implemented and controlled by vast arrays of complex networks of biochemical interactions. Understanding how these networks are integrated and regulated, and how the regulation may be influenced – possibly for therapeutic purposes – is a major goal of molecular cell biologists and bioengineers. While experimental techniques have been, and will continue to be, an indispensable tool in the quest for such an understanding, it is now clear that the sheer complexity of biological networks is such that informal biochemical intuition alone cannot reliably deduce the underlying logic of these networks. This intuition must be supplemented by precise mathematical and computational tools that can provide both qualitative and quantitative insights into the description, analysis and manipulation of biological networks underlying basic cellular function. From a practical point of view, such techniques could potentially reduce the degree of trial-and-error exper-

imentation. More importantly, computational and theoretical approaches can lead to testable predictions regarding the current understanding of biological networks, which can serve as the basis for revising existing hypotheses. These realizations, together with recent technological advances that are increasingly enabling experimental validation of theoretical predictions, have been major driving forces behind a large and growing body of research work, in recent years, on the development and application of analytical and computational tools for the modeling and simulation (for example, [171, 110, 153, 36, 8, 70, 57]), optimization (for example, [34, 113]) and identification (for example, [58]) of biological networks. The reader may also refer to the review papers [77, 161, 8, 1] and the references therein for further results on biological networks.

Biological networks are intrinsically dynamical systems, driving the adaptive responses of a cell in space and time. The behavior of these dynamical systems is determined by “biochemical kinetics,” or rate equations, in which the variables of interest are the concentrations of individual network components (proteins, metabolites, etc.) within the cell, and the dynamics describe the rates of production and decay of these components. The dynamic models of biological networks typically consist of systems of nonlinear ordinary differential equations, permitting the modeler to apply the analytical techniques of nonlinear dynamics. These techniques have been developed considerably in recent decades, making the rate-equation approach a promising avenue for combining mathematical analysis and computational simulation.

While the resulting models are typically based on purely continuous dynamics, the dynamics of biological networks often involve switching between many qualitatively different modes of behavior. At the molecular level, for example, the fundamental process of inhibitor proteins turning off the transcription of genes by RNA polymerase

reflects a switch between two continuous processes. An example of this is the classic genetic switch observed in the bacteriophage  $\lambda$  (for example, see [138, 76, 147]), where two distinct behaviors, lysis and lysogeny, each with different mathematical models, are seen. Also, at the cellular level, the cell growth and division in a eukaryotic cell is usually described as a sequence of four processes, each being a continuous process that is triggered by a set of conditions or events (for example, see [78, 100, 162]). At the inter-cellular level, cell differentiation can also be viewed as a switched system [69]. In addition to naturally occurring switches, switched dynamics can be the result of external intervention that attempts to re-engineer a given network by turning on or off, for example, certain pathways. In all of these examples, the overall behavior of the network is more appropriately viewed as a switched system, i.e., intervals of continuous dynamics interspersed by discrete transitions, and, therefore, a hybrid approach that combines elements of discrete and continuous dynamics is necessary, not only for the modeling, simulation and analysis (for example, see [3, 2]), but also for controlling and modifying the network behavior.

Hybrid system models are increasingly being used for modeling a diverse array of engineering systems, such as automotive and chemical process control systems. A hybrid system consists of a finite family of continuous dynamical subsystems (or modes), each of which is governed by a different set of differential equations, together with a set of discrete events (or logic-based switching rules) that orchestrate the transition between the constituent modes. Research on hybrid systems, both within control systems theory and computer science, has led to the development of systematic tools for the modeling (for example, [177, 12]), simulation (for example, [12]), optimization (for example, [160, 165, 72]), stability analysis (for example, [80, 102, 39]), and control (for example, [21, 13, 94, 184, 49]) of several classes of hybrid systems. Given the

similarity that many biological networks exhibit to switched systems encountered in engineering (for example, involving feedback mechanisms and switching), it is instructive to investigate how all these tools can be applied to model, analyze and possibly modify the dynamics of biological networks.

Changes in network dynamics can result from alterations in local conditions (for example, temperature, nutrient and energy source, light, cell density) and/or changes in the molecular environment of individual regulatory components (for example, intracellular concentrations of transcription factors). Often, the network can be switched between different modes by changes in parameter values. These parameters typically include rate constants and total enzyme concentrations that are under genetic control. Changing the expression of certain genes will change the parameter values of the model and move the network across bifurcation boundaries into regions of qualitatively different behavior (for example, transitions from limit cycles to single and multiple steady-states). Understanding and analyzing the nature of these qualitatively different modes of behavior typically involves bifurcation analysis which determines how the attractors of the vector field depend on parameter values, leading to a characterization of the regions in parameter space where the different behaviors are observed. The boundaries of these regions represent the bifurcation boundaries.

An important question, however, that is not addressed by bifurcation analysis is that of when, or where in the state-space, is a transition from one mode to another feasible. For example, bifurcations can predict that a change in a certain parameter is required for the network to move from an oscillatory mode (stable limit cycle) to a multi-stable mode (multiple stable steady-states) but cannot tell us when, or which, of the new steady-states will be observed upon switching. This is an important consideration when one tries to manipulate the network behavior to achieve a certain

desirable behavior or steady-state. To address this question, bifurcations must be complemented by a dynamical analysis of the transient behavior of the constituent modes of the overall network. Intuitively, one expects that the newly switched mode will exhibit the desired steady-state if, at the time of switching, the network state is in the vicinity of that steady-state. A precise concept from nonlinear dynamical systems theory that quantifies this closeness is that of the domain of attraction, which is the set of all points in the state-space, starting from where the trajectories of the dynamical system converge to a given equilibrium state.

In this chapter, we present a methodology for the dynamic analysis of mode transitions in biological networks [51]. The proposed approach is based on the notion – introduced in [49, 47] – of coupling the switching logic to the domains of attraction of the constituent modes. To this end, we initially model the overall network as a switched nonlinear system that dwells in multiple modes, each governed by a set of continuous-time differential equations. The transition between the continuous modes are triggered by discrete events (changes in model parameters that correspond to alterations in physiological conditions). Then, following the characterization of the steady-state behavior of each mode, Lyapunov techniques are used to characterize the domains of attraction of the steady-states. Finally, by analyzing how the domains of attraction of the various modes overlap with one other, it is possible to determine when, and if, a given steady-state behavior, for a given mode transition, is feasible or not. The proposed method is demonstrated using models of biological networks that arise in cell cycle regulation and the bacteriophage  $\lambda$ -switch system.

## 8.2 A Switched System Representation of Biological Networks

We consider biological networks modeled by systems of nonlinear ordinary differential equations of the general form:

$$\begin{aligned}\frac{dx(t)}{dt} &= f_{i(t)}(x(t), p_{i(t)}) \\ i(t) &\in \mathcal{I} = \{1, \dots, N\}\end{aligned}\tag{8.1}$$

where  $x = [x_1 \ x_2 \ \dots \ x_n]^T \in \mathbb{R}^n$  is the vector of continuous state variables (for example, concentrations of the various network components such as proteins, genes, metabolites, etc.),  $f_i(\cdot)$  is a smooth nonlinear function,  $p_i$  is a vector of network parameters (for example, kinetic constants, total enzyme concentrations) that are typically under genetic control,  $i : [0, \infty) \rightarrow \mathcal{I}$  is the switching signal which is assumed to be a piecewise continuous (from the right) function of time, i.e.,  $i(t_k) = \lim_{t \rightarrow t_k^+} i(t)$  for all  $t_k \geq 0$ ,  $k \in Z_+$ , where  $Z_+$  is the set of positive integers and  $t_k$  is the  $k$ -th switching time, implying that only a finite number of switches occurs on any finite interval of time.  $N$  is the number of modes of the switched system,  $i(t)$ , which takes different values in the finite index set,  $\mathcal{I}$ , represents a discrete state that indexes the vector field  $f_i(\cdot)$  which determines  $\dot{x}$ . For each value that  $i$  takes in  $\mathcal{I}$ , the temporal evolution of the continuous state is governed by a different set of differential equations. The system of Equation 8.1 is therefore a switched (multi-modal) system that consists of a finite family of continuous nonlinear subsystems (modes) and a switching rule that orchestrates the transitions between them. In biological networks, mode transitions can be the result of a fundamental change in the vector field itself (for example, different modes having different  $f_i$ 's) or, more commonly, a change in network parameter values due to changes in levels of gene expression and enzyme activities (which can occur spontaneously or be induced externally).

The basic problem that we address in this chapter is that of determining when (or where in the state-space) can a transition from one mode to another produce a certain desired behavior that exists in the target mode (for example, a desired steady-state). From an analysis point of view, the answer to this question sheds light on why certain naturally-occurring mode transitions seem to always favor a certain steady-state behavior. From a control point of view, on the other hand, the answer provides insight into how and when the designer should enforce the transition in order to bring about a desired steady-state behavior. In the next section, we outline a methodology that addresses these questions.

### **8.3 Methodology for Analysis of Mode Transitions**

The methodology proposed here is based on the idea of designing the switching logic on the basis of the domains of attraction of the constituent modes, which was introduced in [49] in the context of constrained control of switched nonlinear systems. However, unlike the results in [49] where the restrictions on the size of the domains of attraction were a consequence of the constraints imposed on the manipulated input of each mode, the domains of attraction considered here are directly linked to the intrinsic dynamic behavior of the constituent modes, which is dictated by the dependence of the attractors of the vector field on the network parameters. For example, the presence of multiple equilibrium points in a given mode gives rise to multiple stability regions, or domains of attraction, whose union covers the entire state-space. Clearly, which equilibrium state is attained depends on which region contains the system state at the switching time. Below is the proposed methodology:

1. Identify the different modes of the network, where each mode is characterized either by a different set of differential equations or by the same set of equations

but with different parameters.

2. Compute the steady-state(s) of each mode by solving:

$$0 = f_i(x_s, p_i) \tag{8.2}$$

where  $x_s$  is an admissible steady-state solution. Depending on the values of  $p$ , each mode might possess a limit cycle, a single steady-state, or multiple steady-states.

3. Characterize the domain of attraction (stability region) of each steady-state in each mode. For a given steady-state,  $x_s$ , the domain of attraction,  $\Omega(x_s)$ , consists of the set of all states starting from where the system trajectories converge to that steady-state. Estimates of the domain of attraction can be obtained using Lyapunov techniques [91]. For example, consider the case of isolated equilibrium points and let  $V_i$  be a Lyapunov function candidate, i.e.,  $V_i(x_s) = 0$  and  $V_i(x) > 0$  for all  $x \neq x_s$ . Consider also the set  $\Pi(x_s) = \{x \in \mathbb{R}^n : \dot{V}_i(x) < 0\}$ . Then the level set,  $\Omega(x_s) = \{x \in \mathbb{R}^n : V_i(x) \leq c_i^{max}\}$ , where  $c_i^{max} > 0$  is the largest constant for which  $\Omega$  is fully contained in  $\Pi$ , provides an estimate of the domain of attraction of  $x_s$  (see [48, 46] for more details on this issue). Due to the possible conservatism of the resulting estimates, Lyapunov techniques are usually coupled with other methods in order to obtain larger estimates (for example, multiple Lyapunov functions; see chapter 4 in [91] for details).
4. Analyze how the domains of attraction of a given mode overlap with those of another mode. Suppose, for example, that the network is initialized within mode  $k$  and let  $T$  be the transition time from mode  $k$  to mode  $j$ . Also, let  $x_s$  be an admissible steady-state (among several others) of the  $j$ -th mode. Then, if

$$x(T) \in \Omega_j(x_s) \tag{8.3}$$



and  $i(t) = j \forall t \geq T^+$  (i.e., no further switches take place), then we will have  $\lim_{t \rightarrow \infty} x(t) = x_s$ , i.e., the  $x_s$  steady-state will be observed following switching. The switching rule of Equation 8.3 requires monitoring the temporal evolution of the state evolution in order to locate where the state is at the switching time, with respect to the domains of attraction of the mode to be activated.

**Remark 8.1** Referring to the computation of the steady-states of a biological network, we note that it is, in general, difficult to compute all the steady-state solutions of a system of nonlinear ordinary differential equations (ODEs). For an arbitrary system of nonlinear ODEs, where the right-hand side does not possess any kind of structure, one can resort to general search algorithms, such as Newton-type methods, to solve Equation 8.2. These methods are usually local in character and thus may require an extensive search over all possible initial guesses in order to find all possible solutions. For biological systems, the search complexity can be reduced somewhat by taking advantage of the natural limits on the values of the state variables in order to bracket the region in the state-space where the system is expected to operate and where the search needs to be carried out. More importantly, the dynamic models of biological systems often exhibit specific types of structure that arise from physical considerations and can thus be exploited in the computation of all the steady-states using computational algorithms that have been developed in the literature. For example, if each component on the right-hand side of the system of ODEs in Equation 8.1,  $f_i$ , involves linear combinations of rational functions of variables and parameters, then the algorithm developed in [188] can be used to find all the steady-states (the algorithm converts the steady-state equations into a system of polynomial equations and uses a globally convergent homotopy method to find all the roots of the system of polynomials). Most biological models of molecular networks have linear combinations of rational functions for the right-hand side of their system of ODEs (see the cell-cycle and  $\lambda$ -switch models studied in the next two sections for examples). In fact, the right-hand sides are usually even more restricted to mass action and Michaelis-

Menten type kinetics. Mass action kinetics have the form  $k * S_1 * S_2 * \dots * S_n$ ; where  $k$  is a rate constant (parameter) and  $S_i$  represents the concentration of a protein (variable). Michaelis- Menten kinetics have the form  $k * S * E / (K_m + S)$ ; where  $k$  is a rate constant (parameter),  $K_m$  is a Michaelis constant (parameter),  $S$  is the substrate concentration (variable), and  $E$  is the enzyme concentration (variable). Clearly, these kinetics are rational functions. Once the target steady-states are identified, the domains of attraction for each steady-state can be computed. Then, the switching rule of Equation 8.3 ensures a priori where the system will end up upon switching at a given point in the state-space, provided that this point is within the domain of attraction of a stable steady-state. Finally, it should be noted that even in the rare case that a structure cannot be identified – and subsequently not all of the steady-states can be found – the proposed method still provides useful information regarding the feasibility of switching into any of the known steady-states by verifying whether the state at any given given time is contained within its domain of attraction.

**Remark 8.2** The issue of robustness of the proposed approach with respect to model uncertainty can be explicitly handled by modifying the computation of the domains of attraction following the methodology proposed in [46] to account for the presence of parametric model uncertainty in the computation of the domain of attraction using bounds on the variation of the model parameters.

**Remark 8.3** The Lyapunov function-based approach that we follow for the construction of the domains of attraction for the individual stable steady-states yields a domain of attraction estimate that is dependent upon the specific Lyapunov function used. To improve upon the obtained estimate, one can use a group of different Lyapunov functions to come up with a larger estimate of the domain of attraction. Other methods for the construction of the Lyapunov function, such as Zubov’s method (for example, [42]) and the sum of squares decomposition approach [134], can also be used. Acceptability of the computed estimates should ultimately be judged with respect to the size of the expected operating regime. Once the domain of attraction estimates

are obtained, the switching rule of Equation 8.3 ensures that the system will go to a certain stable steady-state if the switching occurs at a point which is within the domain of attraction of this steady-state. Finally, we note that the case of multiple mode switchings can be handled in a sequential fashion – the same way that the first mode switch is handled – by tracking where the state is at the time of each switch.

