

UNIVERSITY OF CALIFORNIA
Los Angeles

Data-Based Monitoring and Fault-Tolerant Control of Nonlinear Processes

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

by

David Chilin

2012

ABSTRACT OF THE DISSERTATION

Data-Based Monitoring and Fault-Tolerant Control of Nonlinear Processes

by

David Chilin

Doctor of Philosophy in Chemical Engineering

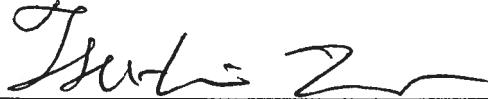
University of California, Los Angeles, 2012

Professor Panagiotis D. Christofides, Chair

Fault-tolerant control is an essential component in modern process industries as abnormal situations account for over \$20 billion in lost annual revenue in the US alone. Traditionally, control systems rely on centralized control architectures utilizing dedicated wired links to measurement sensors and control actuators to operate a plant at desired conditions and a separate monitoring system for detecting faults. While this paradigm to process operations and control has been successful, modern chemical plants that rely on highly automated processes to maintain robust operations and efficient production are vulnerable to abnormal situations like, for example, actuator faults. Loss of control in a chemical process can lead to the waste of raw materials and energy resources, as well as downtime and production losses but most importantly it may lead to personnel injury or death and/or environmental hazard. This issue has prompted significant research efforts in the integration and application of fault-tolerant control to existing legacy control systems. This dissertation will present a paradigm shift to the existing approach of designing control systems and monitoring systems in that it proposes to design distributed control systems that are stabilizing, robust and optimal, and whose design leads to closed-loop system structures that facilitate fault isolation with the flexibility to not only avert disaster in the case of an abnormal situation but maintain optimal plant operation. To present our method of fault-tolerant control, we will focus on a broad class of non-linear process systems subject to disturbances and persistent control actuator faults. In general terms, the method includes the design of distributed model predictive

control laws combined with a fault-detection and isolation approach based on process models and fault-free data that leads to successful detection and isolation of an actuator fault. After isolation of an actuator fault, the fault-tolerant control system estimates the fault magnitude, calculates a new optimal operating point, and ultimately reconfigures the distributed model predictive control system to maintain stability of the process in an optimal manner. Throughout the thesis, detailed examples of large-scale chemical process systems are used to demonstrate the approach.

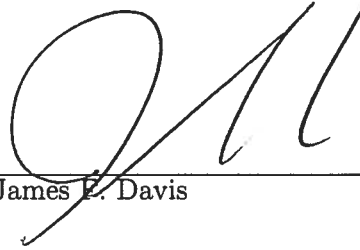
The dissertation of David Chilin is approved.



Tsu-Chin Tsao



Gerassimos Orkoulas



James E. Davis



Panagiotis D. Christofides, Committee Chair

University of California, Los Angeles

2012

Acknowledgements

Chapter 2 is a version of D. Chilin, J. Liu, D. Muñoz de la Peña, P. D. Christofides and J. F. Davis, “Detection, Isolation and Handling of Actuator Faults in Distributed Model Predictive Control Systems,” *J. Proc. Contr.*, 20, 1059-1075, 2010.

Chapter 3 is a version of D. Chilin, J. Liu, J. F. Davis and P. D. Christofides, “Data-Based Monitoring and Reconfiguration of a Distributed Model Predictive Control System,” *Int. J. Rob. & Non. Contr.*, 22, 68-88, 2012.

Chapter 4 is a version of D. Chilin, X. Chen, J. Liu, P. D. Christofides, “Fault Detection and Isolation and Fault Tolerant Control of a Catalytic Alkylation of Benzene Process,” *Chem. Eng. Sci.*, submitted, 2012.

Chapter 5 is a version of A. Leosirikul, D. Chilin, J. Liu, J. F. Davis and P. D. Christofides, “Monitoring and Retuning of Low-Level PID Control Loops,” *Chem. Eng. Sci.*, 69, 287-295, 2012.

I am grateful to my advisors, Professor Panagiotis D. Christofides and Professor James F. Davis, for their guidance and support.

Y les doy las gracias a mis padres por todo el apoyo que me han dado.

Contents

1	Introduction	1
1.1	Background	1
1.2	Thesis objectives and structure	4
2	Detection, Isolation and Handling of Actuator Faults in Distributed Model Predictive Control Systems	5
2.1	Introduction	5
2.2	Problem formulation and preliminaries	6
2.2.1	Class of nonlinear systems	6
2.2.2	Lyapunov-based controller	7
2.2.3	DMPC design for fault-free system	8
2.2.4	FTC considerations and backup DMPC design	10
2.3	FDI and FTC strategies	13
2.3.1	FDI system design	14
2.3.2	FTC system design	15
2.4	Application to a reactor-separator process	19
2.4.1	Process description and modeling	19
2.4.2	Simulation Results	25
2.5	Conclusions	36

3	Data-Based Fault Detection and Isolation Using Adaptive Isolation Windows	37
3.1	Introduction	37
3.2	Notation	38
3.3	Problem formulation and preliminaries	38
3.3.1	Class of nonlinear systems	38
3.3.2	Fault-free control system design	39
3.3.3	FTC considerations	41
3.4	FDI and FTC system design	43
3.4.1	Design of fault detection filters and residuals	43
3.4.2	Fault detection and isolation using adaptive windows	45
3.4.3	Fault parameter estimation	47
3.4.4	FTC strategies	48
3.5	Application to a reactor-separator process	51
3.5.1	Process description and modeling	51
3.5.2	Simulation Results	53
3.6	Conclusions	64
4	Fault Detection and Isolation and Fault Tolerant Control of a Catalytic Alkylation of Benzene Process	65
4.1	Introduction	65
4.2	Description of the alkylation of benzene process	66
4.3	FDIFTC system design	75
4.3.1	Fault-free DMPC system design	75
4.3.2	Fault Detection and Isolation	77
4.3.3	Fault parameter estimation	79
4.3.4	FTC consideration and strategies	80
4.4	Simulation results	85

4.4.1	No Fault Tolerant Control Implementation	86
4.4.2	FTC of a Q_3 heat actuator fault	89
4.4.3	FTC of an F_4 flow actuator fault	91
4.5	Conclusion	101
5	Monitoring and Retuning of Low-Level PID Control Loops	102
5.1	Introduction	102
5.2	Preliminaries	104
5.2.1	Class of Nonlinear Systems	104
5.2.2	Lyapunov-based MPC	105
5.2.3	Low-level PID Loops	107
5.3	Monitoring and Retuning of Low-Level PID Loops	108
5.4	Application to a Nonlinear Chemical Process Network	111
5.5	Conclusion	116
6	Conclusions	117
	Bibliography	118

List of Figures

2.1	Proposed FDI and FTC structure for DMPC.	13
2.2	Two CSTRs and a flash tank with recycle stream.	20
2.3	Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. No FTC is implemented.	26
2.4	Concentration profiles ($C_A = \times$, $C_B = \circ$, $C_C = \diamond$) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. No FTC is implemented.	26
2.5	FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. No FTC is implemented.	27
2.6	Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15a is implemented.	28
2.7	Concentration profiles ($C_A = \times$, $C_B = \circ$, $C_C = \diamond$) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15a is implemented, but cannot stabilize T_2 and T_3 to the desired steady-state.	28
2.8	Control actions with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15a is implemented.	29

2.9	Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15 is implemented.	30
2.10	Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15 is implemented.	30
2.11	FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15 is implemented.	31
2.12	Control actions with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15 is implemented.	31
2.13	Temperature profiles for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.204$ hr and isolated at $t = 0.219$ hr. No FTC is implemented.	32
2.14	Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.204$ hr and isolated at $t = 0.219$ hr. No FTC is implemented.	33
2.15	FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.204$ hr and isolated at $t = 0.219$ hr. No FTC is implemented.	33
2.16	Temperature profiles for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.204$ hr and isolated at $t = 0.219$ hr. The FTC switching rule of Eq. 2.14 is implemented.	34
2.17	Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.204$ hr and isolated at $t = 0.219$ hr. The FTC switching rule of Eq. 2.14 is implemented.	34

2.18	FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the inlet flow actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.204 \text{ hr}$ and isolated at $t = 0.219 \text{ hr}$. The FTC switching rule of Eq. 2.14 is implemented.	35
3.1	An example of residual evolution after the occurrence of a fault at t_f that affects residual $r_{E,p}$	49
3.2	Two CSTRs and a flash tank with recycle stream.	51
3.3	Case 1: Q_2 “small” fault is isolated using longer waiting time calculated from the residual change of T_2 . r_{E,T_2} (top left plot) exceeds $\sigma_{T_{2,3}}$ at 1.065 hr and then exceeds $\sigma_{T_{2,4}}$ where the fault isolation time is set to 3.6 min . When r_{E,T_2} further exceeds $\sigma_{T_{2,5}}$ at 1.080 hr , the waiting time is updated to 36 sec . The fault is isolated and estimated as 18.5 KJ/hr (actual 20.0 KJ/hr) at 1.090 hr	54
3.4	Case 1: Q_2 “small” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Concentrations. Note the new steady state values and scale.	54
3.5	Case 1: Q_2 “small” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Temperatures.	55
3.6	Case 1: Q_2 “small” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Control actions.	56
3.7	Case 2: Q_2 “large” fault is isolated using a shorter waiting time based on residual change of T_2 . r_{E,T_2} (top left plot) immediately exceeds $\sigma_{T_{2,5}}$ in a single measurement update at 1.060 hr . The calculated waiting time is 36 sec . The fault is estimated as 88 KJ/hr (actual 80 KJ/hr) and FTC is implemented at 1.070 hr	56
3.8	Case 2: Q_2 “large” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Concentrations. Note the new steady state values and scale.	57
3.9	Q_2 “large” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Temperatures.	58
3.10	Q_2 “large” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Control actions.	58

3.11	Case 3: Same fault conditions as in case 2, i.e., Q_2 “large” fault, using fixed isolation time. r_{E,T_2} immediately exceeding $\sigma_{T_2,5}$ at 1.060 <i>hr</i> , but the isolation windows is set to worst case condition of 4.8 <i>min</i> . The fault is isolated at 1.135 <i>hr</i> and estimated as 89 <i>KJ/hr</i> (actual 80 <i>KJ/hr</i>) but the FTC is unable to stabilize the closed-loop system.	59
3.12	Case 3: Same fault conditions as in case 2, i.e., Q_2 “large” fault, using fixed isolation time. The control system is reconfigured at 1.135 <i>hr</i> and is unable to stabilize the closed-loop system - Concentrations.	60
3.13	Case 3: Same fault conditions as in case 2, i.e., Q_2 “large” fault, using a fixed isolation time. The control system is reconfigured at 1.135 <i>hr</i> and is unable to stabilize the closed-loop system - Temperatures.	60
3.14	Case 4: F_{20} fault demonstrates FDI system with multiple residuals exceeding the thresholds under small magnitude fault. The residual r_{T_2} first exceeds $\sigma_{T_2,3}$ at 1.100 <i>hr</i> and $r_{C_{B_2}}$ exceeds $\sigma_{C_{B_2,4}}$ at 1.105 <i>hr</i> . A fault is declared at 1.115 <i>hr</i> and an isolation time window of 4.8 <i>min</i> is calculated. A new fault isolation window of 36 <i>sec</i> is calculated when $r_{C_{B_2}}$ exceeds $\sigma_{C_{B_2,5}}$ at 1.120 <i>hr</i> and at the end of the new isolation window, no matching fault signature is found. The FDI system continues monitoring the residuals until 1.150 <i>hr</i> a matching fault signature is found. The FTC is implemented once the fault is isolated with a fault estimate 1.08 m^3/hr (actual 0.85 m^3/hr).	61
3.15	Case 4: F_{20} fault is isolated and control system is reconfigured to stabilize the closed-loop system - Concentrations.	61
3.16	Case 4: F_{20} fault is isolated and control system is reconfigured to stabilize the closed-loop system - Temperatures.	62
3.17	Case 4: F_{20} fault is isolated and control system is reconfigured to stabilize the closed-loop system - Control actions.	62
4.1	Process flow diagram of alkylation of benzene.	66
4.2	Sequential distributed LMPC for the catalytic alkylation of benzene process.	75

4.3	Sequential distributed LMPC with FDIFTC system.	79
4.4	Temperatures trajectories for the five vessels under normal fault free operation. Dotted line represents target operating point. The process reaches steady-state conditions around 200 <i>min</i>	87
4.5	Trajectories of ethylene concentration (<i>mole/m³</i>) for the five vessels under normal fault free operation. Dotted line represents target operating point. The process reaches steady-state after 200 <i>min</i>	88
4.6	Temperature trajectories of the five vessels after triggering a Q_3 fault at time 200 <i>min</i> with no fault tolerant control. Note that the vessel 3 disturbance eventually propagates downstream to vessel 4 and 5 after 310 <i>min</i> and 360 <i>min</i> , respectively.	89
4.7	Residual plots of key isolation residuals showing residual pattern upon triggering a fault in the heat actuator to the third vessel (Q_3) with no fault tolerant control. Note how only the residual associated with the temperature of vessel three (r_{T_3}) is severely affected.	90
4.8	Residual plots of key isolation residuals showing residual pattern upon triggering a fault in the heat actuator to the third vessel (Q_3) and using FTC at time 205 <i>min</i> . Note how residual r_{T_3} trajectory changes immediately after reconfiguration.	92
4.9	Temperature trajectories of the five vessels after triggering a Q_3 actuator fault at 200 <i>min</i> and achieving fault isolation at 205 <i>min</i> . The small peak above the threshold in T_3 from 200 – 210 <i>min</i> is the result of the actuator fault.	93
4.10	Residual plots after initiating a flow actuator fault in tank 2 (F_4) with no fault tolerant control. Note that this fault causes a shift in the residuals for tank 2 ethylene concentration at 200 <i>min</i>	95
4.11	Temperature trajectories of the five vessels after triggering a F_4 fault at time 200 <i>min</i> with no fault tolerant control.	96
4.12	Manipulated input trajectories after initiating a flow actuator fault in tank 2 (F_4) with no fault tolerant control. Final cost 6.7×10^7 <i>units</i> . Units of F_4 are <i>m³/s</i> and Q_1, \dots, Q_5 are <i>J/s</i> ; all inputs are scaled to be in the range of $[-1, 1]$ using the values of Table 4.4.	97

4.13	Manipulated input trajectories after initiating a flow actuator fault in tank 2 (F_4) with fault tolerant control. Final cost 5.3×10^7 units. Units of F_4 are m^3/s and Q_1, \dots, Q_5 are J/s ; all inputs are scaled to be in the range of $[-1, 1]$ using the values of Table 4.4.	98
4.14	Temperature trajectories of the five vessels after triggering a F_4 fault at time 200 min with FTC reconfiguration. Note smaller deviation peak in T_2 compared to no FTC implementation in fig. 4.11	99
4.15	Residual plots after initiating a flow actuator fault in tank 2 (F_4) with fault tolerant control.	100
5.1	Closed-loop system with MPC as advanced model-based controller and low-level PID controller implemented to regulate the control actuators.	103
5.2	Monitoring scheme of PID response behavior based on the EWMA residuals of the process state. Poor tuning is declared after $r_{E,i}$ exceeds its threshold $\Omega_{E,i}$ continuously for $t = t_d$	110
5.3	Requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is not implemented.	113
5.4	Temperature residuals for the 3 vessels computed via EWMA when PID retuning is not implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$	114
5.5	Requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is implemented.	114
5.6	Temperature residuals for the 3 vessels computed via EWMA when PID retuning is implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$	115

List of Tables

2.1	Process variables	22
2.2	Parameter values	23
3.1	The desired operating steady-state x_s	59
3.2	The steady-state input values.	62
3.3	EWMA Residual Means and Standard Deviations.	63
4.1	Process variables	72
4.2	Parameter values	73
4.3	Steady-state input values for x_s	73
4.4	Manipulated input constraints.	74
4.5	Fault signature shows which residuals are triggered by faults in particular actuators. Note that some signatures overlap (i.e., Q_2 fault signature overlaps with F_4 fault signature and Q_3 fault signature overlaps with F_6 fault signature)	77

VITA

- July 13, 1979 Born, Los Angeles, California
- 2001–2002 Process Specialist
Integrated Nanosystems Research Facility
Irvine, California
- 2002 Bachelor of Science, Chemical Engineering
Department of Chemical Engineering and Materials Science
Univeristy of California, Irvine
- 2003–2007 Process Engineer
Amgen, Inc
Thousand Oaks, California
- 2010 Master of Science, Chemical Engineering
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles
- 2008–2012 Graduate Student Researcher
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles
- 2010–2012 Teaching Assistant/Associate/Consultant
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles

PUBLICATIONS AND PRESENTATIONS

1. Chilin, D., J. Liu, J. F. Davis and P. D. Christofides, Data-Based Monitoring and Reconfiguration of Distributed Model Predictive Control, *Int. J. Rob. & Non. Contr.*, 22, 68-88, 2012.
2. Chilin, D., J. Liu, D. Muñoz de la Peña, J. F. Davis and P. D. Christofides, Detection, Isolation and Handling of Actuator Faults in Distributed Model Predictive Control Systems, *Journal of Process Control*, 20, 1059-1075, 2010.
3. Leosirikul, A., D. Chilin, J. Liu, J. F. Davis and P. D. Christofides, Monitoring and Retuning of Low-Level PID Control Loops, *Chem. Eng. Sci.*, 69, 287-295, 2012.

4. Rathore, A.S., A. Sharma, and D. Chilin, Applying Process Analytical Technology to Biotech Unit Operations, *Biopharm International*, August 2006.
5. Chilin, D., A. Wang, B. Dransart, T. Hong, F. Mori, and A. Rathore, Process characterization of a cation-exchange chromatography step used for purification of a therapeutic protein product, *Poster, 229th American Chemical Society National Meeting*, San Diego, CA , 2005.
6. Sharma, A., A.S. Rathore, Alice Wang, and David Chilin, Experiences with scale-down of unit operations used for purification of biotech products, *Poster 229th American Chemical Society National Meeting*, San Diego, CA, 2005.
7. Leosirikul, A., D. Chilin, J. Liu, J. F. Davis and P. D. Christofides, Monitoring of Low-Level PID Control Loops, *Proceedings of the American Control Conference*, in press, Montreal, Canada, 2012.
8. Chilin, D., J. Liu, D. Muñoz de la Peña, J. F. Davis and P. D. Christofides, Data-Based Monitoring and Reconfiguration of a Distributed Model Predictive Control System, *Proceedings of the American Control Conference*, 3158-3165, San Francisco, CA, 2011.
9. Chilin, D., J. Liu, D. Muñoz de la Peña, P. D. Christofides and J. F. Davis, Monitoring and Handling of Actuator Faults in a Distributed Model Predictive Control System, *Proceedings of the American Control Conference*, 2847-2854, Baltimore, Maryland, 2010.
10. Chilin, D., J. Liu, X. Chen and P. D. Christofides, Fault Detection and Isolation and Fault Tolerant Control of a Catalytic Alkylation of Benzene Process, *Chemical Engineering Science*, accepted, 2012.