**Remark 8.4** It should be noted that the proposed approach is not limited by the dimensionality of the system under consideration but applies to systems of any dimensionality. The estimation of the domain of attraction utilizes only simple algebraic computations and does not incur prohibitive computational costs with increasing dimensionality. In the simulation studies presented below, the domains of attraction are plotted for the sake of a visual demonstration. However, a plot of the domain of attraction is not *required* for the implementation of the switching rule, and, therefore, poses no limitation when considering systems of higher dimensions. The knowledge of the domain of attraction is contained completely in the value of the level set,  $c_i$ , obtained when computing the estimate of the domain of attraction. At the time of implementation, to ascertain whether the state is within the domain of attraction requires only evaluating the Lyapunov function and verifying if  $V_i(x(T)) \leq c_i$ . To reduce the possible conservatism of the resulting estimate, it is often desirable to find the largest value of  $c_i$  for which the estimate  $\Omega_{c_i} = \{x : V_i(x) \leq c_i\}$  is fully contained within  $\Pi_i$ . For this purpose, an iterative procedure to recompute (and enlarge) the estimate of the domain of attraction can be employed whereby the value of  $c_i$  is increased gradually in each iteration until a value,  $c_i^{max}$ , is reached where for any  $c_i > c_i^{max}$ ,  $\Omega_{c_i}$  is no longer fully contained in  $\Pi_i$ . The level set  $\Omega_{c_i^{max}}$  then is the largest estimate of the domain of attraction that can be obtained using the level sets of the given Lyapunov function. Note that, for a given value of  $c_i$  in each iteration, the determination of whether  $\Omega_{c_i}$  is fully contained in  $\Pi_i$  involves only inexpensive algebraic computations and thus this iterative procedure does not incur prohibitive computational costs as the dimensionality of the system increases. The same procedure also applies when a family of Lyapunov functions is used to estimate

the domain of attraction of a given steady-state. Finally, it should be noted that how close the obtained estimate is to the actual domain of attraction depends on the particular system structure as well as the method used to compute this estimate (in this case the particular Lyapunov functions chosen). In general, it is expected that the estimate will not capture the entire domain of attraction which implies that the union of all the estimates of the domains of attraction of all the steady-states will not cover the entire state-space. An implication of this, for the case when switching of the network is controlled externally and a priori stability guarantees are sought, is that switching should be delayed until the state trajectory enters the computed estimate of the domain of attraction of the desired target steady-state. The “gaps” in between the different estimates (and hence the conservatism of the switching policy) can be reduced either with the help of dynamic simulations or by augmenting the individual estimates using any of the methods cited in Remark 8.3.

**Remark 8.5** The proposed approach models biological networks using deterministic differential equations and does not account for possible network stochastic behavior. Such stochasticity can be modeled as uncertainty in the model parameters, and therefore be handled directly by modifying the computation of the domains of attraction in a way that accounts explicitly for the effect of parameter model uncertainty following the methodology proposed in [46].

In the next two sections, we demonstrate, through computer simulations, the application of this methodology to the analysis of mode transitions in two biological networks, one arising in eukaryotic cell cycle regulation and the other in the bacteriophage  $\lambda$ -switch system. We note here that the focus in both examples is not on the modeling aspect, but rather on illustrating how the proposed analysis method can be applied given some available model of the network (which could come either from first-principles or from data).

## 8.4 Application to Eukaryotic Cell Cycle Regulation

We consider here an example network of biochemical reactions, based on cyclin-dependent kinases and their associated proteins, which are involved in cell cycle control in frog egg development. A detailed description of this network is given in [133] where the authors use standard principles of biochemical kinetics and rate equations to construct a nonlinear dynamic model of the network that describes the time evolution of the key species including free cyclin, the M-phase promoting factor (MPF), and other regulatory enzymes. The model parameters have either been estimated from kinetic experiments in frog egg extracts or assigned values consistent with experimental observations. For illustration purposes, we will consider below the simplified network model derived by the authors (focusing only on the positive-feedback loops in the network) which captures the basic stages of frog egg development. The model is given by:

$$\begin{aligned}\frac{du}{dt} &= \frac{k'_1}{G} - (k'_2 + k''_2 u^2 + k_{wee})u + (k'_{25} + k''_{25} u^2) \left( \frac{v}{G} - u \right) \\ \frac{dv}{dt} &= k'_1 - (k'_2 + k''_2 u^2)v\end{aligned}\tag{8.4}$$

where  $G = 1 + \frac{k_{INH}}{k_{CAK}}$ ,  $k_{INH}$  is the rate constant for inhibition of INH, a protein that negatively regulates MPF,  $k_{CAK}$  is the rate constant for activation of CAK, a cdc2-activating kinase,  $u$  is a dimensionless concentration of active MPF and  $v$  is a dimensionless concentration of total cyclin,  $k'_2$  and  $k''_2$  are rate constants for the low-activity and high activity forms, respectively, of cyclin degradation,  $k'_{25}$  and  $k''_{25}$  are rate constants for the low-activity and high activity forms, respectively, of tyrosine dephosphorylation of MPF,  $k'_1$  is a rate constant for cyclin synthesis,  $k_{wee}$  is the rate constant for inhibition of *Wee1*, an enzyme responsible for the tyrosine phosphorylation of MPF (which inhibits MPF activity) (see [133] for model derivation from the molecular mechanism and Table 8.1 for the parameter values). Bifurcation

and phase-plane analysis of the above model [133] shows that, by changing the values of  $k'_2$ ,  $k''_2$  and  $k_{wee}$ , the following four modes of behavior are predicted (see Table 8.2):

- A G2-arrested state (blocked before the G2-M transition) characterized by high cyclin concentration and little MPF activity. This corresponds to a unique, asymptotically stable steady-state ( $k'_2 = 0.01$ ,  $k''_2 = 10$ ,  $k_{wee} = 3.5$ ; see Figure 8.1(a)).
- An M-arrested state (blocked before the meta- to anaphase transition) state with lots of active MPF. This corresponds to a unique, asymptotically stable steady-state ( $k'_2 = 0.01$ ,  $k''_2 = 0.5$ ,  $k_{wee} = 2.0$ ; see Figure 8.1(b)).
- An oscillatory state (alternating phases of DNA synthesis and mitosis) exhibiting sustained, periodic fluctuation of MPF activity and total cyclin protein. This corresponds to a stable limit cycle surrounding an unstable equilibrium point ( $k'_2 = 0.01$ ,  $k''_2 = 10$ ,  $k_{wee} = 2.0$ ; see Figure 8.1(c)).
- Co-existing stable steady-states of G2-arrest and M-arrest. This corresponds to three steady-states; one unstable and two locally asymptotically stable ( $k'_2 = 0.015$ ,  $k''_2 = 0.1$ ,  $k_{wee} = 3.5$ ; see Figure 8.1(d)).

Table 8.1: Parameter values for the cell cycle model in Equation 8.4 [133].

$k'_1$	=	0.01
$k'_{25}$	=	0.04
$k''_{25}$	=	100
$k_{INH}$	=	0.1
$k_{CAK}$	=	1

The above analysis predicts that slight increases in  $k'_2$  and  $k_{wee}$ , accompanied by a significant drop in  $k''_2$  (which could be driven, for example, by down-regulation of

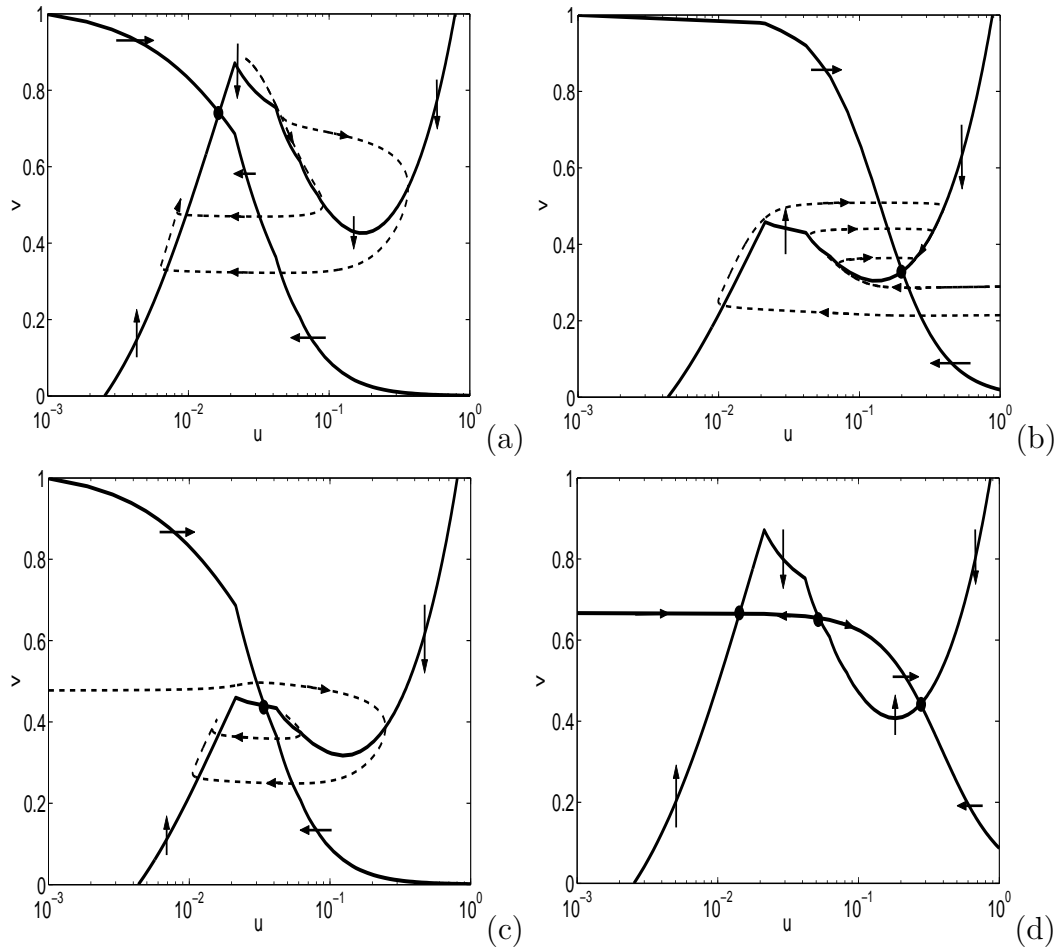


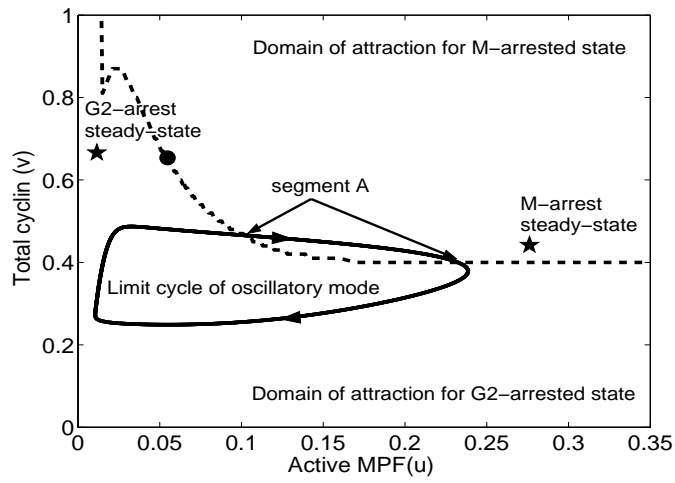
Figure 8.1: Phase-plane portraits of the system of Equation 8.4, for different values of  $k'_2$ ,  $k''_2$ , and  $k_{wee}$ , showing: (a) Stable steady-state with most MPF inactive, (b) Stable steady-state with most MPF active, (c) Unstable steady-state surrounded by a limit cycle, and (d) Bi-stability: two stable steady-states separated by an unstable saddle point.

Table 8.2: Steady-state values  $(u_s, v_s)$  for the cell cycle model for different values of  $k'_2$ ,  $k''_2$  and  $k_{wee}$ .

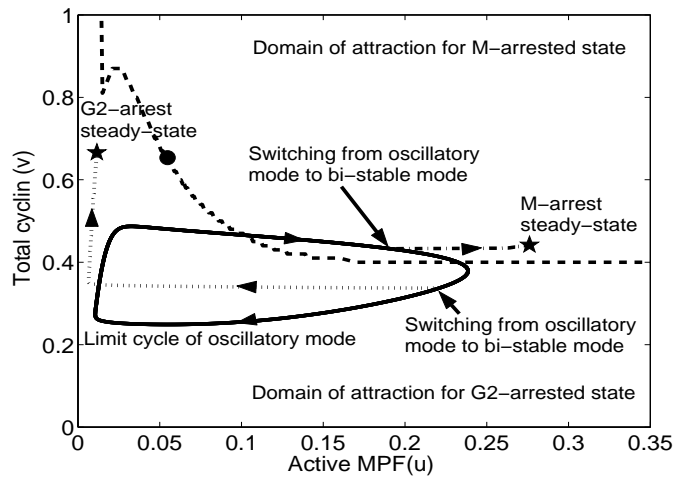
$k'_2$	$k''_2$	$k_{wee}$	Mode	$M$ -arrest state	$G2$ -arrest state	Reference
0.01	10	3.5	$G2$ -arrest	n/a	(0.016, 0.802)	
0.01	0.5	2.0	$M$ -arrest	(0.202, 0.329)	n/a	
0.015	0.1	3.5	Bi-stable	(0.276, 0.442)	(0.012, 0.666)	
0.01	10	2.0	Oscillatory	n/a	n/a	Figure 8.2(b)
0.01	10	2.5	Oscillatory	n/a	n/a	Figure 8.4

cyclin degradation) can induce a transition from the oscillatory mode of MPF activity (early embryo stage) to the bi-stable mode. However, it is not clear from this analysis alone whether the cell will end up in a  $G2$ - or an  $M$ -arrested state upon switching. To address this question, we initially compute the domains of attraction of both steady-states in the bi-stable mode. This is done using a Lyapunov function of the form  $V = (u - u_s)^4 + 10(v - v_s)^2$ , where  $u_s$  and  $v_s$  are the steady-state values. The basic idea here is to compute, for each steady-state, the region in the  $(u, v)$  space where the time-derivative of  $V$  is negative-definite along the trajectories of the dynamical system of Equation 8.4, and then use this region to obtain an estimate of the domain of attraction. While several candidate functions could be used, this particular function was found to yield acceptable estimates of the domains of attraction in the sense that the region obtained for each steady-state covered a distinct and large part of the operating range considered (the two regions were mostly separated from one another along the separatrix running through the three steady-states, and their union covered the entire range) with little overlapping between the two regions occurring only in the vicinity of the two steady-states. Computer simulations were then used to check the regions of overlap and determine which domain of attraction they were contained in.





(a)

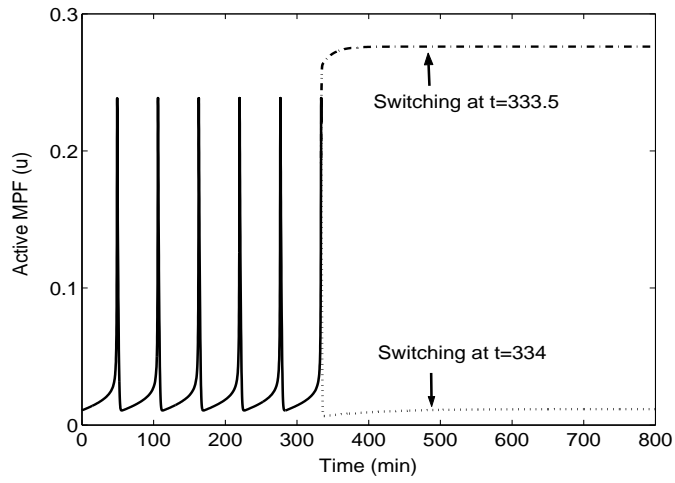


(b)

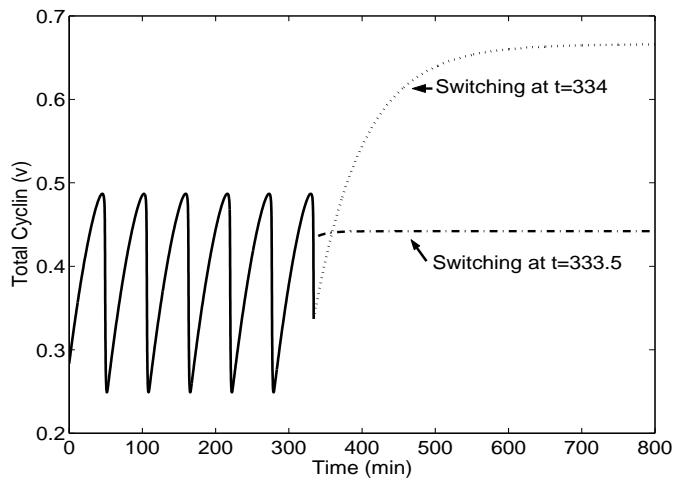
Figure 8.2: (a) A plot showing the overlap of the limit cycle of the oscillatory mode with the domains of attraction for the M-arrested steady-state (entire area above dashed curve) and for the G2-arrested steady-state (entire area below the dashed curve), (b) A plot showing that switching from the oscillatory to the bi-stable mode moves the system to different steady-states depending on where switching takes place. In both cases, the oscillatory mode is fixed at  $k'_2 = 0.01$ ,  $k''_2 = 10$ ,  $k_{wee} = 2.0$ .

The domains of attraction for both steady-states are depicted in Figure 8.2(a). The entire area above the dashed curve (the separatrix) is the domain of attraction of the M-arrested state while the area below is the domain of attraction of the G2-arrested state. Both stable steady-states are denoted by asterisks on the plot and the unstable steady-state is denoted by a circle on the separatrix. By plotting the limit cycle (obtained from the oscillatory mode) on the same plot, we see that a portion of the limit cycle lies within the domain of attraction of the M-arrested steady-state (segment A in Figure 8.2(a)) while the rest is completely within the domain of attraction of the G2-arrested steady-state. Based on this analysis, we conclude that switching from the oscillatory mode to the bi-stable mode would move the cell to the G2-arrested state only if the transition occurs at times when the state is not on segment A, while it would end up in the M-arrested state if switching were to occur on segment A. This conclusion is verified by the dotted and dash-dotted state trajectories, respectively, shown in Figure 8.2(b). The corresponding plots of the time-evolution of the states in both switching scenarios are given in Figure 8.3 for two representative switching times. Note that because of the periodic nature of the solution in the oscillatory mode, there are many time-intervals, between  $t = 0$  and  $t = 333.5$  min, when the limit cycle trajectory is on segment A. These intervals are separated by one period of the limit cycle. Switching during any of these intervals to the bi-stable mode moves the system to the M-arrested state. Similarly, there are many time-intervals when the trajectory is not on segment A. Switching during any of those intervals will land the system at the G2-arrested state.

Figure 8.4 shows the limit cycle resulting when the rate of inhibition of *Wee1* is increased to  $k_{wee} = 2.5$  (with  $k'_2$  and  $k''_2$  remaining fixed at 0.01 and 10, respectively). Comparing with Figure 8.2(a), we observe that a larger portion of the limit cycle



(a)



(b)

Figure 8.3: The time evolution plots of (a) active MPF, and (b) total cyclin upon switching from the oscillatory to the bi-stable mode at two representative switching times. At  $t = 333.5$  min, the state trajectory lies on segment A (see Figure 8.2(a)) and therefore switching lands the state in the M-arrested steady-state (dash-dotted line), while at  $t = 334$  min, switching lands the state in the G2-arrested steady-state (dotted line). In both cases, the oscillatory mode is fixed at  $k'_2 = 0.01$ ,  $k''_2 = 10$ ,  $k_{wee} = 2.0$ .

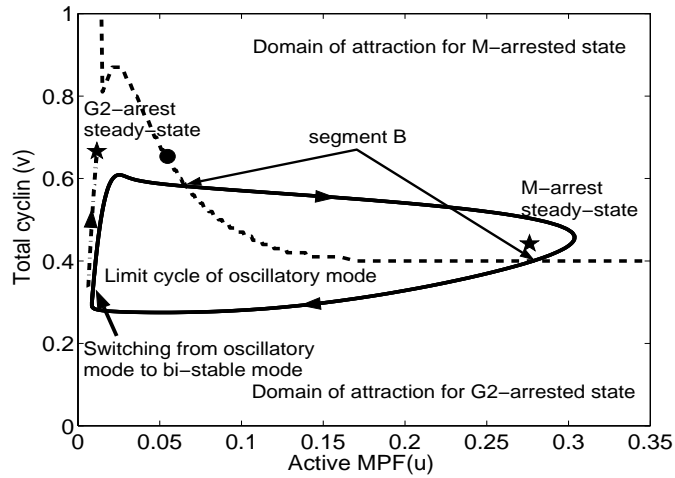
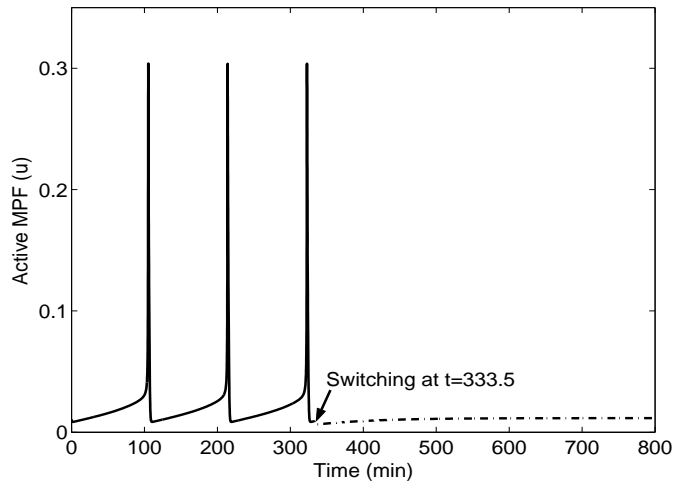
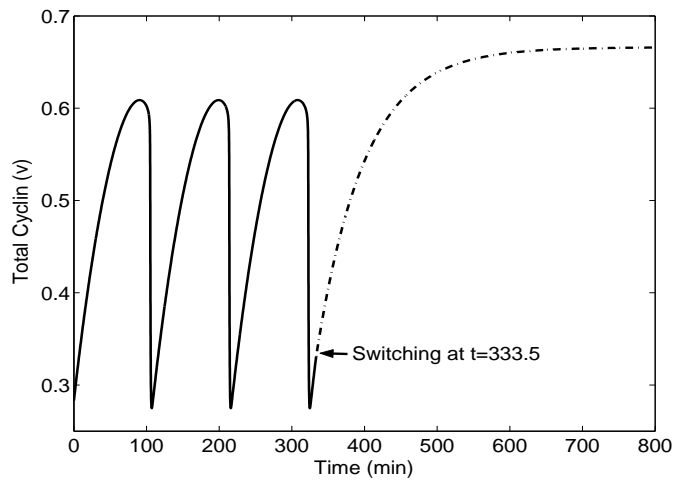


Figure 8.4: A plot showing that switching from the oscillatory mode (of the following parameter values:  $k_2' = 0.01$ ,  $k_2'' = 10$ ,  $k_{wee} = 2.5$ ) to the bi-stable mode at same time as in Figure 8.2(b) ( $t = 333.5$  min) moves the system to G2-arrest steady-state (instead of M-arrest steady-state) because switching does not occur on segment B. Note that the portion of the limit cycle overlapping the domain of attraction of the M-arrested steady-state (segment B) is larger than the one in Figure 8.2(a) (segment A).

(segment B in Figure 8.4) lies within the domain of attraction of the M-arrested steady-state. Therefore, unlike the case of Figure 8.2(a), when switching from the oscillatory mode to the bi-stable mode takes place at  $t = 333.5$  min, the state is not within the domain of attraction of the M-arrested steady-state. Switching in this case lands the system at the G2-arrested steady-state. The corresponding time-evolution plots are given in Figure 8.5.



(a)



(b)

Figure 8.5: The time evolution plots of (a) active MPF, and (b) total cyclin upon switching from the oscillatory to the bi-stable mode at  $t = 333.5$  min. In both cases, the oscillatory mode is fixed at  $k'_2 = 0.01$ ,  $k''_2 = 10$ ,  $k_{wee} = 2.5$

## 8.5 Application to the Bacteriophage $\lambda$ -Switch System

We consider an example of a biological switch observed in the bacteriophage  $\lambda$ . An excellent review and detailed description of the molecular regulatory mechanisms in the bacteriophage  $\lambda$ -switch can be found in [138]. Bacteriophage  $\lambda$  is a virus capable of infecting *Escherichia coli* bacteria. The virus attaches its tail to the surface of host bacterium cell, drills a hole in the cell wall, and squirts its chromosome into the bacterium, leaving its coat behind.  $\lambda$  is an obligate parasite – it must inject its DNA into the bacterium to multiply. Upon infection, it can follow either one of two different pathways. First, the injected phage chromosome lysogenizes its host: all but one of the phage genes are turned off, and one phage chromosome, called prophage, becomes part of the host chromosome. As the lysogen (the bacterium bearing the prophage) grows and divides, the prophage is passively replicated and quiescently distributed to the progeny bacteria. Second, the phage chromosome enters the lytic mode: various sets of phage genes are turned on and off according to a precisely regulated program, the  $\lambda$  chromosome is extensively replicated, new head and tail proteins are synthesized, new phage particles are formed within bacterium, and some 45 minutes following the infection the bacterium lyses and releases about 100 progeny phage. Once the virus is in the lysogenic state, it can shift to the lysis state under certain conditions, for example, if the bacterial culture is irradiated with ultraviolet (UV) light.

The molecular regulatory mechanism responsible for the lysogeny/lysis decision is known as the phage  $\lambda$ -switch. The switch to lytic growth is called induction. A schematic representation of the  $\lambda$ -switch performance in the lysogenic and lytic steady-states is shown in Figure 8.6.

To understand how the switch works, we need to consider two regulatory genes

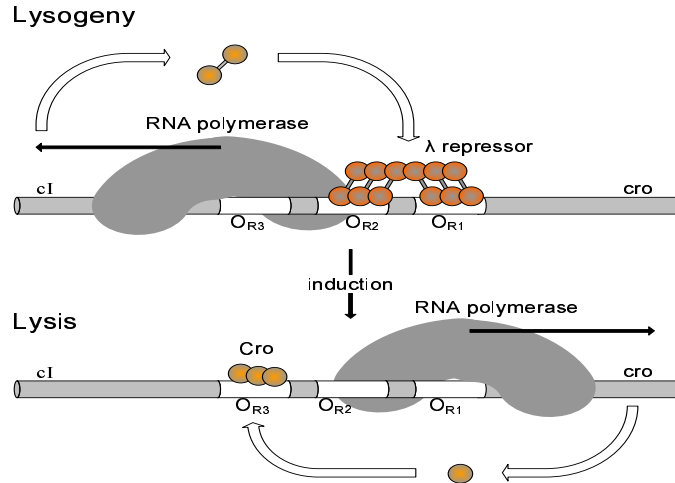


Figure 8.6: A schematic representation of the molecular mechanism responsible for the lysogenic to lytic mode transition in the bacteriophage  $\lambda$ .

( $cI$  and  $cro$ ) and the regulatory region called  $O_R$  (right operator). In a lysogen,  $cI$  is on and  $cro$  is off, and vice versa when lytic growth ensues. The operator comprises three binding sites ( $O_{R1}$ ,  $O_{R2}$ , and  $O_{R3}$ ) that overlap two opposing promoters. One of these,  $P_R$ , directs transcription of lytic genes and the other,  $P_{RM}$ , directs transcription of the  $cI$  gene. In a lysogen, the  $\lambda$  repressor (the product of  $cI$  gene), at  $O_R$ , is bound at the two adjacent sites  $O_{R1}$  and  $O_{R2}$ . At these positions, it performs two functions: it represses rightward transcription from the promoter  $P_R$ , thereby turning off expression of  $cro$  and other lytic genes; simultaneously it activates transcription of its own gene from the promoter  $P_{RM}$ . Upon induction, repressor vacates the operator and transcription from  $P_R$  commences spontaneously. The first newly made protein is  $Cro$ . This protein binds first to  $O_{R3}$ , apparently helping to abolish repressor synthesis.

To illustrate the application of our methodology, we consider the following bacteriophage  $\lambda$  synthetic network model described in [76] (other more detailed models

can also be used):

$$\begin{aligned}\frac{dx}{dt} &= \frac{m_x(1 + x^2 + \alpha\sigma_1x^4)}{Q(x, y)} - \gamma_x x \\ \frac{dy}{dt} &= \frac{m_y\rho_y(1 + y^2)}{Q(x, y)} - \gamma_y y\end{aligned}\tag{8.5}$$

where

$$Q(x, y) = 1 + x^2 + \sigma_1x^4 + \sigma_1\sigma_2x^6 + y^2 + (\beta_1 + \beta_2)y^4 + \beta_1\beta_3y^6 + \sigma_1\beta_4x^4y^2 + \beta_5x^2y^2\tag{8.6}$$

$x$  and  $y$  represent dimensionless concentrations of the *CI* and *Cro* proteins, respectively;  $t$  represents dimensionless time;  $\sigma_1$  and  $\sigma_2$  are prefactors denoting the relative affinities for dimer binding to  $O_{R1}$  versus that of binding to  $O_{R2}$  and  $O_{R3}$ , respectively;  $\alpha > 1$  represents degree to which transcription is enhanced by dimer occupation of  $O_{R2}$ ;  $\beta_1$ – $\beta_5$  represent prefactors denoting binding strengths on reactions entailing the binding of *Cro* to different operator sites (see Equation 17 in [76]); the integers  $m_x$  and  $m_y$  represent the plasmid copy numbers for the two species;  $\rho_y$  represents a constant related to the scaling of  $y$  relative to  $x$ ;  $\gamma_x$  and  $\gamma_y$  are directly proportional to the decay rates of *CI* and *Cro* proteins, respectively. The even polynomials in  $x$  occur due to dimerization and subsequent binding to the promoter region. The  $x^4$  term represents the transcription when the two operator sites  $O_{R1}$  and  $O_{R2}$  are occupied ( $x^2x^2$ ). The  $x^6$  term represents the occupation of all three operator sites and arises in the denominator because dimer occupation of  $O_{R3}$  inhibits polymerase binding and shuts off transcription. The values of the model parameters in Equations 8.5-8.6 are given in Table 8.3. The steady-state values for different *CI* and *Cro* degradation rates are given in Table 8.4.