Chapter 1

Introduction

1.1 Background

World markets are becoming increasingly competitive, such that manufacturers are driven to pursue every bit of performance gain from current operations in order to maintain competitiveness. In the pursuit of this ultimate performance, manufactures are increasingly relying on advanced process control systems. With advances constantly being made in computational capabilities, model predictive control (MPC) has emerged as a reasonable and potentially profitable solution to achieve optimal process operation and control. MPC lends itself well as an overlying layer that can be implemented on top of existing classical plant control systems and does well to handle input and state constraints. As the complexity of manufacturing plants has increased, cooperative, distributed MPC architectures have emerged that also deal well with plant modernization that may include sensor and actuator networks that may be implemented using wireless or wired networks. One of the largest pitfalls for closed-loop process performance are abnormal situations which account for at least \$20 billion in lost revenue annually in the U.S. alone. In this context, an added advantage of MPC is the ability to handle constraints such that when combined with fault tolerant control strategies introduces flexibility and optimization that can not only avert disaster in the case of an abnormal situation but can maintain optimal plant operation.

Traditionally, control systems rely on centralized control architectures utilizing dedicated wired links to measurement sensors and control actuators to regulate appropriate process variables at de-

sired values. While this paradigm to process control has been successful, it is limited in the number of process state variables, manipulated inputs and measurements in a chemical plant because the computational time needed to solve a centralized control problem may increase significantly and may impede the ability of the centralized control systems (particularly when nonlinear constrained optimization-based control systems like MPC are used) to carry out real-time calculations within the limits set by process dynamics and operating conditions. One feasible alternative to overcome this problem is to utilize cooperative, distributed control architectures in which the manipulated inputs are computed by solving more than one control (optimization) problems in separate processors in a coordinated fashion. Cooperative, distributed control systems can also take advantage of additional sensing/actuation capabilities and network accessible data to dramatically improve process performance and deal with abnormal situations (see [55, 7] for a series of papers and reports calling for attention to the broad issue of distributed decision making/control in the context of chemical plants).

MPC is a natural control framework to deal with the design of cooperative, distributed control systems because of its ability to handle input and state constraints, and also because it can compensate for the actions of other actuators in computing the control response of a given set of control inputs in real-time. With respect to available results in this direction, several distributed MPC (DMPC) methods have been proposed in the literature that deal with the coordination of separate MPC controllers that communicate in order to obtain optimal input trajectories in a distributed manner; see [3, 41, 44] for reviews of results in this area. More specifically, in [12], the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints was considered. In [42, 18], the effect of the coupling was modeled as a bounded disturbance compensated with a robust MPC formulation. In [52], it was proven that through the use of multiple communications between distributed controllers and the use of system-wide control objective functions, stability of the closed-loop system can be guaranteed. In [19], DMPC of decoupled systems (a class of systems relevant to the context of multi-agents systems) was studied. In [29], an MPC algorithm was proposed for the case where the nonlinear system is discrete-time and no information exchange exist between the local controllers, and in [40], the MPC for nonlinear systems was studied from an input-to-state stability point of view. A game theory based DMPC scheme for constrained linear systems was proposed in [28].

In a previous work [25], a DMPC architecture with one-directional communication for nonlinear process systems was proposed. In this architecture, two separate MPC algorithms designed via Lyapunov-based MPC (LMPC) were considered, in which one LMPC was used to guarantee the stability of the closed-loop system and the other LMPC was used to improve the closed-loop performance. In [26], the design of DMPC architectures for systems with asynchronous and delayed measurements was also considered. In a recent work [24], the DMPC architecture developed in [25] was extended to include multiple distributed controllers and the requirement that one of the distributed controllers should be able to stabilize the closed-loop system was relaxed. In the new DMPC architecture proposed in [24], there are several distributed controllers, where individually they can not stabilize the closed-loop system, but cooperatively can achieve closed-loop stability and a desired level of closed-loop performance. The above results deal with the design of DMPC systems and do not address the problems of monitoring and reconfiguration of DMPC in the event of actuator faults.

On the other hand, the occurrence of faults in chemical processes poses a number of challenges in process monitoring and fault-tolerant control. Specifically, the problem of using fundamental process models for the purpose of detecting faults has been studied extensively in the context of linear systems [14, 15, 57, 30]; and also, some existential results in the context of nonlinear systems have been derived [10, 11]. The model-based approach to fault detection relies on the use of fundamental models for the construction of residuals, that capture some measure of the difference between normal and ‘faulty’ dynamics, to achieve fault detection and isolation. Fault-tolerant control has been an active area of research primarily within the context of aerospace control engineering (see, e.g., [37]). Over the last ten years, our group has initiated an effort of FTC on nonlinear processes by trying to bring together the disconnected fields of process fault-diagnosis and nonlinear process control. We have looked at both actuator [32] and sensor [33] faults and their impact and handling in the context of chemical process control. Despite this progress, there are no results on monitoring and reconfiguration of cooperative, distributed control systems.

1.2 Thesis objectives and structure

The objective of this dissertation is to present novel methods of monitoring and reconfiguration of distributed model predictive control systems applied to general nonlinear processes in the event of control actuator faults. Specifically, we consider nonlinear systems controlled with a cooperative, distributed control scheme in which several Lyapunov-based model predictive controllers manipulate different sets of control inputs and coordinate their actions to achieve the desired closed-loop stability and performance specifications. To deal with control actuator faults which may reduce the ability of the distributed control system to stabilize the process, we design a model-based fault detection and isolation and fault-tolerant control system. The main concepts behind fault-tolerant control consist of detecting a fault, localizing the source of the fault, and using logic-based switching between different control configurations, each of which contains multiple control laws and different sets of manipulated inputs, to provide different regions of closed-loop stability. A detailed mathematical analysis is carried out to determine precise conditions for the stabilizability of the FDI and FTC systems.

The thesis is organized as follows: Chapter 2 presents the basic framework for fault detection, isolation and fault-tolerant control which the other chapters expand on. Chapter 3 introduces an extension of the FDIFTC system to address a larger range of faults. The effectiveness of FDI and FTC systems for monitoring and reconfiguration of DMPC systems applied to a nonlinear chemical process in the presence of actuator faults is further demonstrated in a more practical setting in chapter 4. Chapter 5 takes the FDIFTC methodologies presented in previous chapters and adapts them to address the problem of monitoring and retuning of low-level PID control loops.

Chapter 2

Detection, Isolation and Handling of Actuator Faults in Distributed Model Predictive Control Systems

2.1 Introduction

The focus of this chapter is on the development of FDI and FTC systems for the monitoring and reconfiguration of DMPC systems applied to general nonlinear processes in the presence of control actuator faults. Specifically, we consider a DMPC system in which two distributed LMPC controllers manipulate two different sets of control inputs and coordinate their actions to achieve closed-loop stability and performance specifications. We first design a model-based FDI system which effectively detects and isolates actuator faults; and then based on the assumption that there exists a backup control configuration which is able to stabilize the closed-loop system within the DMPC system, we develop FTC switching rules to handle faults in the actuators of the distributed control system to minimize closed-loop system performance degradation. Sufficient conditions for the stabilizability of the FDI and FTC system are obtained based on a detailed mathematical analysis. The proposed design is applied to a chemical process example, consisting of two continuous stirred tank reactors (CSTRs) and a flash tank separator with a recycle stream operated at an unstable steady state, to demonstrate its applicability and effectiveness.