Bifurcation and phase-plane analysis of the above model show that, by changing the values of  $\gamma_x$  and  $\gamma_y$ , the system of Equation 8.5 can exhibit one of the following modes of behavior:



Table 8.3: Parameter values for the bacteriophage  $\lambda$  model in Equation 8.5 [76].

$\rho_y$	=	62.92
$\alpha$	=	11
$m_x$	=	1
$m_y$	=	1
$\sigma_1$	=	2
$\sigma_2$	=	0.08
$\beta_1$	=	0.08
$\beta_2$	=	0.08
$\beta_3$	=	0.08
$\beta_4$	=	1
$\beta_5$	=	1

Table 8.4: Steady-state values  $(x_s, y_s)$  for the lysogenic, lytic, and unstable steady-states for different values of  $\gamma_x$  and  $\gamma_y$ .

$\gamma_x$	$\gamma_y$	Lysogenic state	Lytic state	Unstable state	Domain of attraction
0.004	0.008	(32.39,0)	(0,16.22)	(2.79,15.27)	Figure 8.8(a)
0.05	0.008	(13.71,0.01)	(0,16.22)	(4.89,6.38)	Figure 8.7
0.1	0.008	(10.75,0.03)	(0,16.22)	(5.37,4.10)	Figure 8.8(b)
1	0.008	n/a	(0,16.22)	n/a	
0.004	1	(32.39,0)	n/a	n/a	
0.05	0.0005	(13.71,0.11)	(0,28.59)	(10.24,3.47)	Figure 8.11(a)
0.05	0.06	(13.71,0)	(0,10.60)	(2.64,7.65)	Figure 8.11(b)

- A mode with a single globally asymptotically stable equilibrium point corresponding to the lysogenic steady-state (low  $\gamma_x$  and high  $\gamma_y$ ).
- A mode with a single globally asymptotically stable equilibrium point corresponding to the lytic steady-state (high  $\gamma_x$  and low  $\gamma_y$ ).
- A bi-stable mode where the stable lysogenic and lytic steady-states coexist together with a third unstable steady-state.

Note from Table 8.4 that for a fixed  $\gamma_y$ , as the degradation rate of protein *CI* is increased (larger  $\gamma_x$  value), the lysogenic steady-state keeps shifting to smaller concentrations until the system exhibits only the lytic steady-state (the lysogenic steady-state vanishes). By contrast, for a fixed  $\gamma_x$ , when the degradation rate of protein *Cro* is increased (larger  $\gamma_y$  value), the lytic steady-state keeps shifting to smaller concentrations until the system exhibits only the lysogenic steady-state (the lytic steady-state vanishes).

Table 8.5: Lyapunov functions used in estimating the invariant set  $\Omega_{lysogenic}$  for the lysogenic state and the invariant set  $\Omega_{lytic}$  for the lytic state.

$\gamma_x$	$\gamma_y$	Lyapunov Function for $\Omega_{lysogenic}$	$c^{max}$
0.004	0.008	$V = (x - x_s)^2 + (y - y_s)^2$	800
0.1	0.008	$V = (x - x_s)^2 + 0.6(y - y_s)^4$	100
0.05	0.0005	$V = (x - x_s)^2 + (y - y_s)^6$	150
0.05	0.06	$V = (x - x_s)^2 + 0.5(y - y_s)^2$	150
$\gamma_x$	$\gamma_y$	Lyapunov Function for $\Omega_{lytic}$	$c^{max}$
0.004	0.008	$V = 20(x - x_s)^2 + (y - y_s)^2$	100
0.1	0.008	$V = 0.5(x - x_s)^2 + (y - y_s)^2$	150
0.05	0.0005	$V = (x - x_s)^2 + 0.01(y - y_s)^4$	700
0.05	0.06	$V = 20(x - x_s)^2 + (y - y_s)^2$	100

Focusing on the bi-stable mode, we initially compute estimates of the domains of attraction of both steady-states for different values of the *CI* and *Cro* protein

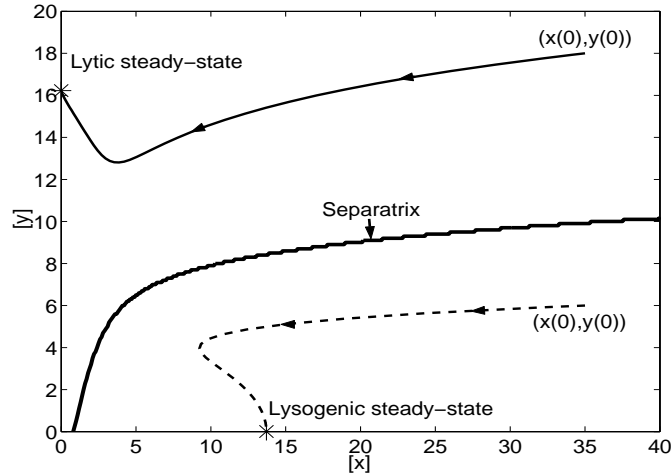
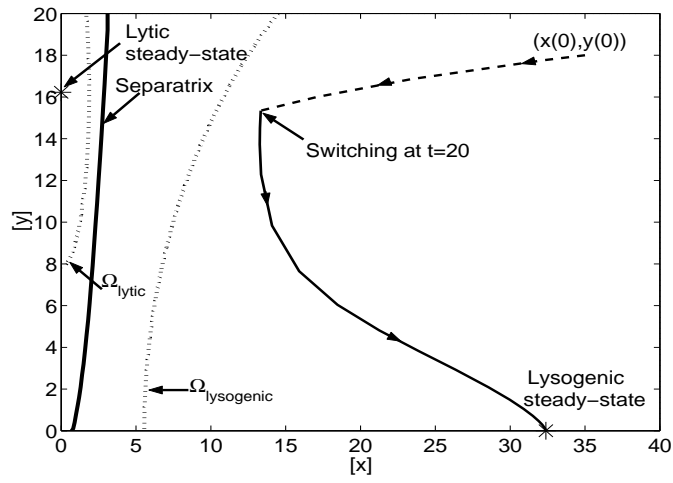
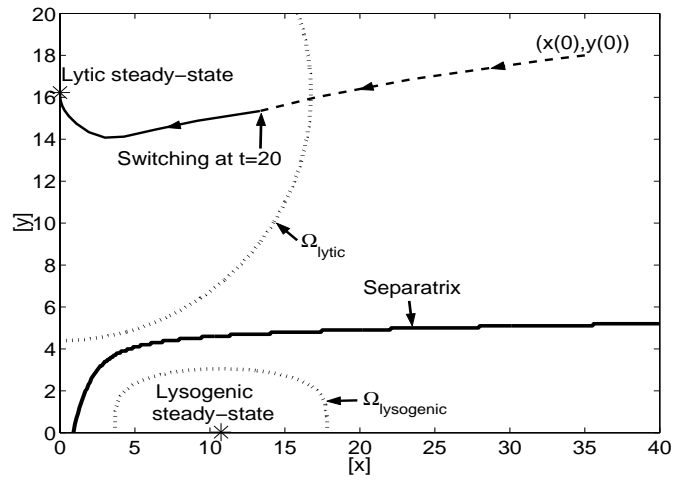


Figure 8.7: A phase plot for the moderate *CI* degradation mode showing that an initial condition within the lysogenic domain of attraction (entire area below the dotted curve) will converge to the lysogenic steady-state (dashed trajectory) and that an initial condition within the lytic domain of attraction (entire area above the dotted curve) will converge to the lytic steady-state (solid trajectory). Here, the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

degradation rate. Due to the complex nonlinearity of the system – relative to that of the cell cycle model – the Lyapunov function used in the cell cycle example did not yield good estimates of the domain of attraction for the  $\lambda$ -switch system. However, we were able to get “conservative” estimates of the domains of attraction using several other polynomial Lyapunov functions which are listed in Table 8.5. For each steady-state, we initially used the corresponding  $V$  to determine the region,  $\Pi$ , where  $\dot{V} < 0$  and then constructed an invariant set (a level set) within this region,  $\Omega = \{x : V(x) \leq c_{max}\}$ , where  $c_{max}$  is a positive constant for which  $\Omega$  is contained in  $\Pi$ . The boundaries of the invariant sets,  $\Omega_{lysogenic}$  and  $\Omega_{lytic}$ , are depicted by the dotted lines in Figures 8.8, 8.11, 8.14, and 8.16) for the lysogenic state and lytic state (note that, for each level set, only the part that is contained within the given  $x$ - $y$  range is shown). To get an idea of the possible conservatism of these estimates, we also used computer



(a)



(b)

Figure 8.8: A phase plot showing the system of Equation 8.5 being initialized using  $\gamma_x = 0.05$  (dashed trajectory) and undergoing: (a) a decrease in the degradation rate of *CI* protein (to  $\gamma_x = 0.004$ ) at  $t = 20$ , leading the state to converge to the lysogenic steady-state, and (b) an increase in the degradation rate of *CI* protein (to  $\gamma_x = 0.1$ ) at  $t = 20$ , leading the state to converge to the lytic steady-state. In both cases, the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

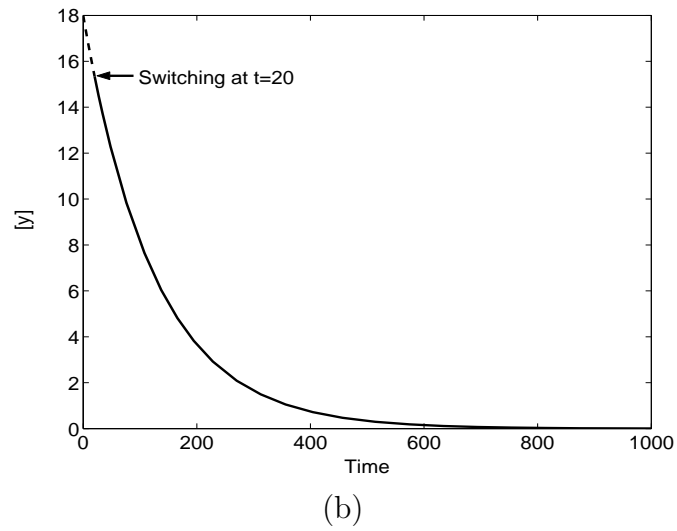
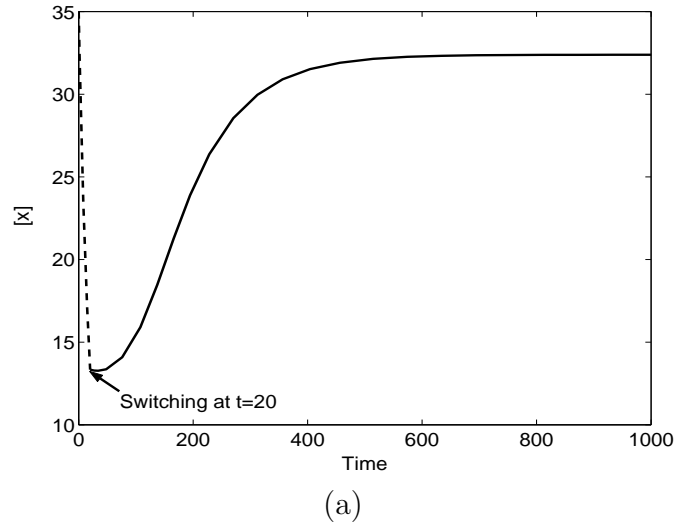


Figure 8.9: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.004$  mode at  $t = 20$  and converges (solid lines) to the lysogenic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

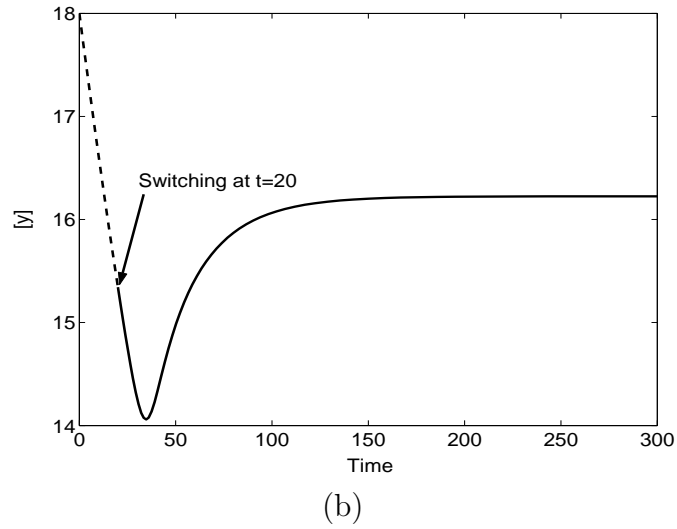
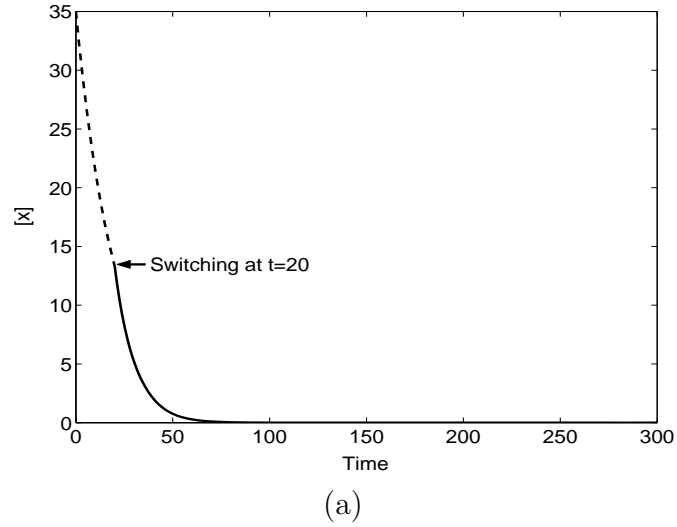


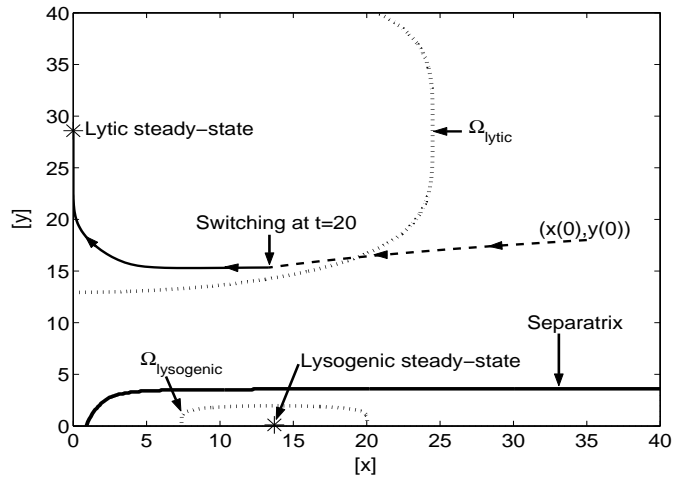
Figure 8.10: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.1$  mode at  $t = 20$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

simulations to compare, for each steady-state, the entire domain of attraction (shaded regions) with the estimate provided by the corresponding level set.