2.2 Problem formulation and preliminaries

2.2.1 Class of nonlinear systems

We consider nonlinear process systems described by the following state-space model

$$\dot{x} = f(x) + g_1(x)(u_1 + \tilde{u}_1) + g_2(x)(u_2 + \tilde{u}_2) \quad (2.1)$$

where $x \in R^n$ denotes the set of state variables, $u_1 \in R^{m_1}$ and $u_2 \in R^{m_2}$ denote two sets of manipulated inputs, $\tilde{u}_1 \in R^{m_1}$ and $\tilde{u}_2 \in R^{m_2}$ denote the unknown fault vectors for u_1 and u_2 , respectively. We consider that $u_1 + \tilde{u}_1$ and $u_2 + \tilde{u}_2$ take values in non-empty convex sets $U_1 \in R^{m_1}$ and $U_2 \in R^{m_2}$, respectively. The convex sets U_1 and U_2 are defined as follows:

$$\begin{aligned} U_1 &= \{u_1 + \tilde{u}_1 \in R^{m_1} : |u_1 + \tilde{u}_1| \leq u_1^{\max}\} \\ U_2 &= \{u_2 + \tilde{u}_2 \in R^{m_2} : |u_2 + \tilde{u}_2| \leq u_2^{\max}\}. \end{aligned}$$

We consider a different fault $\tilde{u}_{f,j} \in R$, $j = 1, \dots, m_1 + m_2$, for each element of the vector $[\tilde{u}_1^T \ \tilde{u}_2^T]^T \in R^{m_1+m_2}$. Under fault-free operating conditions, we have $\tilde{u}_1 = 0$ and $\tilde{u}_2 = 0$, and hence, $\tilde{u}_{f,j} = 0$ for all $j = 1, \dots, m_1 + m_2$. When fault j occurs, $\tilde{u}_{f,j}$ takes a non-zero value. We assume that f , g_1 , g_2 are locally Lipschitz vector functions and that $f(0) = 0$. This means that the origin is an equilibrium point for the fault-free system ($\tilde{u}_1 = 0$ and $\tilde{u}_2 = 0$ for all t) with $u_1 = 0$ and $u_2 = 0$. We also assume that the state x of the system is sampled synchronously and continuously and the time instants where we have measurement samplings are indicated by the time sequence $\{t_{k \geq 0}\}$ with $t_k = t_0 + k\Delta$, $k = 0, 1, \dots$ where t_0 is the initial time and Δ is the sampling time.

Remark 2.1. *The variable $\tilde{u}_{f,j}$ associated with the j^{th} element in $[u_1^T \ u_2^T]^T$ can be used to model different kinds of faults that may occur in an actuator. For example, $\tilde{u}_{f,j}$ can model a constant deviation of the control input from its calculated value u_j ; or it can be a function of the form $\tilde{u}_{f,j} = -u_j + c$ to model faults in an actuator that keep the output of the actuator constant. We also note that the approach presented here can be extended to handle actuator faults in DMPC systems which include multiple controllers.*

2.2.2 Lyapunov-based controller

We assume that there exists a Lyapunov-based controller $u_1(t) = h(x)$ which renders the origin of the fault-free closed-loop system asymptotically stable with $u_2(t) = 0$. This assumption is essentially a standard stabilizability requirement made in all linear/nonlinear control methods and implies that, in principle, it is not necessary to use the extra input u_2 in order to achieve closed-loop stability. However, one of the main objectives of the distributed control method is to profit from the extra control effort to improve the closed-loop performance while maintaining the stability properties achieved by only implementing u_1 . Using converse Lyapunov theorems [20], this assumption implies that there exist functions $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$ of class \mathcal{K}^* and a continuous differentiable Lyapunov function $V(x)$ for the nominal closed-loop system that satisfy the following inequalities:

$$\begin{aligned} \alpha_1(|x|) &\leq V(x) \leq \alpha_2(|x|) \\ \frac{\partial V(x)}{\partial x}(f(x) + g_1(x)h(x)) &\leq -\alpha_3(|x|) \\ \left| \frac{\partial V(x)}{\partial x} \right| &\leq \alpha_4(|x|) \\ h(x) &\in U_1 \end{aligned} \tag{2.2}$$

for all $x \in D \subseteq R^n$ where D is an open neighborhood of the origin. We denote the region $\Omega_\rho^\dagger \subseteq D$ as the stability region of the closed-loop system under the control $u_1 = h(x)$ and $u_2 = 0$. We also note that: a) in certain applications it is possible to attain global stability under $h(x)$ (i.e., $D = R^n$), b) the construction of $V(x)$ can be readily done using a variety of methods (see [20] for examples), c) dynamic local controllers, like for example proportional-integral (PI) controllers, can be used in a straightforward fashion as $h(x)$, and d) while we address here stabilization of $x = 0$, the problem of set-point tracking can be readily handled by working with deviation variables with respect to the desired, non-zero operating point.

By continuity and the local Lipschitz property assumed for the vector fields f , g_1 and g_2 , the fact that the manipulated inputs $u_1 + \tilde{u}_1$ and $u_2 + \tilde{u}_2$ are bounded in convex sets and the continuous differentiable property of the Lyapunov function V , there exist positive constant M_1 and $L_{x,1}$ such

*A continuous function $\alpha : [0, a) \rightarrow [0, \infty)$ is said to belong to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$.

†We use Ω_r to denote the set $\Omega_\rho := \{x \in R^n : V(x) \leq \rho\}$.

that

$$|f(x) + g_1(x)(u_1 + \tilde{u}_1) + g_2(x)(u_2 + \tilde{u}_2)| \leq M_1 \quad (2.3)$$

$$\left| \frac{\partial V}{\partial x}(f(x) + g_1(x)u_1 + g_2(x)u_2) - \frac{\partial V}{\partial x}(f(x') + g_1(x')u_1 + g_2(x')u_2) \right| \leq L_{x,1}|x - x'| \quad (2.4)$$

for all $x, x' \in \Omega_\rho$, $u_1 + \tilde{u}_1 \in U_1$ and $u_2 + \tilde{u}_2 \in U_2$.

2.2.3 DMPC design for fault-free system

Following [25], we design a DMPC architecture to achieve the desired closed-loop system stability and performance specifications and to reduce the computational burden in the evaluation of the optimal manipulated inputs. Specifically, for the system of Eq. 2.1, we design two separate LMPC controllers to compute u_1 and u_2 and refer to the LMPCs computing the trajectories of u_1 and u_2 as LMPC 1 and LMPC 2, respectively. The implementation strategy of the DMPC is as follows: 1) at each sampling instant t_k , both LMPC 1 and LMPC 2 receive the state measurement $x(t_k)$ from the sensors; 2) LMPC 2 evaluates the optimal input trajectory of u_2 based on $x(t_k)$ and sends the first step input value to its corresponding actuators and the entire optimal input trajectory to LMPC 1; 3) once LMPC 1 receives the entire optimal input trajectory for u_2 from LMPC 2, it evaluates the future input trajectory of u_1 based on the $x(t_k)$ and the entire optimal input trajectory of u_2 ; 4) LMPC 1 sends the first step input value of u_1 to its corresponding actuators.

We first discuss the design of LMPC 2. The optimization problem of LMPC 2 depends on the latest state measurement $x(t_k)$, however, LMPC 2 does not have any information about the value that u_1 will take. In order to make a decision, LMPC 2 must assume a trajectory for u_1 along the prediction horizon. To this end, the Lyapunov-based controller $u_1 = h(x)$ is used. In order to inherit the stability properties of this controller, u_2 must satisfy a stability constraint that guarantees a given minimum decrease rate of the Lyapunov function $V(x)$. The LMPC 2 is based

on the following optimization problem:

$$\min_{u_2 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + u_1(\tau)^T R_{c1} u_1(\tau) + u_2(\tau)^T R_{c2} u_2(\tau)] d\tau \quad (2.5a)$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + g_1(\tilde{x}(\tau))u_1(\tau) + g_2(\tilde{x}(\tau))u_2(\tau) \quad (2.5b)$$

$$u_1(\tau) = h(\tilde{x}(j\Delta)), \quad \forall \tau \in [j\Delta, (j+1)\Delta), \quad j = 0, \dots, N-1 \quad (2.5c)$$

$$\tilde{x}(0) = x(t_k) \quad (2.5d)$$

$$u_2(\tau) \in U_2 \quad (2.5e)$$

$$\frac{\partial V(x)}{\partial x} g_2(x(t_k)) u_2(0) \leq 0. \quad (2.5f)$$

In the optimization problem of Eq. 2.5, \tilde{x} is the predicted trajectory of the fault-free system with u_2 being the input trajectory computed by LMPC 2 of Eq. 2.5 and u_1 being the Lyapunov-based controller $h(x)$ applied in a sample-and-hold fashion. Δ is the sampling rate of the controller, Q_c , R_{c1} and R_{c2} are positive definite weighting matrices and N is the prediction horizon. The optimal solution to this optimization problem is denoted by $u_2^*(\tau|t_k)$. This information is sent to LMPC 1. The constraint of Eq. 2.5e defines the constraint on the manipulated input u_2 and the stability constraint of Eq. 2.5f is required to guarantee the closed-loop stability.

Next, we discuss the design of LMPC 1. The optimization problem of LMPC 1 depends on $x(t_k)$ and the decision made by LMPC 2 (i.e., $u_2^*(\tau|t_k)$). This allows LMPC 1 to compute an input u_1 such that the closed-loop performance is optimized, while guaranteeing that the stability properties of the Lyapunov-based controller are preserved. Specifically, LMPC 1 is based on the following optimization problem:

$$\min_{u_1 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + u_1(\tau)^T R_{c1} u_1(\tau) + u_2^*(\tau|t_k)^T R_{c2} u_2^*(\tau|t_k)] d\tau \quad (2.6a)$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + g_1(\tilde{x}(\tau))u_1(\tau) + g_2(\tilde{x}(\tau))u_2^*(\tau|t_k) \quad (2.6b)$$

$$\tilde{x}(0) = x(t_k) \quad (2.6c)$$

$$u_1(\tau) \in U_1 \quad (2.6d)$$

$$\frac{\partial V(x)}{\partial x} g_1(x(t_k)) u_1(0) \leq \frac{\partial V(x)}{\partial x} g_1(x(t_k)) h(x(t_k)). \quad (2.6e)$$

In the optimization problem of Eq. 2.6, \tilde{x} is the predicted trajectory of the fault-free system with u_2 being the optimal input trajectory $u_2^*(\tau|t_k)$ computed by LMPC 2 and u_1 being the input trajectory computed by LMPC 1 of Eq. 2.6. The optimal solution to this optimization problem is denoted by $u_1^*(\tau|t_k)$. The constraint of Eq. 2.6d defines the constraint on the manipulated input u_1 and the stability constraint of Eq. 2.6e is also required to guarantee the closed-loop stability.