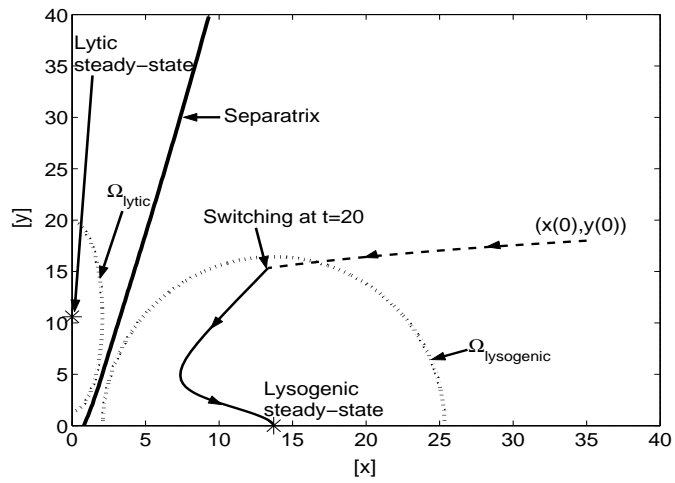
Figures 8.7, 8.8(a), and 8.8(b) show the domains of attraction for the lysogenic and lytic steady-states for: (1) a moderate *CI* degradation rate ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.008$ ), (2) a relatively low *CI* degradation rate ( $\gamma_x = 0.004$ ,  $\gamma_y = 0.008$ ), and (3) a relatively high *CI* degradation rate ( $\gamma_x = 0.1$ ,  $\gamma_y = 0.008$ ), respectively, keeping the *Cro* protein degradation rate constant. Figures 8.7, 8.11(a), and 8.11(b) show the domains of attraction for the lysogenic and lytic steady-states for: (1) a moderate *Cro* degradation rate ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.008$ ), (2) a relatively low *Cro* degradation rate ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.0005$ ), and (3) a relatively high *Cro* degradation rate ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.06$ ), respectively, keeping the *CI* protein degradation rate constant. The entire area below (or to the right of) the separatrix is the entire domain of attraction for the lysogenic steady-state, while the area above (or to the left of) the separatrix is the entire domain of attraction for the lytic steady-state. Both stable steady-states are denoted by asterisks on each plot.

It is clear from the plots that an increase in the *CI* degradation rate results in a smaller domain of attraction for the lysogenic state (and a larger one for the lytic state) and vice versa. In the limiting case of very high degradation rates, the lysogenic state vanishes and the domain of attraction of the lytic state occupies the entire state-space (single globally asymptotically stable equilibrium point). The opposite trend is observed when the *Cro* protein degradation rate is increased. In particular, increasing  $\gamma_y$  leads to a smaller domain of attraction for the lytic state and a larger one for the lysogenic state. For very high *Cro* degradation rates, the lytic steady-state vanishes and the domain of attraction for the lysogenic state turns into the entire state-space.

Therefore, in the bi-stable mode, the initial condition plays a critical role in decid-



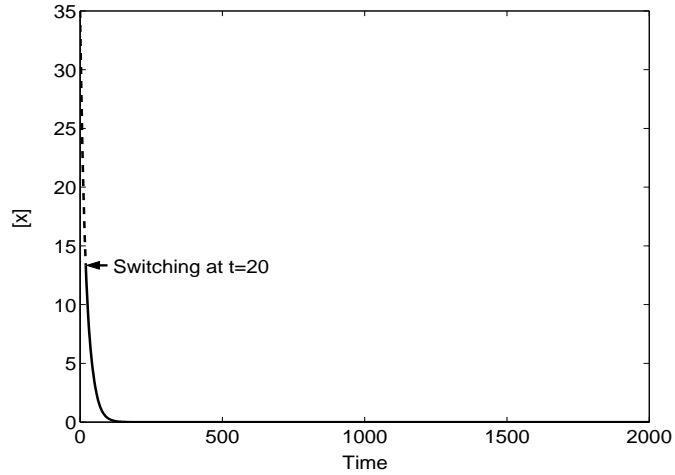
(a)



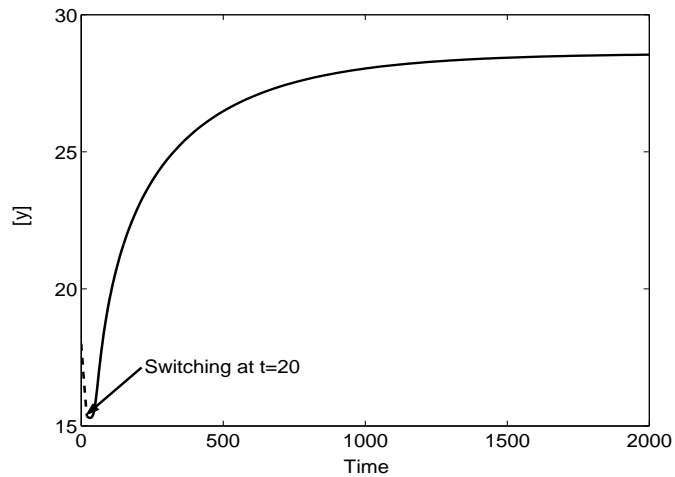
(b)

Figure 8.11: A phase plot showing the system of Equation 8.5 being initialized using  $\gamma_y = 0.008$  (dashed trajectory) and undergoing: (a) a decrease in the degradation rate of *Cro* protein (to  $\gamma_y = 0.0005$ ) at  $t = 20$ , leading the state to converge to the lytic steady-state, and (b) an increase in the degradation rate of *Cro* protein (to  $\gamma_y = 0.06$ ) at  $t = 20$ , leading the state to converge to the lysogenic steady-state. In both cases, the *CI* degradation rate is fixed at  $\gamma_x = 0.05$ .





(a)



(b)

Figure 8.12: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system initialized at  $(x(0), y(0)) = (35, 18)$  undergoes a transition from the  $\gamma_y = 0.008$  mode (dashed lines) to the  $\gamma_y = 0.0005$  mode at  $t = 20$  and converges (solid lines) to the lytic steady-state. The *CI* degradation rate is fixed at  $\gamma_x = 0.05$ .

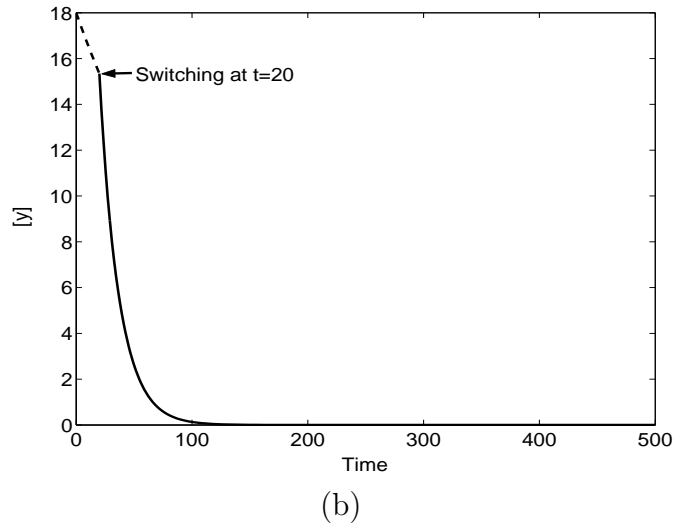
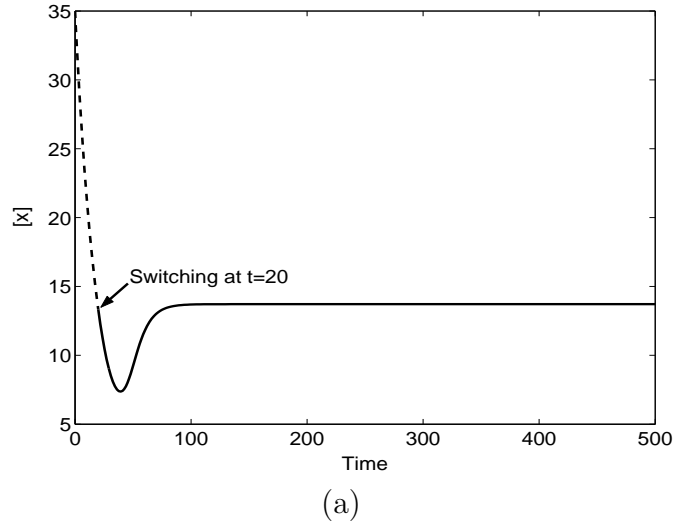


Figure 8.13: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system initialized at  $(x(0), y(0)) = (35, 18)$  undergoes a transition from the  $\gamma_y = 0.008$  mode (dashed lines) to the  $\gamma_y = 0.06$  mode at  $t = 20$  and converges (solid lines) to the lysogenic steady-state. The *CI* degradation rate is fixed at  $\gamma_x = 0.05$ .

ing which steady-state the bacteriophage  $\lambda$  will attain. Also, the size of the domain of attraction for each state helps explain why the lysogenic state is more likely to be observed under a given set of conditions (for example, [147]), while the lytic state is more likely to be seen under a different set of conditions. Figure 8.7 shows that starting from an initial condition of high  $CI$  and  $Cro$  concentrations, the phage ends up in the lytic state since the initial condition is within its domain of attraction (solid trajectory). Initializing the system, however, at high  $CI$  but low  $Cro$  concentrations drives the phage to the lysogenic state (dashed trajectory).

We now demonstrate the effect of switching in the  $CI$  protein degradation rate on whether the bacteriophage will exhibit the lytic or lysogenic steady-state. To this end, we initialize the system within the moderate  $CI$  degradation mode ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.008$ ) at the initial condition  $(x(0), y(0)) = (35, 18)$  and allow it to evolve in this mode until, at  $t = 20$ , a mode transition is enforced (see dashed trajectories in Figures 8.8(a-b)). The results show that, for a fixed transition time, depending on which mode is being switched in, the phage takes a different path. For example, Figure 8.8(a) shows that when the system switches to the relatively low  $CI$  degradation mode ( $\gamma_x = 0.004$ ,  $\gamma_y = 0.008$ ) at  $t = 20$ , the system state is within the invariant set of the lysogenic steady-state ( $\Omega_{lysogenic}$ ) and, therefore, the phage ends up with lysogeny. Figure 8.8(b), on the other hand, shows that when the relatively high  $CI$  degradation mode ( $\gamma_x = 0.1$ ,  $\gamma_y = 0.008$ ) is switched in at  $t = 20$ , the system state is within the invariant set of the lytic steady-state ( $\Omega_{lytic}$ ) and, therefore, the phage ends up with lysis instead. The time evolution plots for both scenarios are depicted in Figures 8.9-8.10, respectively.

Figure 8.8(b) gives some insight into the implications of using a conservative estimate of the domain of attraction as the basis for switching, in lieu of the true domain

of attraction (which could be more computationally expensive to obtain). In particular, if the relatively high *CI* degradation mode ( $\gamma_x = 0.1$ ,  $\gamma_y = 0.008$ ) is switched in before the states enter the invariant set of the lytic state  $\Omega_{lytic}$ , then, having no knowledge about what the actual domain of attraction looks like, the only conclusion we would be able to make is that there is no guarantee that the phage would end up in the lytic state if switching were to take place at such a time. Switching has to be “delayed” until the state enters  $\Omega_{lytic}$  in order to guarantee that the phage would end up with lysis.

To demonstrate the effect of switching in the *Cro* protein degradation rate, the system is initialized within the moderate *Cro* degradation mode ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.008$ ) at the same initial condition  $(x(0), y(0)) = (35, 18)$  and allowed to evolve in this mode until, at  $t = 20$ , a mode transition is enforced (see dashed trajectories in Figures 8.11(a-b)). The results show that, for a fixed transition time, depending on which mode is being switched in, the phage takes a different path. For example, Figure 8.11(a) shows that when the system switches to the relatively low *Cro* degradation mode ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.0005$ ) at  $t = 20$ , the system state is within  $\Omega_{lytic}$  and, therefore, the phage ends up with lysis. Figure 8.11(b), on the other hand, shows that when the relatively high *Cro* degradation mode ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.06$ ) is switched in at  $t = 20$ , the system state is within  $\Omega_{lysogenic}$  and, therefore, the phage ends up with lysogeny instead. The time evolution plots for both scenarios are depicted in Figures 8.12-8.13, respectively.

So far in our analysis, we have fixed the transition time and showed that which mode is activated at that time determines the final state of the phage. Here, we demonstrate the effect of varying the transition time, for a given mode transition, on the steady-state behavior of the phage. To this end, we reconsider the switching

scenario presented in Figure 8.8(a), where the system switches from the moderate ( $\gamma_x = 0.05$ ) to the relatively low ( $\gamma_x = 0.004$ ) *CI* degradation mode and the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ . However, instead of carrying out the transition at  $t = 20$  as in Figure 8.8(a), the switch is delayed until  $t = 70$ . The result is depicted in Figure 8.14 which shows that at  $t = 70$ , the system state is within the invariant set of the lytic steady-state ( $\Omega_{lytic}$ ) and, therefore, the phage ends up with lysis. The corresponding time evolution plots are given in Figure 8.15. By comparing Figure 8.8(a) with Figure 8.14, we conclude that an early transition from moderate to relatively low *CI* degradation rate favors lysogeny, while a late transition favors lysis.