Once both optimization problems are solved (see [25] for results on the feasibility and stability of the LMPCs of Eq. 2.5 and 2.6), the manipulated inputs of the DMPC system based on LMPC 1 and LMPC 2 are defined as follows:

$$\begin{aligned} u_1^L(t|x) &= u_1^*(t - t_k|t_k), \quad \forall t \in [t_k, t_{k+1}) \\ u_2^L(t|x) &= u_2^*(t - t_k|t_k), \quad \forall t \in [t_k, t_{k+1}). \end{aligned}$$

The closed-loop system of Eq. 2.1 under this DMPC scheme with inputs defined by $u_1 = u_1^L$ and $u_2 = u_2^L$ maintains the same stability region Ω_ρ and asymptotic stability as the Lyapunov-based control law h [25].

2.2.4 FTC considerations and backup DMPC design

In order to carry out FTC, there must be a backup control configuration for the system under consideration. The presence of the control action u_2 brings extra control flexibility to the closed-loop system which can be used to carry out FTC. Specifically, we assume that the control input u_1 can be decomposed into two subsets. That is $u_1 = [u_{11}^T \ u_{12}^T]^T$. We further assume that, under continuous state measurements, there exists a Lyapunov-based control law $h_2(x) = [h_{21}(x)^T \ h_{22}(x)^T]^T$ which is able to asymptotically stabilize the closed-loop system and satisfies the input constraints on u_1 and u_2 while controlling only u_{11} and u_2 ; that is, $u_{11} = h_{21}(x)$, $u_{12} = 0$ and $u_2 = h_{22}(x)$.

Using converse Lyapunov theorems, this assumption on h_2 implies that there exist functions $\alpha'_i(\cdot)$, $i = 1, 2, 3, 4$ of class \mathcal{K} and a continuously differentiable Lyapunov function $V_2(x)$ for the fault-free system of Eq. 2.1 with $u_{11} = h_{21}(x)$, $u_2 = h_{22}(x)$ and $u_{12} = 0$ that satisfy the following

inequalities

$$\begin{aligned}
\alpha'_1(|x|) &\leq V_2(x) \leq \alpha'_2(|x|) \\
\frac{\partial V(x)}{\partial x} (f(x) + g_1(x)[h_{21}(x)^T \ 0^T]^T + g_2(x)h_{22}(x)) &\leq -\alpha'_3(|x|) \\
\left| \frac{\partial V_2(x)}{\partial x} \right| &\leq \alpha'_4(|x|) \\
h_{21}(x) &\in U_1, \quad h_{22}(x) \in U_2
\end{aligned} \tag{2.7}$$

for all $x \in D_2 \subseteq R^n$ where D_2 is an open neighborhood of the origin. We denote $\Omega_{2,\gamma}^\ddagger \subseteq D_2$ as the stability region of the closed-loop fault-free system with $u_1 = [h_{21}(x)^T \ 0^T]^T$ and $u_2 = h_{22}(x)$.

Similarly there exist positive constants M_2 and $L_{x,2}$ such that

$$|f(x) + g_1(x)(u_1 + \tilde{u}_1) + g_2(x)(u_2 + \tilde{u}_2)| \leq M_2 \tag{2.8}$$

$$\left| \frac{\partial V_2}{\partial x} (f(x) + g_1(x)u_1 + g_2(x)u_2) - \frac{\partial V_2}{\partial x} (f(x') + g_1(x')u_1 + g_2(x')u_2) \right| \leq L_{x,2}|x - x'| \tag{2.9}$$

for all $x, x' \in \Omega_{2,\gamma}$, $u_1 + \tilde{u}_1 \in U_1$ and $u_2 + \tilde{u}_2 \in U_2$.

Based on $h_2(x)$, we can design a backup DMPC system to manipulate u_{11} and u_2 to stabilize the closed-loop system following the results developed in [24]. We still design two LMPC controllers in this DMPC system. One LMPC is used to manipulated u_{11} and the other one is used to manipulate u_2 . We refer to the LMPC manipulating u_{11} as the backup LMPC 1 and the LMPC manipulating u_2 as the backup LMPC 2. The implementation strategy of the backup DMPC is the same as the one used by the DMPC system introduced in Section 2.2.3.

[‡]We use $\Omega_{2,\gamma}$ to denote the set $\Omega_{2,\gamma} := \{x \in R^n : V_2(x) \leq \gamma\}$.

The backup LMPC 2 optimizes u_2 and is designed as follows:

$$\min_{u_2 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + u_1(\tau)^T R_{c1} u_1(\tau) + u_2(\tau)^T R_{c2} u_2(\tau)] d\tau \quad (2.10a)$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + g_1(\tilde{x}(\tau)) [u_{11}(\tau)^T u_{12}(\tau)^T]^T + g_2(\tilde{x}(\tau)) u_2(\tau) \quad (2.10b)$$

$$u_{11}(\tau) = h_{21}(\tilde{x}(j\Delta)), \quad \forall \tau \in [j\Delta, (j+1)\Delta), \quad j = 0, \dots, N-1 \quad (2.10c)$$

$$u_{12}(\tau) = 0 \quad (2.10d)$$

$$\tilde{x}(0) = x(t_k) \quad (2.10e)$$

$$u_2(\tau) \in U_2 \quad (2.10f)$$

$$\frac{\partial V(x)}{\partial x} g_2(x(t_k)) u_2(0) \leq \frac{\partial V(x)}{\partial x} g_2(x(t_k)) h_{22}(x(t_k)). \quad (2.10g)$$

The solution to the optimization problem of Eq. 2.10 is denoted by $u_2^{b,*}(\tau|t_k)$. The backup LMPC 1 optimizes u_{11} and is designed as follows:

$$\min_{u_{11} \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + u_1(\tau)^T R_{c1} u_1(\tau) + u_2^{b,*}(\tau|t_k)^T R_{c2} u_2^{b,*}(\tau|t_k)] d\tau \quad (2.11a)$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + g_1(\tilde{x}(\tau)) [u_{11}(\tau)^T 0^T]^T + g_2(\tilde{x}(\tau)) u_2^{b,*}(\tau|t_k) \quad (2.11b)$$

$$\tilde{x}(0) = x(t_k) \quad (2.11c)$$

$$u_{11}(\tau) \in U_1 \quad (2.11d)$$

$$\frac{\partial V(x)}{\partial x} g_1(x(t_k)) [u_{11}(0)^T 0^T]^T \leq \frac{\partial V(x)}{\partial x} g_1(x(t_k)) [h_{21}(x(t_k))^T 0^T]^T. \quad (2.11e)$$

The solution to the optimization problem of Eq. 2.11 is denoted by $u_{11}^{b,*}(\tau|t_k)$. The control inputs of the closed-loop system under the backup DMPC are defined as follows:

$$\begin{aligned} u_{11}^b(t|x) &= u_{11}^{b,*}(t - t_k|t_k), \quad \forall t \in [t_k, t_{k+1}) \\ u_{12}^b(t|x) &= 0, \quad \forall t \\ u_2^b(t|x) &= u_2^{b,*}(t - t_k|t_k), \quad \forall t \in [t_k, t_{k+1}). \end{aligned} \quad (2.12)$$

The fault-free closed-loop system of Eq. 2.1 under the backup DMPC control with inputs defined by $u_{11} = u_{11}^b$, $u_{12} = 0$ and $u_2 = u_2^b$ maintains the same stability region $\Omega_{2,\gamma}$ as $h_2(x)$ and is practically stable [24].

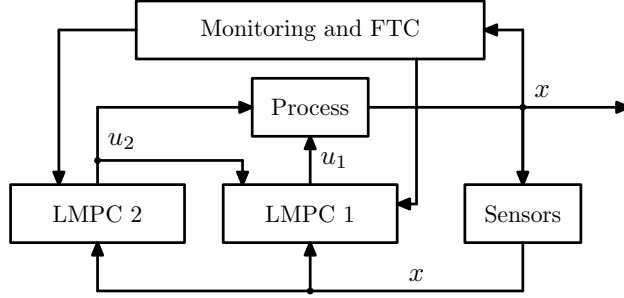


Figure 2.1: Proposed FDI and FTC structure for DMPC.

Remark 2.2. Note that in the DMPC design of Eqs. 2.5-2.6, the main objective of LMPC 1 is to stabilize the closed-loop system and the main objective of LMPC 2 is to maintain the closed-loop stability achieved by LMPC 1 and try to improve the closed-loop performance. This DMPC design has the potential to maintain the closed-loop stability and performance in the face of failing controllers or actuators, for example, a zero input of LMPC 2 does not affect the closed-loop stability. On the other hand, in the backup DMPC design of Eqs. 2.10-2.11, LMPC 1 and LMPC 2 are both needed in order to guarantee the closed-loop stability, and this design may be fragile to controller or actuator failures.