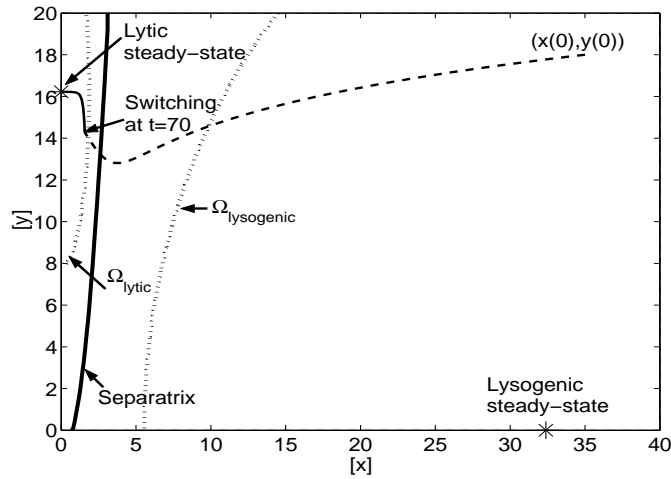


Figure 8.14: A phase plot showing the system undergoing a transition from the  $\gamma_x = 0.05$  mode (dashed trajectory) to the  $\gamma_x = 0.004$  at  $t = 70$  and converging (solid trajectory) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

In the last simulation run, we demonstrate the effect of the initial condition on the outcome of switching for a given transition time. To this end, we initialize the system within the moderate *CI* degradation mode ( $\gamma_x = 0.05$ ,  $\gamma_y = 0.008$ ) at an initial condition different from the one considered in Figure 8.8 and characterized

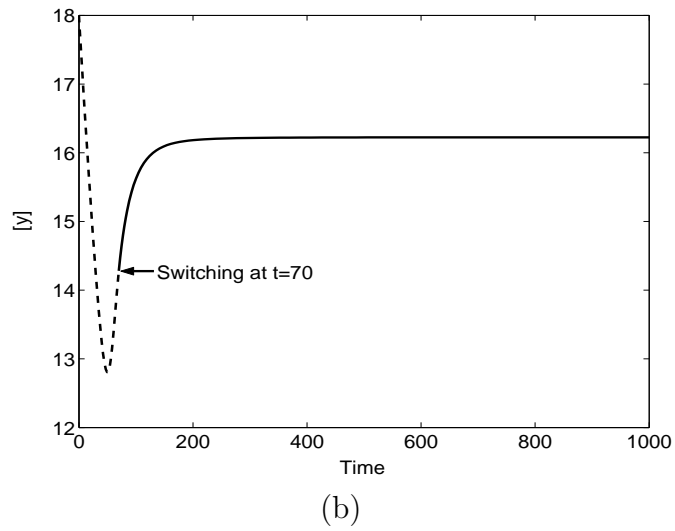
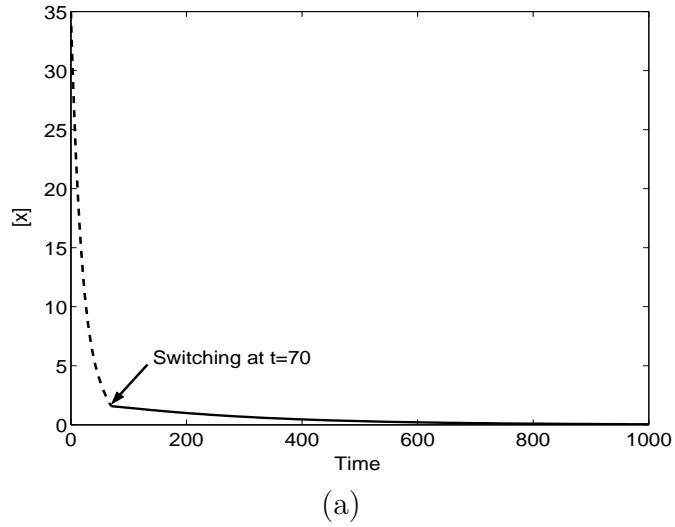
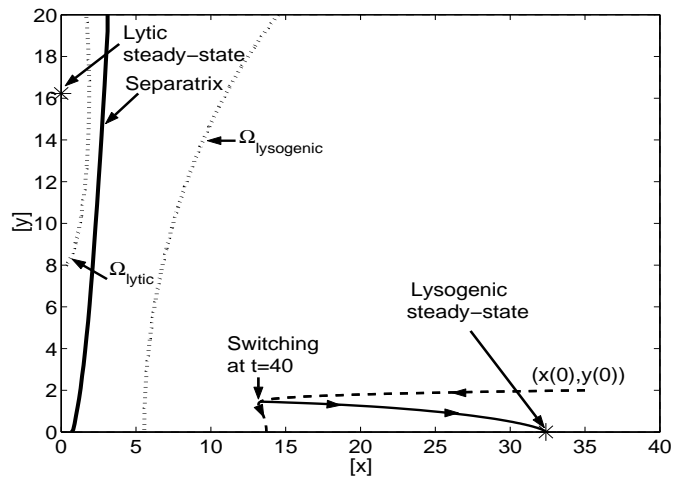
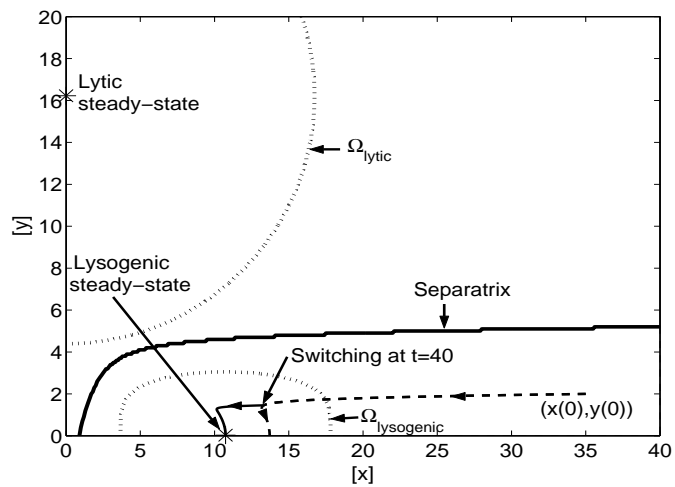


Figure 8.15: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.004$  mode at  $t = 70$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .



(a)



(b)

Figure 8.16: A phase plot showing the system of Equation 8.5 being initialized using  $\gamma_x = 0.05$  (dashed trajectory) and undergoing: (a) a decrease in the degradation rate of *CI* protein (to  $\gamma_x = 0.004$ ) at  $t = 40$  and (b) an increase in the degradation rate of *CI* protein (to  $\gamma_x = 0.1$ ) at  $t = 40$ , both leading the state to converge to the lysogenic steady-state. In both cases, the *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

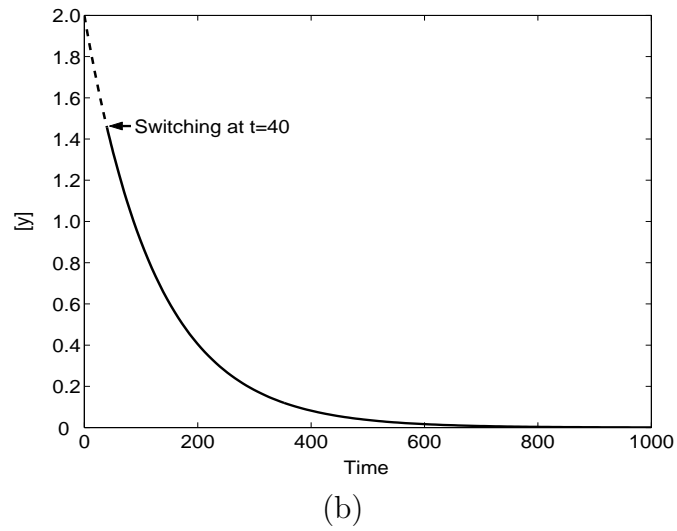
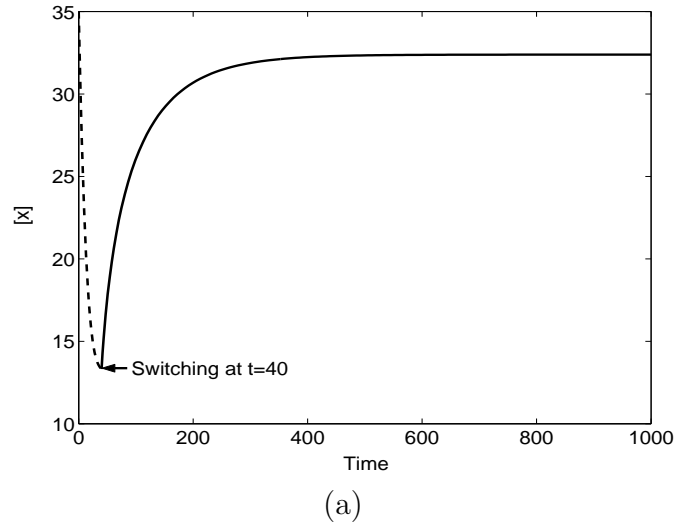


Figure 8.17: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system initialized at  $(x(0), y(0)) = (35, 2)$  undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.004$  mode at  $t = 40$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .



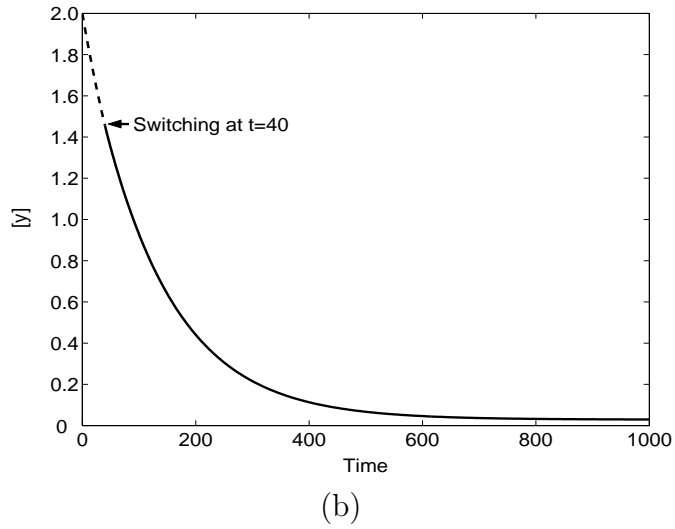
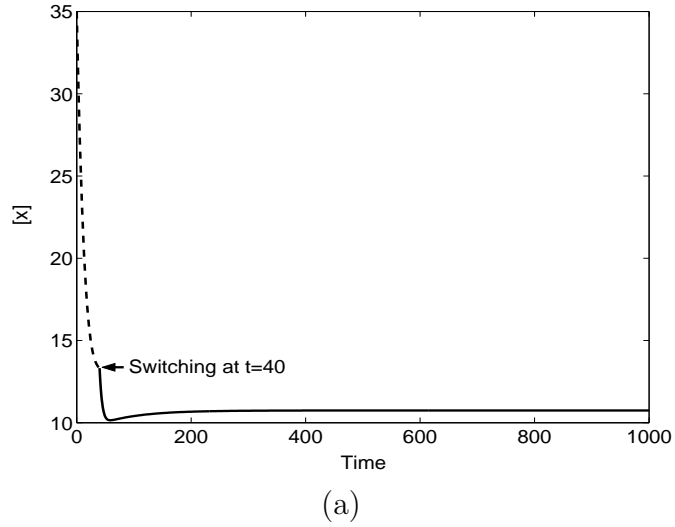


Figure 8.18: The time evolution plots of the *CI* (left) and *Cro* (right) protein concentrations when the system initialized at  $(x(0), y(0)) = (35, 2)$  undergoes a transition from the  $\gamma_x = 0.05$  mode (dashed lines) to the  $\gamma_x = 0.1$  mode at  $t = 40$  and converges (solid lines) to the lytic steady-state. The *Cro* degradation rate is fixed at  $\gamma_y = 0.008$ .

by high concentration of *CI* and low concentration of *Cro* ( $x(0) = 35$ ,  $y(0) = 2$ ). We allow the system to evolve in this mode until, at  $t = 40$ , a mode transition is enforced (see dashed trajectories in Figures 8.16(a-b)). The results show that, for a fixed transition time, switching close to a particular steady-state will converge to that particular steady-state, independently of which mode is being switched in. For example, Figure 8.16(a) shows that when the system switches to the relatively low *CI* degradation mode ( $\gamma_x = 0.004$ ,  $\gamma_y = 0.008$ ) at  $t = 40$ , the state is within the invariant set of the lysogenic steady-state ( $\Omega_{lysogenic}$ ) and, therefore, the phage ends up with lysogeny. Similarly, Figure 8.16(b) shows that when the relatively high *CI* degradation mode ( $\gamma_x = 0.1$ ,  $\gamma_y = 0.008$ ) is switched in at  $t = 40$ , the state is again within the invariant set of the lysogenic steady-state ( $\Omega_{lysogenic}$ ) and, therefore, the phage ends up with lysogeny (albeit with a smaller steady-state concentration of *CI* protein). The time evolution plots for both scenarios are depicted in Figures 8.17-8.18, respectively. Note that this result is different from the one obtained in Figure 8.8 where the final steady-state behavior is dependent on which mode is being switched in. The difference lies in the fact that the system state at the switching time considered in Figure 8.16 is contained within the invariant set of the lysogenic steady-state ( $\Omega_{lysogenic}$ ) for both the low and high *CI* degradation modes, and therefore only the lysogenic steady-state can be observed regardless of whether the low or high *CI* degradation mode is activated.

## 8.6 Conclusions

In this chapter, a methodology for the analysis of mode transitions in biological networks was presented. The proposed approach was predicated upon the notion of orchestrating switching between the domains of attraction of the steady-states

of the constituent modes. The proposed method was demonstrated using models of biological networks that arise in cell cycle regulation and the bacteriophage  $\lambda$ -switch system. The proposed approach has implications both for understanding the outcome of naturally-occurring mode transitions and for the ability to manipulate network behavior by enforcing mode transitions.

## Chapter 9

# Conclusions

This work proposed a methodology for design of fault-tolerant control systems for nonlinear processes with actuator constraints and measurement data loss in presence of uncertainties and disturbances for handling actuator and sensor faults, incorporating performance and robustness considerations. The proposed approach was predicated upon the idea of integrating fault-detection, feedback control, supervisory over networks, and hybrid systems.

Initially, a family of candidate control configurations, characterized by different manipulated inputs, were identified. For each control configuration, a Lyapunov-based nonlinear feedback controller, that enforced asymptotic closed-loop stability in the presence of constraints, was designed. For each control configuration, the stability region (i.e., the set of initial conditions starting from where closed-loop stabilization under continuous availability of measurements is guaranteed) as well as the maximum allowable data loss rate which preserved closed-loop stability was computed. A fault-detection filter was used to compute the expected closed-loop behavior in the absence of faults. Deviations of the process states from the expected closed-loop behavior were used to detect faults. To deal with the problem of lack of process state measurements,

a nonlinear observer was designed to generate estimates of the states, which were then used to implement the state feedback controller and the fault-detection filter.

A switching policy was 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 was detected. The switching laws were implemented by a higher-level supervisor that constantly monitors the process and communicates with the various control configurations over a network. The effects of delays in fault-detection, network communication and actuator activation were taken explicitly into account in executing the switching logic. Reducing network usage using knowledge of the plant dynamics was considered due to limited network bandwidth. Specifically, the controller used the explicit model of the plant and made possible stabilization of the plant even under the presence of delays. The efficacy and implementation of the proposed approach were demonstrated through a single unit chemical reactor, a cascading multi-unit chemical process example, a polyethylene reactor, batch and continuous crystallizers.