Remark 2.3. The assumption that there exists a Lyapunov-based control law h_2 that can stabilize the closed-loop system by manipulating u_{11} and u_2 implies that when there is a fault in the subset u_{12} of u_1 , we can switch off the actuators associated with u_{12} and the remaining control actions (i.e., u_{11} and u_2) can still maintain the closed-loop stability. Please see Section 2.3.2 for further discussion on this issue.

Remark 2.4. Note that the proposed backup control configuration is one of the many possible options for FTC; however, under the proposed backup control configuration, stability of the closed-loop system can be proved. Please see Section 2.3 for the proposed fault-tolerant control methods and see [34, 32] for more discussion on the relationship between system structure and FTC schemes.

2.3 FDI and FTC strategies

In this section, we look at the closed-loop system under the DMPC control of Eqs. 2.5-2.6 where, upon detection and isolation of actuator faults, the DMPC control system can be switched off or

reconfigured to maintain stability of the closed-loop system. The structure of the integrated system is shown in Fig. 2.1.

2.3.1 FDI system design

We consider control actuator faults that can be detected by an appropriate nonlinear dynamic filter by observing the evolution of the closed-loop system state. This consideration requires that a fault in a control actuator influences the evolution of at least some of the states. In order to isolate the occurrence of a fault, it is further required to assume that the control actuator in question is the only one influencing a certain set of the system states (i.e., each fault has a unique fault signature). For more discussions on systems having verifiable isolable structures, please see [34, 36].

The DMPC system of Eqs. 2.5-2.6 is the control configuration for the fault-free system of Eq. 2.1. We first design an FDI scheme to detect faults in this control system. In this FDI scheme, a filter is designed for each state and the design of the filter for the p^{th} , $p = 1, \dots, n$, state in the system state vector x is as follows [34]:

$$\dot{\hat{x}}_p(t) = f_p(X_p) + g_{1p}(X_p)u_1^L(X_p) + g_{2p}(X_p)u_2^L(X_p) \quad (2.13)$$

where \hat{x}_p is the filter output for the p^{th} state, f_p , g_{1p} and g_{2p} are the p^{th} components of the vector functions f , g_1 and g_2 , respectively. With a little abuse of notation, we have dropped the time index in the control functions and denote $u_1^L(t|x)$, $u_2^L(t|x)$ with $u_1^L(x)$, $u_2^L(x)$, respectively, in order to simplify the FDI definitions. The state X_p is obtained from both the actual state measurements, x , and the filter output, \hat{x}_p , as follows:

$$X_p(t) = [x_1(t), \dots, x_{p-1}(t), \hat{x}_p(t), x_{p+1}(t), \dots, x_n(t)]^T.$$

Note that in the filter of Eq. 2.13, the control inputs $u_1^L(X_p)$ and $u_2^L(X_p)$ are determined by the same LMPC 1 of Eq. 2.6 and the LMPC 2 of Eq. 2.5 as applied to the actual process, and are updated every control sampling time Δ (i.e., the sampling time instants $\{t_{k \geq 0}\}$).

The states of the FDI filters are initialized at $t = 0$ to the actual state values; that is, $\hat{x}_p = x_p$. The FDI filters are only initialized at $t = 0$ such that $\hat{x}_p(0) = x_p(0)$. The information generated by

the filters provides a fault-free estimate of the real state at any time t and allows easy detection of the actual system deviating due to faults. For each state associated with a filter, the FDI residual can be defined as [34]:

$$r_p(t) = |\hat{x}_p(t) - x_p(t)|,$$

with $p = 1, \dots, n$. The residual r_p is computed continuously because $\hat{x}_p(t)$ is known for all t and the state measurement, x , is also available for all t . If no fault occurs, the filter states track the system states. In this case, the dynamics of the system states and the FDI filter states are identical, so $r_p(t) = 0$ for all times. When there is a fault in the system, filter residuals affected directly by the fault will deviate from zero soon after the occurrence of the fault. For more detailed discussion on the properties of the filters, please see [34].

Note that due to sensor measurement and process noise, the residuals will be nonzero even without an actuator fault. This necessitates the use of fault detection thresholds so that a fault is declared only when a residual exceeds a specific threshold value, $r_{p,\max}$. This threshold value is chosen to avoid false alarms due to process and sensor measurement noise, but should still be sensitive enough to detect faults in a timely manner so that effective fault-tolerant control can be performed.

The objective of the FDI scheme is to quickly detect an actuator fault when it occurs, and then identify which of the $m_1 + m_2$ possible different actuator faults (i.e., $\tilde{u}_{f,j}$, $j = 1, \dots, m_1 + m_2$) has occurred. When a fault $\tilde{u}_{f,j}$ occurs, one or more of the filter residuals will become nonzero. Once a fault is detected, the monitoring system will declare a fault alarm. In order to isolate a fault, the system must have an isolable structure in which different faults have different fault signatures. If a fault is isolated, an FTC system may be used which will send the fault information and reconfiguration policy to the two distributed controllers as shown in Fig. 2.1.

2.3.2 FTC system design

When an actuator fault is detected and isolated, automated FTC action can be initiated. An FTC switching rule may be employed to orchestrate the reconfiguration of the control system. This rule determines which of the backup control loops can be activated, in the event that the main control loop fails, in order to preserve closed-loop stability. In general, when there is a fault in the control

system, it is impossible to carry out FTC unless there is another backup control loop. However, in the distributed control architecture introduced in Section 2.2.3, because of the extra control flexibility brought into the whole system by u_2 (LMPC 2), it is possible in some cases to carry out FTC when there is a fault in the control system without activating new control actuators.

When there is a fault in the loop of u_2 , and the fault can be detected and isolated in a reasonable time frame, it is possible to shut down the control action of u_2 and to only use u_1 in the control system. This FTC strategy will maintain the closed-loop stability, however, the performance of the closed-loop system may degrade to some extent. When the loop of u_2 is shut down, in the DMPC scheme of Eqs. 2.5-2.6, only LMPC 1 is evaluated each sampling time, LMPC 1 does not have to be modified and does not wait for the information sent by LMPC 2. In this case, the input trajectory of LMPC 2 is replaced by a zero trajectory (i.e., $u_2^*(\tau|x(t_k)) = 0$ for $\tau \in [0, N\Delta)$). Theorem 1 below describes the switching rule and the stability properties of the closed-loop system when there is an actuator fault in the loop of u_2 .

When there is a fault in the loop of u_1 , successful FTC depends on the availability of backup control loops. From the analysis of Section 2.2.4, we know u_1 is essential for the stabilization of the closed-loop system, however, because of the extra control flexibility introduced by u_2 , there may exist a subset of u_1 , that is u_{11} , which together with u_2 can stabilize the closed-loop system. When there is a fault in the subset u_{12} , the FTC strategy would shut down the control action of u_{12} and reconfigure the DMPC algorithms to the backup DMPC of Eqs. 2.10-2.11 to manipulate u_{11} and u_2 to control the process. Theorem 2 states the switching rule and reconfiguration strategy for this case.

However, when there is a fault in the subset u_{11} , it is impossible to successfully carry out FTC without activating backup actuators within the DMPC systems and class of nonlinear systems considered in this work.

The proposed FTC switching rules for the system of Eq. 2.1 within the DMPC system of Eqs. 2.5-2.6 are described as follows:

1. When a fault in the actuator associated with u_2 is detected at t_f , the proposed FTC switching

rule is:

$$\begin{aligned} u_1(t) &= u_1^L(x), \quad \forall t \\ u_2(t) &= \begin{cases} u_2^L(x), & t \leq t_f \\ 0, & t > t_f \end{cases} \end{aligned} \quad (2.14)$$

2. When a fault in the actuator associated with u_{12} is detected at t_f , the proposed FTC switching rule is:

$$u_1(t) = \begin{cases} u_1^L(x), & t \leq t_f \\ \begin{bmatrix} u_{11}^b(x) \\ 0 \end{bmatrix}, & t > t_f \end{cases} \quad (2.15a)$$

$$u_2(t) = \begin{cases} u_2^L(x), & t \leq t_f \\ u_2^b(x), & t > t_f \end{cases} \quad (2.15b)$$

In what follows, we summarize the properties of the switching rules of Eqs. 2.14 and 2.15 in Theorems 1 and 2. In order to state and prove the two theorems, we first introduce the following proposition.

Proposition 1 (c.f. [35]). *Consider the sampled trajectory \hat{x} of the fault-free system of Eq. 2.1 in closed-loop with the Lyapunov-based control law h applied in a sample-and-hold fashion. Let $\Delta, \epsilon_s > 0$ and $\rho > \rho_s > 0$ satisfy*

$$-\alpha_3(\alpha_2^{-1}(\rho_s)) + \alpha_4(\alpha_1^{-1}(\rho))L_{x,1}M_1\Delta \leq -\epsilon_s/\Delta. \quad (2.16)$$

Then, if $\rho_{\min} < \rho$ where

$$\rho_{\min} = \max\{\rho_s, \max\{V(\hat{x}(t + \Delta)) : V(\hat{x}(t)) \leq \rho_s\}\} \quad (2.17)$$

and $\hat{x}(0) \in \Omega_\rho$, the following inequality holds

$$V(\hat{x}(k\Delta)) \leq \max\{V(\hat{x}(0)) - k\epsilon_s, \rho_{\min}\}. \quad (2.18)$$

Proposition 1 ensures that if the fault-free system of Eq. 2.1 under the control law $h(x)$ implemented in a sample-and-hold fashion starts in Ω_ρ , then it is ultimately bounded in $\Omega_{\rho_{\min}}$. By applying Proposition 1, we know that when the fault-free system of Eq. 2.1 is controlled under $h_2(x)$ implemented in a sample-and-hold fashion, there exists a region $\Omega_{2,\gamma_{\min}}$ in which the state of the closed-loop system, starting in $\Omega_{2,\gamma}$, is ultimately bounded.