We also extended Lyapunov-based tools, hybrid systems theory, and concept of stability regions to biological networks. We presented a methodology for the analysis and control of mode transitions in biological networks. The proposed approach was predicated upon the notion of orchestrating switching between the domains of attraction of the steady-states of the constituent modes. Initially, the overall network was modeled as a switched nonlinear system that consisted of multiple modes, each governed by a set of continuous-time differential equations. The transitions between the continuous modes were triggered by discrete events (changes in model parameters that corresponded to alterations in physiological conditions). Then, following the characterization of the steady-state behavior of each mode, Lyapunov techniques

were used to characterize the domains of attraction of the steady-states. Finally, by analyzing how the domains of attraction of the various modes overlap with one other, a switching rule was derived to determine when, and if, a given mode transition at a given time results in the desired steady-state behavior. The proposed method was demonstrated using models of biological networks that arise in cell cycle regulation and the bacteriophage  $\lambda$ -switch system.

# Bibliography

- [1] E. Alm and A. P. Arkin. Biological networks. *Current Opinion in Structural Biology*, 13:193–202, 2003.
- [2] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. Pappas, H. Rubin, and J. Schug. Hybrid modeling and simulation of biomolecular networks. In *Lecture Notes in Computer Science Series*, volume 2034, pages 19–32, Di Benedetto, M. D. and A. Sangiovanni-Vincentelli (Eds.), Berlin, Germany: Springer-Verlag, 2001.
- [3] R. Alur, C. Belta, V. Kumar, M. Mintz, G. J. Pappas, H. Rubin, and J. Schug. Modeling and analyzing biomolecular networks. *Computing in Science and Engineering*, 4:20–31, 2002.
- [4] 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 Journal*, 50:2455–2466, 2004.
- [5] H. B. Aradhye, B. R. Bakshi, R. A. Strauss, and J. F. Davis. Multiscale SPC using wavelets: Theoretical analysis and properties. *AIChE Journal*, 49:939–958, 2003.
- [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. Bagajewicz and E. Cabrera. A new MILP formulation for instrumentation network design and upgrade. *AIChE Journal*, 48:2271–2282, 2002.
- [8] J. E. Bailey. Mathematical modeling and analysis in biochemical engineering: Past accomplishments and future opportunities. *Biotechnology Progress*, 14:8–20, 1998.
- [9] J. Bao, W. Z. Zhang, and P. L. Lee. Passivity-based decentralized failure-tolerant control. *Industrial & Engineering Chemistry Research*, 41:5702–5715, 2002.
- [10] J. Bao, W. Z. Zhang, and P. L. Lee. Decentralized fault-tolerant control system design for unstable processes. *Chemical Engineering Science*, 58:5045–5054, 2003.
- [11] S. Barthe and R. W. Rousseau. Utilization of focused beam reflectance measurement in the control of crystal size distribution in a batch cooled crystallizer. *Chemical Engineering & Technology*, 29:206–211, 2006.
- [12] P. I. Barton and C. C. Pantelides. Modeling of combined discrete/continuous processes. *AIChE Journal*, 40:966–979, 1994.
- [13] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35:407–427, 1999.
- [14] B. W. Bequette. Nonlinear control of chemical processes – a review. *Industrial & Engineering Chemistry Research*, 30:1391–1413, 1991.
- [15] B. W. Bequette. Nonlinear predictive control using multirate sampling. *Canadian Journal of Chemical Engineering*, 69:136–143, 1991.
- [16] V. Bhamidi, S. Varanasi, and C. A. Schall. Measurement and modeling of protein crystal nucleation kinetics. *Crystal Growth & Design*, 2:395–400, 2002.
- [17] S. P. Bhat and D. S. Bernstein. Finite-time stability of continuous autonomous systems. *SIAM Journal on Control and Optimization*, 38:751–766, 2000.



- [18] M. Blanke, R. Izadi-Zamanabadi, S. A. Bogh, and C. P. Lunau. Fault-tolerant control systems – a holistic view. *Control Engineering Practice*, 5:693–702, 1997.
- [19] C. F. Bohren and D. R. Huffman. *Absorption and Scattering of Light by Small Particles*. Wiley, New York, 1983.
- [20] R. D. Braatz and S. Hasebe. Particle size and shape control in crystallization processes. In *AIChE Symposium Series: Proceedings of 6th International Conference on Chemical Process Control*, pages 307–327, Rawlings, J. B. *et al.* (Eds.), Berlin, Germany: Springer-Verlag, 2002.
- [21] M. S. Branicky and S. K. Mitter. Algorithms for optimal hybrid control. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 2661–2666, New Orleans, LA, 1995.
- [22] R. Brockett. Minimum attention control. In *Proceedings of the 36th Conference on Decision and Control*, pages 2628–2632, San Diego, CA, 1997.
- [23] J. G. Chen, J. A. Bandoni, and J. A. Romagnoli. Robust PCA and normal region in multivariate statistical process monitoring. *AIChE Journal*, 42:3563–3566, 1996.
- [24] T. Chiu and P. D. Christofides. Nonlinear control of particulate processes. *AIChE Journal*, 45:1279–1297, 1999.
- [25] T. Chiu and P. D. Christofides. Robust control of particulate processes using uncertain population balances. *AIChE Journal*, 46:266–280, 2000.
- [26] K.-Y. Choi and W. H. Ray. The dynamic behavior of fluidized-bed reactors for solid catalyzed gas-phase olefin polymerization. *Chemical Engineering Science*, 40:2261–2279, 1985.
- [27] P. D. Christofides. Robust output feedback control of nonlinear singularly perturbed systems. *Automatica*, 36:45–52, 2000.

- [28] P. D. Christofides. *Model-Based Control of Particulate Processes*. Kluwer Academic Publishers, The Netherlands, 2002.
- [29] P. D. Christofides and N. H. El-Farra. *Control of Nonlinear and Hybrid Process Systems: Designs for Uncertainty, Constraints and Time-Delays*. Springer, New York, 2005.
- [30] P. D. Christofides, N. H. El-Farra, M. Li, and P. Mhaskar. Model-based control of particulate processes. *Chemical Engineering Science*, submitted, 2007.
- [31] P. D. Christofides and A. R. Teel. Singular perturbations and input-to-state stability. *IEEE Transactions on Automatic Control*, 41:1645–1650, 1996.
- [32] S. A. Dadebo, M. L. Bell, P. J. McLellan, and K. B. McAuley. Temperature control of industrial gas phase polyethylene reactors. *Journal of Process Control*, 7:83–95, 1997.
- [33] P. Daoutidis and M. Henson. Dynamics and control of cell populations. In *AIChE Symposium Series: Proceedings of 6th International Conference on Chemical Process Control*, pages 274–289, Rawlings, J. B. *et al.* (Eds.), Berlin, Germany: Springer-Verlag, 2002.
- [34] M. S. Dasika, A. Gupta, and C. D. Maranas. A mixed integer linear programming (MILP) framework for inferring time delay in gene regulatory networks. In *Proceedings of Pacific Symposium on Biocomputing*, pages 474–485, Hawaii, HI, 2004.
- [35] J. F. Davis, M. L. Piovoso, K. Kosanovich, and B. R. Bakshi. Process data analysis and interpretation. *Advances in Chemical Engineering*, 25:1–103, 1999.
- [36] H. De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103, 2002.
- [37] C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46:853–865, 2001.

- [38] C. De Persis and A. Isidori. On the design of fault detection filters with game-theoretic-optimal sensitivity. *International Journal of Robust & Nonlinear Control*, 12:729–747, 2002.
- [39] R. A. DeCarlo, M. S. Branicky, S. Petterson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88:1069–1082, 2000.
- [40] M. A. Demetriou. A model-based fault detection and diagnosis scheme for distributed parameter systems: A learning systems approach. *ESAIM-Control Optimisation and Calculus of Variations*, 7:43–67, 2002.
- [41] F. J. Doyle, M. Soroush, and C. Cordeiro. Control of product quality in polymerization processes. In *AIChE Symposium Series: Proceedings of 6th International Conference on Chemical Process Control*, pages 290–306, Rawlings, J. B. *et al.* (Eds.), Berlin, Germany: Springer-Verlag, 2002.
- [42] S. Djuljevic and N. Kazantzis. A new Lyapunov design approach for nonlinear systems based on Zubov’s method. *Automatica*, 38:1999–2007, 2002.
- [43] R. Dunia and S. J. Qin. Subspace approach to multidimensional fault identification and reconstruction. *AIChE Journal*, 44:1813–1831, 1998.
- [44] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42:2797–2812, 1996.
- [45] N. H. El-Farra, T. Chiu, and P. D. Christofides. Analysis and control of particulate processes with input constraints. *AIChE Journal*, 47:1849–1865, 2001.
- [46] N. H. El-Farra and P. D. Christofides. Integrating robustness, optimality and constraints in control of nonlinear processes. *Chemical Engineering Science*, 56:1841–1868, 2001.
- [47] 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*

*Series*, volume 2289, pages 164–178, Tomlin, C. J. and M. R. Greenstreet (Eds.), Berlin, Germany: Springer-Verlag, 2002.

- [48] N. H. El-Farra and P. D. Christofides. Bounded robust control of constrained multivariable nonlinear processes. *Chemical Engineering Science*, 58:3025–3047, 2003.
- [49] N. H. El-Farra and P. D. Christofides. Coordinating feedback and switching for control of hybrid nonlinear processes. *AIChE Journal*, 49:2079–2098, 2003.
- [50] N. H. El-Farra and P. D. Christofides. Coordinated feedback and switching for control of spatially-distributed processes. *Computers & Chemical Engineering*, 28:111–128, 2004.
- [51] N. H. El-Farra, A. Gani, and P. D. Christofides. Analysis of mode transitions in biological networks. *AIChE Journal*, 51:2220–2234, 2005.
- [52] N. H. El-Farra, A. Gani, and P. D. Christofides. Fault-tolerant control of process systems using communication networks. *AIChE Journal*, 51:1665–1682, 2005.
- [53] N. H. El-Farra, Y. Lou, and P. D. Christofides. Fault-tolerant control of fluid dynamic systems via coordinating feedback and switching. *Computers & Chemical Engineering*, 27:1913–1924, 2003.
- [54] 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*, 14:199–225, 2004.
- [55] 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.
- [56] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Output feedback control of switched nonlinear systems using multiple Lyapunov functions. *Systems & Control Letters*, 54:1163–1182, 2005.

- [57] D. Endy and R. Brent. Modelling cellular behavior. *Nature*, 409:391–395, 2001.
- [58] X.-J. Feng and H. Rabitz. Optimal identification of biochemical reaction networks. *Biophysical Journal*, 86:1270–1281, 2004.
- [59] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss. State and output feedback nonlinear model predictive control: An overview. *European Journal of Control*, 9:190–206, 2003.
- [60] 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.
- [61] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7:403–424, 1997.
- [62] R. A. Freeman and P. V. Kokotovic. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Birkhauser, Boston, 1996.
- [63] O. Galkin and P. G. Vekilov. Direct determination of the nucleation rates of protein crystals. *Journal of Physical Chemistry B*, 103:10965–10971, 1999.
- [64] A. Gani, P. Mhaskar, and P. D. Christofides. Fault-tolerant control of a polyethylene reactor. *Journal of Process Control*, 17:439–451, 2007.
- [65] A. Gani, P. Mhaskar, and P. D. Christofides. Handling sensor malfunctions in control of particulate processes. *Chemical Engineering Science*, in press, 62, 2007.
- [66] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control - theory and practice - a survey. *Automatica*, 25:335–348, 1989.
- [67] E. A. Garcia and P. M. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice*, 5:663–670, 1997.

- [68] V. Garcia-Osorio and B. E. Ydstie. Distributed, asynchronous and hybrid simulation of process networks using recording controllers. *International Journal of Robust & Nonlinear Control*, 14:227–248, 2004.
- [69] R. Ghosh and C. J. Tomlin. Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model. In *Lecture Notes in Computer Science Series*, volume 2034, pages 232–246, Di Benedetto, M. D. and A. Sangiovanni-Vincentelli (Eds.), Berlin, Germany: Springer-Verlag, 2001.
- [70] C. Giersch. Mathematical modelling of metabolism. *Current Opinion in Plant Biology*, 3:249–253, 2000.
- [71] J. W. Grizzle and P. V. Kokotovic. Feedback linearization of sampled-data systems. *IEEE Transactions on Automatic Control*, 33:857–859, 1988.
- [72] I. E. Grossmann, S. A. van den Heever, and I. Harjukoski. 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.
- [73] I. Harjunkoski, V. Jain, and I. E. Grossmann. Hybrid mixed-integer/constrained logic programming strategies for solving scheduling and combinatorial optimization problems. *Computers & Chemical Engineering*, 24:337–343, 2000.
- [74] T. J. Harris, F. Boudreau, and J. F. MacGregor. Performance assessment of multivariable feedback controllers. *Automatica*, 32:1505–1518, 1996.
- [75] A. Hassibi, S. P. Boyd, and J. P. How. Control of asynchronous dynamical systems with rate constraints on events. In *Proceedings of 38th IEEE Conference on Decision and Control*, pages 1345–1351, Phoenix, AZ, 1999.
- [76] J. Hasty, F. Isaacs, M. Dolnik, D. McMillen, and J. J. Collins. Designer gene networks: Towards fundamental cellular control. *Chaos*, 11:207–220, 2001.

- [77] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins. Computational studies of gene regulatory networks: In numero molecular biology. *Nature Reviews Genetics*, 2:268–279, 2001.
- [78] V. Hatzimanikatis, K. H. Lee, and J. E. Bailey. A mathematical description of regulation of the G1-S transition of the mammalian cell cycle. *Biotechnology and Bioengineering*, 65:631–637, 1999.
- [79] M. A. Henson and D. E. Seborg. *Nonlinear Process Control*. Prentice-Hall, Englewood Cliffs, NJ, 1997.
- [80] J. 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.
- [81] H. M. Hulburt and S. Katz. Some problems in particle technology - a statistical mechanical formulation. *Chemical Engineering Science*, 19:555–574, 1964.
- [82] I. Hyaneek, J. Zacca, F. Teymour, and W. H. Ray. Dynamics and stability of polymerization process flow sheets. *Industrial & Engineering Chemistry Research*, 34:3872–3877, 1995.
- [83] G. R. Jerauld, Y. Vasatis, and M. F. Doherty. Simple conditions for the appearance of sustained oscillations in continuous crystallizers. *Chemical Engineering Science*, 38:1675–1681, 1983.
- [84] A. Kalani and P. D. Christofides. Nonlinear control of spatially-inhomogeneous aerosol processes. *Chemical Engineering Science*, 54:2669–2678, 1999.
- [85] N. Kapoor and P. Daoutidis. Stabilization of systems with input constraints. *International Journal of Control*, 66:653–675, 1997.
- [86] N. Kapoor and P. Daoutidis. Stabilization of nonlinear processes with input constraints. *Computers & Chemical Engineering*, 24:9–21, 2000.

- [87] N. Kazantzis. A functional equations approach to nonlinear discrete-time feedback stabilization through pole-placement. *Systems & Control Letters*, 43:361–369, 2001.
- [88] N. Kazantzis and C. Kravaris. Energy-predictive control: a new synthesis approach for nonlinear process control. *Chemical Engineering Science*, 54:1697–1709, 1999.
- [89] N. Kazantzis and C. Kravaris. Nonlinear observer design using Lyapunov’s auxiliary theorem. *Systems & Control Letters*, 34:241–247, 1999.
- [90] N. Kazantzis, C. Kravaris, and R. A. Wright. Nonlinear observer design for process monitoring. *Industrial & Engineering Chemistry Research*, 39:408–419, 2000.
- [91] H. K. Khalil. *Nonlinear Systems*. Macmillan Publishing Company, New York, NY, second edition, 1996.
- [92] P. Kokotovic and M. Arcak. Constructive nonlinear control: a historical perspective. *Automatica*, 37:637–662, 2001.
- [93] S. L. D. Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45:1053–1071, 2000.
- [94] X. D. Koutsoukos, P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon. Supervisory control of hybrid systems. *Proceedings of the IEEE*, 88:1026–1049, 2000.
- [95] C. Kravaris and Y. Arkun. Geometric nonlinear control - an overview. In *Proceedings of 4th International Conference on Chemical Process Control*, pages 477–515, Y. Arkun and W. H. Ray Eds., Padre Island, TX, 1991.
- [96] J. V. Kresta, J. F. Macgregor, and T. E. Marlin. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69:35–47, 1991.



- [97] N. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design*. Wiley, New York, first edition, 1995.
- [98] S. J. Lei, R. Shinnar, and S. Katz. The stability and dynamic behavior of a continuous crystallizer with a fines trap. *AIChE Journal*, 17:1459–1470, 1971.
- [99] A. K. W. Leung, M. M. V. Park, and D. W. Borhani. An improved method for protein crystal density measurements. *Journal of Applied Crystallography*, 32:1006–1009, 1999.
- [100] B. Lewin. *Genes VII*. Oxford University Press, Cambridge, UK, 2000.
- [101] F.-L. Lian. *Analysis, Design, Modeling, and Control of Networked Control Systems*. PhD thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 2001.
- [102] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19:59–70, 1999.
- [103] Y. Lin and E. D. Sontag. A universal formula for stabilization with bounded controls. *Systems & Control Letters*, 16:393–397, 1991.
- [104] D. L. Ma, D. K. Tafti, and R. D. Braatz. Optimal control and simulation of multidimensional crystallization processes. *Computers & Chemical Engineering*, 26:1103–1116, 2002.
- [105] M. Mahmoud, J. Jiang, and Y. Zhang. Active fault tolerant control systems: Stochastic analysis and synthesis. In *Lecture Notes in Control and Information Sciences*, volume 287, pages 1–187, Heidelberg, Germany: Springer-Verlag, 2003.
- [106] N. A. Mahmoud and H. K. Khalil. Asymptotic regulation of minimum phase nonlinear systems using output feedback. *IEEE Transactions on Automatic Control*, 41:1402–1412, 1996.

- [107] N. V. Mantzaris and P. Daoutidis. Cell population balance modeling and control in continuous bioreactors. *Journal of Process Control*, 14:775–784, 2004.
- [108] M. Massoumnia, G. C. Verghese, and A. S. Wilsky. Failure detection and identification. *IEEE Transactions on Automatic Control*, 34:316–321, 1989.
- [109] 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.
- [110] H. H. McAdams and A. Arkin. Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*, 27:199–224, 1998.
- [111] K. B. McAuley, D. A. Macdonald, and P. J. McLellan. Effects of operating conditions on stability of gas-phase polyethylene reactors. *AIChE Journal*, 41:868–879, 1995.
- [112] N. Mehranbod, M. Soroush, and C. Panjapornpon. A method of sensor fault detection and identification. *Journal of Process Control*, 15:321–339, 2005.
- [113] P. Mendes and D. Kell. Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14:869–883, 1998.
- [114] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Hybrid predictive control of process systems. *AIChE Journal*, 50:1242–1259, 2004.
- [115] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50:1670–1680, 2005.
- [116] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Robust hybrid predictive control of nonlinear systems. *Automatica*, 41:209–217, 2005.

- [117] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Systems & Control Letters*, 55:650–659, 2006.
- [118] P. Mhaskar, A. Gani, and P. D. Christofides. Fault-tolerant control of nonlinear processes: Performance-based reconfiguration and robustness. *International Journal of Robust & Nonlinear Control*, 16:91–111, 2006.
- [119] P. Mhaskar, A. Gani, N. H. El-Farra, C. McFall, P. D. Christofides, and J. F. Davis. Integrated fault-detection and fault-tolerant control for process systems. *AIChE Journal*, 52:2129–2148, 2006.
- [120] P. Mhaskar, A. Gani, C. McFall, P. D. Christofides, and J. F. Davis. Fault-tolerant control of nonlinear process systems subject to sensor faults. *AIChE Journal*, 53:654–668, 2007.
- [121] P. Mhaskar, C. McFall, A. Gani, P. D. Christofides, and J. F. Davis. Fault-tolerant control of nonlinear systems: Fault-detection and isolation and controller reconfiguration. *Automatica*, submitted, 2006.
- [122] P. Mhaskar, C. McFall, A. Gani, P. D. Christofides, and J. F. Davis. Fault-tolerant control of nonlinear systems: Fault detection and isolation and controller reconfiguration. In *Proceedings of the American Control Conference*, pages 5115–5122, Minneapolis, Minnesota, 2006.
- [123] H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40:995–1006, 1995.
- [124] L. A. Montestruque and P. J. Antsaklis. On the model-based control of networked systems. *Automatica*, 39:1837–1843, 2003.
- [125] E. Musulin, M. Bagajewicz, J. M. Nougues, and L. Puigjaner. Instrumentation design and upgrade for principal components analysis monitoring. *Industrial & Engineering Chemistry Research*, 43:2150–2159, 2004.

- [126] A. Negiz and A. Cinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal*, 43:2002–2020, 1997.
- [127] P. R. C. Nelson, P. A. Taylor, and J. F. MacGregor. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and Intelligent Laboratory Systems*, 35:45–65, 1996.
- [128] D. Nesic, A. R. Teel, and P. V. Kokotovic. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters*, 38:259–270, 1999.
- [129] H. Niemann, A. Saberi., A. A. Stoorvogel, and P. Sannuti. Exact, almost and delayed fault detection: An observer based approach. *International Journal of Robust & Nonlinear Control*, 9:215–238, 1999.
- [130] J. Nilsson. *Real-Time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.
- [131] P. Nomikos and J. F. Macgregor. Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40:1361–1375, 1994.
- [132] M. N. Nounou, B. R. Bakshi, P. K. Goel, and X. Shen. Bayesian principal component analysis. *Journal of Chemometrics*, 16:576–595, 2002.
- [133] B. Novak and J. J. Tyson. Modeling the cell division cycle: M-phase trigger, oscillations and size control. *Journal Theoretical Biology*, 165:101–134, 1993.
- [134] A. Papachristodoulou, S. Prajna, and J. C. Doyle. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proceedings of 41st IEEE Conference on Decision and Control*, pages 3482–3487, Las Vegas, NV, 2002.
- [135] R. Patankar. A model for fault-tolerant networked control system using TTP/C communication. In *Proceedings of the American Control Conference*, pages 533–537, Denver, CO, 2003.

- [136] 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.
- [137] J. A. Primbs, V. Nevistic, and J. C. Doyle. A receding horizon generalization of pointwise min-norm controllers. *IEEE Transactions on Automatic Control*, 45:898–909, 2000.
- [138] M. Ptashne. *A Genetic Switch: Gene Control and Phage  $\lambda$* . Cell Press, Cambridge, MA, 1986.
- [139] R. S. Raji. Smart networks for control. *IEEE Spectrum*, 31:49–55, 1994.
- [140] D. Ramkrishna. The status of population balances. *Reviews in Chemical Engineering*, 3:49–95, 1985.
- [141] C. V. Rao and J. B. Rawlings. Constrained process monitoring: Moving-horizon approach. *AIChE Journal*, 48:97–109, 2002.
- [142] J. B. Rawlings, S. M. Miller, and W. R. Witkowski. Model identification and control of solution crystallization processes – a review. *Industrial & Engineering Chemistry Research*, 32:1275–1296, 1993.
- [143] A. Ray and Y. Halevi. Integrated communication and control systems: Part ii – design considerations. *ASME Journal of Dynamic Systems, Measurement, and Control*, 110:374–381, 1988.
- [144] D. K. Rollins and J. F. Davis. Gross error-detection when variance-covariance matrices are unknown. *AIChE Journal*, 39:1335–1341, 1993.
- [145] D. R. Rollins and J. F. Davis. An unbiased estimation technique when gross errors exist in process measurements. *AIChE Journal*, 38:563–572, 1992.
- [146] A. Saberi, A. A. Stoorvogel, P. Sannuti, and H. Niemann. Fundamental problems in fault detection and identification. *International Journal of Robust & Nonlinear Control*, 10:1209–1236, 2000.

- [147] M. Santillán and M. C. Mackey. Why the lysogenic state of phage  $\lambda$  is so stable: A mathematical modeling approach. *Biophysical Journal*, 86:75–84, 2004.
- [148] 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. *Journal of Crystal Growth*, 169:355–360, 1996.
- [149] R. Sepulchre, M. Jankovic, and P. V. Kokotovic. *Constructive Nonlinear Control*. Springer-Verlag, Berlin, 1997.
- [150] J. Sheng, T. W. Chen, and S. L. Shah. Optimal filtering for multirate systems. *IEEE Transactions on Circuits and Systems II - Express Briefs*, 52:228–232, 2005.
- [151] D. Shi, N. H. El-Farra, M. Li, P. Mhaskar, and P. D. Christofides. Predictive control of particle size distribution in particulate processes. *Chemical Engineering Science*, 61:268–281, 2006.
- [152] D. Shi, P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of crystal size distribution in protein crystallization. *Nanotechnology*, 16:S562–S574, 2005.
- [153] P. Smolen, D. A. Baxter, and J. H. Byrne. Modeling transcriptional control in gene networks: Methods, recent results, and future directions. *Bulletin of Mathematical Biology*, 62:247–292, 2000.
- [154] M. Soroush and N. Zambare. Nonlinear output feedback control of a class of polymerization reactors. *IEEE Transactions on Control Systems Technology*, 8:310–320, 2000.
- [155] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall, New Jersey, 1996.
- [156] E. Tatara and A. Cinar. An intelligent system for multivariate statistical process monitoring and diagnosis. *ISA Transactions*, 41:255–270, 2002.

- [157] S. Tatiraju, M. Soroush, and B. A. Ogunnaike. Multirate nonlinear state estimation with application to a polymerization reactor. *AIChE Journal*, 45:769–780, 1999.
- [158] A. R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems & Control Letters*, 18:165–171, 1992.
- [159] Y. Tipsuwan and M.-Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11:1099–1111, 2003.
- [160] M. Turkay and I. E. Grossmann. Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering*, 20:959–978, 1996.
- [161] J. J. Tyson, K. Chen, and B. Novak. Network dynamics and cell physiology. *Nature Reviews Molecular Cell Biology*, 2:908–916, 2001.
- [162] J. J. Tyson, A. Csikasz-Nagy, and B. Novak. The dynamics of cell cycle regulation. *BioEssays*, 24:1095–1109, 2002.
- [163] S. Valluri and M. Soroush. A non-linear controller design method for processes with saturating actuators. *International Journal of Control*, 76:698–716, 2003.
- [164] S. Valluri, M. Soroush, and M. Nikraves. Shortest-prediction-horizon nonlinear model-predictive control. *Chemical Engineering Science*, 53:273–292, 1998.
- [165] A. Vecchietti and I. E. Grossmann. LOGMIP: A disjunctive 0-1 nonlinear optimizer for process systems models. *Computers & Chemical Engineering*, 21:S427–S432, 1997.
- [166] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri. A review of process fault detection and diagnosis part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27:313–326, 2003.

- [167] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis part III: Process history based methods. *Computers & Chemical Engineering*, 27:327–346, 2003.
- [168] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27:293–311, 2003.
- [169] G. C. Walsh, H. Ye, and L. G. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10:438–446, 2002.
- [170] J. R. Whiteley and J. F. Davis. Qualitative interpretation of sensor patterns. *IEEE Expert*, 8:54–63, 1992.
- [171] H. S. Wiley, S. Y. Shvartsman, and D. A. Lauffenburger. Computational modeling of the EGF-receptor system: A paradigm for systems biology. *Trends in Cell Biology*, 13:43–50, 2003.
- [172] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1998.
- [173] N. E. Wu. Coverage in fault-tolerant control. *Automatica*, 40:537–548, 2004.
- [174] T. Y. Xie, K. B. McAuley, J. C. C. Hsu, and D. W. Bacon. Gas-phase ethylene polymerization – production processes, polymer properties, and reactor modeling. *Industrial & Engineering Chemistry Research*, 33:449–479, 1994.
- [175] W. Xie, S. Rohani, and A. Phoenix. Dynamic modeling and operation of a seeded batch cooling crystallizer. *Chemical Engineering Communications*, 187:229–249, 2001.
- [176] Y. Xu and J. Hespanha. Communication logics for networked control systems. In *Proceedings of the American Control Conference*, pages 572–577, Boston, MA, 2004.



- [177] E. C. Yamalidou and J. Kantor. Modeling and optimal control of discrete-event chemical processes using petri nets. *Computers & Chemical Engineering*, 15:503–519, 1990.
- [178] G. H. Yang, J. L. Wang, and Y. C. Soh. Reliable  $H_\infty$  control design for linear systems. *Automatica*, 37:717–725, 2001.
- [179] G. H. Yang, S. Y. Zhang, J. Lam, and J. Wang. Reliable control using redundant controllers. *IEEE Transactions on Automatic Control*, 43:1588–1593, 1998.
- [180] B. E. 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.
- [181] B. E. Ydstie. New vistas for process control: Integrating physics and communication networks. *AIChE Journal*, 48:422–426, 2002.
- [182] L. Zaccarian, A. R. Teel, and D. Nesic. On finite gain  $L_P$  stability of nonlinear sampled-data systems. *Systems & Control Letters*, 49:201–212, 2003.
- [183] G. P. Zhang and S. Rohani. On-line optimal control of a seeded batch cooling crystallizer. *Chemical Engineering Science*, 58:1887–1896, 2003.
- [184] P. Zhang and C. G. Cassandras. An improved forward algorithm for optimal control of a class of hybrid systems. *IEEE Transactions on Automatic Control*, 47:1735–1739, 2002.
- [185] W. Zhang. *Stability Analysis of Networked Control Systems*. PhD thesis, Department of Electrical Engineering & Computer Science, Case Western Reserve University, Cleveland, OH, 2001.
- [186] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21:84–99, 2001.

- [187] X. D. Zhang, T. Parisini, and M. M. Polycarpou. Adaptive fault-tolerant control of nonlinear uncertain systems: An information-based diagnostic approach. *IEEE Transactions on Automatic Control*, 49:1259–1274, 2004.
- [188] J. W. Zwolak, J. J. Tyson, and L. T. Watson. Finding all steady state solutions of chemical kinetic models. *Nonlinear Analysis–Real World Applications*, 5:801–814, 2004.