Theorem 1. *Consider the system of Eq. 2.1 in closed-loop under the DMPC scheme of Eqs. 2.5-2.6. If $x(t_0) \in \Omega_\rho$ where t_0 is the initial time, and a fault in u_2 is detected and isolated at time t_f , then the switching rule of Eq. 2.14 guarantees that the state of the closed-loop system $x(t)$ is ultimately bounded in $\Omega_{\rho_{\min}}$.*

Proof. Assume that a fault occurs at time t_f in u_2 . Because of the properties of the filter design of Eq. 2.13, this fault can be detected and isolated immediately after t_f . According to the switching rule of Eq. 2.14, from t_0 to t_f , the closed-loop system of Eq. 2.1 is controlled under the DMPC scheme of Eq. 2.5-2.6 with $u_1 = u_1^L(x)$ and $u_2 = u_2^L(x)$. Following from the practical stability property of the DMPC scheme of Eqs. 2.5-2.6, if $x(t_0) \in \Omega_\rho$, the state of the closed-loop system of Eq. 2.1 will stay in Ω_ρ and converge to $\Omega_{\rho_{\min}}$, which implies that at t_f , the closed-loop system state is still in the stability region of $h(x)$, that is $x(t_f) \in \Omega_\rho$.

According to the switching rule of Eq. 2.14, after t_f , the closed-loop system will be controlled with $u_1 = u_1^L$ and $u_2 = 0$. Because of the fact that $x(t_f) \in \Omega_\rho$ and because of the stability properties of the LMPC 1 of Eq. 2.6, the closed-loop state will converge to the region $\Omega_{\rho_{\min}}$ and will be ultimately bounded in $\Omega_{\rho_{\min}}$. This proves Theorem 1. \square

Theorem 2. *Consider the system of Eq. 2.1 in closed-loop under the DMPC scheme of Eqs. 2.5-2.6. If $x(t_0) \in \Omega_\rho$ where t_0 is the initial time and a fault in u_{12} is detected and isolated at time t_f , and if $x(t_f) \in \Omega_{2,\gamma}$, then the switching rule of Eq. 2.15 guarantees that the state of the closed-loop system $x(t)$ is ultimately bounded in $\Omega_{2,\gamma_{\min}}$.*

Proof. Assume that a fault occurs at t_f in u_{12} . Because of the properties of the filter design of Eq. 2.13, this fault can be detected and isolated immediately after t_f . According to the switching rule of Eq. 2.15, from t_0 to t_f , the closed-loop system of Eq. 2.1 is controlled under the DMPC scheme of Eq. 2.5-2.6 with $u_1 = u_1^L(x)$ and $u_2 = u_2^L(x)$. Following from the practical stability

property of the DMPC scheme of Eqs. 2.5-2.6, if $x(t_0) \in \Omega_\rho$, the state of the closed-loop system of Eq. 2.1 will be maintained in Ω_ρ .

According to the switching rule of Eq. 2.14, after t_f , the closed-loop system will be controlled with $u_{11} = u_{11}^b$, $u_{12} = 0$ and $u_2 = u_2^b$. If $x(t_f) \in \Omega_{2,\gamma}$, the closed-loop state will converge to the region $\Omega_{2,\gamma_{\min}}$ and will be ultimately bounded in $\Omega_{2,\gamma_{\min}}$ following the practical stability property of the backup DMPC scheme of Eqs. 2.10-2.11. This proves Theorem 2. \square

Remark 2.5. *Note that in this work, we assume that upon detection and isolation of a control actuator fault, the faulty actuator can be shut down and the influence of the faulty actuator can be completely separated from the rest of the system. This assumption implies that in the normal fault-free operation and the operation after FTC reconfiguration, the steady-state of the system considered remains unchanged.*

Remark 2.6. *In Theorems 1 and 2, we do not consider process or measurement noise and assume that a fault can be detected and isolated immediately after its occurrence. However, in the presence of process and measurement noise, faults are detected and isolated when their corresponding residuals exceed their thresholds which introduce delays in the FDI process. These delays may degrade the performance but the closed-loop stability under the proposed FTC switching rules can be maintained provided the delays are small enough. This point is demonstrated in the application of the proposed methods to a chemical process in Section 2.4.*

2.4 Application to a reactor-separator process

2.4.1 Process description and modeling

The process considered in this study is a three vessel, reactor-separator system consisting of two CSTRs and a flash tank separator as shown in Fig. 3.2. A feed stream to the first CSTR contains the reactant, A , which is converted into the desired product, B . Species A can also react into an undesired side-product, C . The solvent does not react and is labeled as D . The effluent of the first CSTR along with additional fresh feed makes up the inlet to the second CSTR. The reactions $A \rightarrow B$ and $A \rightarrow C$ (referred to as 1 and 2, respectively) take place in the two CSTRs in series before the effluent from CSTR 2 is fed to a flash tank. The overhead vapor from the flash tank is

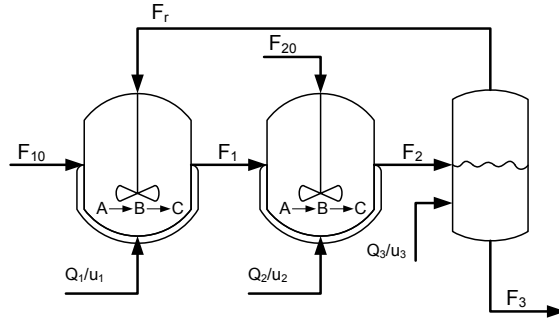


Figure 2.2: Two CSTRs and a flash tank with recycle stream.

condensed and recycled to the first CSTR, and the bottom product stream is removed. All three vessels are assumed to have static holdup. The dynamic equations describing the behavior of the system, obtained through material and energy balances under standard modeling assumptions, are given below:

$$\frac{dT_1}{dt} = \frac{F_{10}}{V_1}(T_{10} - T_1) + \frac{F_r}{V_1}(T_3 - T_1) + \frac{-\Delta H_1}{\rho C_p} k_1 e^{\frac{-E_1}{RT_1}} C_{A1} + \frac{-\Delta H_2}{\rho C_p} k_2 e^{\frac{-E_2}{RT_1}} C_{A1} + \frac{Q_1}{\rho C_p V_1} \quad (2.19a)$$

$$\frac{dC_{A1}}{dt} = \frac{F_{10}}{V_1}(C_{A10} - C_{A1}) + \frac{F_r}{V_1}(C_{Ar} - C_{A1}) - k_1 e^{\frac{-E_1}{RT_1}} C_{A1} - k_2 e^{\frac{-E_2}{RT_1}} C_{A1} \quad (2.19b)$$

$$\frac{dC_{B1}}{dt} = \frac{-F_{10}}{V_1} C_{B1} + \frac{F_r}{V_1}(C_{Br} - C_{B1}) + k_1 e^{\frac{-E_1}{RT_1}} C_{A1} \quad (2.19c)$$

$$\frac{dC_{C1}}{dt} = \frac{-F_{10}}{V_1} C_{C1} + \frac{F_r}{V_1}(C_{Cr} - C_{C1}) + k_2 e^{\frac{-E_2}{RT_1}} C_{A1} \quad (2.19d)$$

$$\begin{aligned} \frac{dT_2}{dt} = & \frac{F_1}{V_2}(T_1 - T_2) + \frac{(F_{20} + \Delta F_{20})}{V_2}(T_{20} - T_2) + \frac{-\Delta H_1}{\rho C_p} k_1 e^{\frac{-E_1}{RT_2}} C_{A2} \\ & + \frac{-\Delta H_2}{\rho C_p} k_2 e^{\frac{-E_2}{RT_2}} C_{A2} + \frac{Q_2}{\rho C_p V_2} \end{aligned} \quad (2.19e)$$

$$\frac{dC_{A2}}{dt} = \frac{F_1}{V_2}(C_{A1} - C_{A2}) + \frac{(F_{20} + \Delta F_{20})}{V_2}(C_{A20} - C_{A2}) - k_1 e^{\frac{-E_1}{RT_2}} C_{A2} - k_2 e^{\frac{-E_2}{RT_2}} C_{A2} \quad (2.19f)$$

$$\frac{dC_{B2}}{dt} = \frac{F_1}{V_2}(C_{B1} - C_{B2}) - \frac{(F_{20} + \Delta F_{20})}{V_2} C_{B2} + k_1 e^{\frac{-E_1}{RT_2}} C_{A2} \quad (2.19g)$$

$$\frac{dC_{C2}}{dt} = \frac{F_1}{V_2}(C_{C1} - C_{C2}) - \frac{(F_{20} + \Delta F_{20})}{V_2} C_{C2} + k_2 e^{\frac{-E_2}{RT_2}} C_{A2} \quad (2.19h)$$

$$\frac{dT_3}{dt} = \frac{F_2}{V_3}(T_2 - T_3) - \frac{H_{vap} F_r}{\rho C_p V_3} + \frac{Q_3}{\rho C_p V_3} \quad (2.19i)$$

$$\frac{dC_{A3}}{dt} = \frac{F_2}{V_3}(C_{A2} - C_{A3}) - \frac{F_r}{V_3}(C_{Ar} - C_{A3}) \quad (2.19j)$$

$$\frac{dC_{B3}}{dt} = \frac{F_2}{V_3}(C_{B2} - C_{B3}) - \frac{F_r}{V_3}(C_{Br} - C_{B3}) \quad (2.19k)$$

$$\frac{dC_{C3}}{dt} = \frac{F_2}{V_3}(C_{C2} - C_{C3}) - \frac{F_r}{V_3}(C_{Cr} - C_{C3}) \quad (2.19l)$$

$$(2.19m)$$

The definitions for the variables used in Eq.2.19 can be found in Table 2.1, with the parameter values given in Table 2.2. Each of the tanks has an external heat input/removal actuator. The model of the flash tank separator operates under the assumption that the relative volatility for each of the species remains constant within the operating temperature range of the flash tank. This assumption allows calculating the mass fractions in the overhead based upon the mass fractions in the liquid portion of the vessel. It has also been assumed that there is a negligible amount of reaction taking place in the separator. The following algebraic equations model the composition of

Table 2.1: Process variables

C_{A1}, C_{A2}, C_{A3}	concentrations of A in vessels 1, 2, 3
C_{B1}, C_{B2}, C_{B3}	concentrations of B in vessels 1, 2, 3
C_{C1}, C_{C2}, C_{C3}	concentrations of C in vessels 1, 2, 3
C_{Ar}, C_{Br}, C_{Cr}	concentrations of A, B, C in the recycle
T_1, T_2, T_3	temperatures in vessels 1, 2, 3
T_{10}, T_{20}	feed stream temperatures to vessels 1, 2
F_1, F_2, F_3	effluent flow rates from vessels 1, 2, 3
F_{10}, F_{20}	feed stream flow rates to vessels 1, 2
C_{A10}, C_{A20}	concentrations of A in the feed stream to vessels 1, 2
F_r	recycle flow rate
V_1, V_2, V_3	volumes of vessels 1, 2, 3
u_1, u_2, u_3, u_4	manipulated inputs
E_1, E_2	activation energy for reactions 1, 2
k_1, k_2	pre-exponential values for reactions 1, 2
$\Delta H_1, \Delta H_2$	heats of reaction for reactions 1, 2
H_{vap}	heat of vaporization
$\alpha_A, \alpha_B, \alpha_C, \alpha_D$	relative volatility of A, B, C, D
MW_A, MW_B, MW_C	molecular weights of $A, B,$ and C
C_p, R	heat capacity and gas constant

the overhead stream relative to the composition of the liquid holdup in the flash tank:

$$C_{Ar} = \frac{\alpha_A C_{A3}}{K}, C_{Br} = \frac{\alpha_B C_{B3}}{K}, C_{Cr} = \frac{\alpha_C C_{C3}}{K} \quad (2.20)$$

$$K = \alpha_A C_{A3} \frac{MW_A}{\rho} + \alpha_B C_{B3} \frac{MW_B}{\rho} + \alpha_C C_{C3} \frac{MW_C}{\rho} + \alpha_D x_D \rho$$

where x_D is the mass fraction of the solvent in the flash tank liquid holdup and is found from a mass balance.

The system of Eq. 2.19 is modeled with sensor measurement noise and Gaussian process noise. The sensor measurement noise is generated using a zero-mean normal distribution with a standard deviation of 10^{-1} for the three temperature states and 10^{-2} for the nine concentration states. Noise is applied to each continuous measurement of temperatures and concentrations with a frequency of $\Delta_m = 0.001 \text{ hr}$. The process noise is generated similarly, with a zero-mean normal distribution and the same standard deviation values. Process noise is added to the right-hand side of the ODEs in the system of Eq. 2.19 and changes with a frequency of $\Delta_p = 0.001 \text{ hr}$. In all three vessels, the heat input/removal is a manipulated variable for controlling the reactors at the appropriate operating

Table 2.2: Parameter values

$T_{10} = 300, T_{20} = 300$	K
$F_{10} = 5, F_{20} = 5, F_r = 1.9$	$\frac{m^3}{hr}$
$C_{A10} = 4, C_{A20} = 3$	$\frac{kmol}{m^3}$
$V_1 = 1.0, V_2 = 0.5, V_3 = 1.0$	m^3
$E_1 = 5E4, E_2 = 5.5E4$	$\frac{kJ}{kmol}$
$k_1 = 3E6, k_2 = 3E6$	$\frac{1}{hr}$
$\Delta H_1 = -5E4, \Delta H_2 = -5.3E4$	$\frac{kJ}{kmol}$
$H_{vap} = 5$	$\frac{kJ}{kmol}$
$C_p = 0.231$	$\frac{kJ}{kgK}$
$R = 8.314$	$\frac{kJ}{kmolK}$
$\rho = 1000$	$\frac{kg}{m^3}$
$\alpha_A = 2, \alpha_B = 1, \alpha_C = 1.5, \alpha_D = 3$	unit-less
$MW_A = 50, MW_B = 50, MW_C = 50$	$\frac{kg}{kmol}$

temperature. In addition the second tank's inlet flow rate is used as a manipulated variable. The system has one unstable and two stable steady states. The desired operating steady state is the unstable steady state:

$$\begin{aligned}
 x_{uss} &= [T_1 \ C_{A1} \ C_{B1} \ C_{C1} \ T_2 \ C_{A2} \ C_{B2} \ C_{C2} \ T_3 \ C_{A3} \ C_{B3} \ C_{C3}]^T \\
 &= [370 \ 3.32 \ 0.17 \ 0.04 \ 435 \ 2.75 \ 0.45 \ 0.11 \ 435 \ 2.88 \ 0.50 \ 0.12]^T
 \end{aligned}$$

The first set of manipulated inputs is the heats injected to or removed from the three vessels, that is $u_1 = [Q_1 \ Q_2 \ Q_3]^T$; the second set of manipulated input is the the inlet flow rate to vessel 2, that is $u_2 = \Delta F_{20} = F_{20} - F_{20s}$. The control variables are deviation variables, whose values at the desired steady state are zero and subject to the constraints $|Q_i| \leq 10^6 \text{ KJ/hr}$, ($i = 1, 2, 3$) and $|\Delta F_{20}| \leq 5 \text{ m}^3/\text{hr}$.

We consider a quadratic Lyapunov function $V(x) = x^T P x$ with

$$P = \text{diag}^{\S}([20 \ 10^3 \ 10^3 \ 10^3 \ 10 \ 10^3 \ 10^3 \ 10^3 \ 10 \ 10^3 \ 10^3 \ 10^3])$$

^{\S} $\text{diag}(v)$ denotes a matrix with its diagonal elements being the elements of vector v and all the other elements being zeros.

and design the controller $h(x)$ as three PI controllers with proportional gains $K_{p1} = K_{p2} = K_{p3} = 8000$ and integral time constants $\tau_{I1} = \tau_{I2} = \tau_{I3} = 10$ based on the measurements of T_1 , T_2 and T_3 , respectively. Note that, in the absence of process and measurement noise, this design of $h(x)$ manipulating $u_1 = [Q_1 \ Q_2 \ Q_3]$ can stabilize the closed-loop system asymptotically without the help of u_2 . Based on $h(x)$ and $V(x)$, we design LMPC 1 to determine u_1 and LMPC 2 to determine u_2 following the forms given in Eqs. 2.6 and 2.5, respectively. This control design is the fault-free control configuration for the closed-loop system. In the design of the LMPC controllers, the weighting matrices are chosen to be $Q_c = \text{diag}([20 \ 10^3 \ 10^3 \ 10^3 \ 10 \ 10^3 \ 10^3 \ 10^3 \ 10 \ 10^3 \ 10^3 \ 10^3])$, $R_1 = \text{diag}([(5 \ 5 \ 5) \cdot 10^{-12}])$ and $R_2 = 100$. The horizon for the optimization problem is $N = 5$ with a time step of $\Delta = 0.01 \text{ hr}$.

In addition, the control input u_1 can be divided into two sets, $u_{11} = [Q_1 \ Q_3]^T$ and $u_{12} = Q_2$. The input combination u_{11} and u_2 is able to stabilize the closed-loop system which can be used as the input configuration of the backup DMPC system of Eqs. 2.10-2.11. In order to design the backup DMPC, we need to design a second Lyapunov-based controller $h_2(x)$ which manipulates Q_1 , Q_3 and ΔF_{20} . We also design h_2 through PI control law with proportional gains $K_{p1}^b = K_{p2}^b = 8000$, $K_{p3}^b = -0.3$ and integral time constants $\tau_{I1}^b = \tau_{I2}^b = \tau_{I3}^b = 10$ based on the measurements of T_1 , T_3 and T_2 , respectively. The control design h_2 can stabilize the closed-loop system asymptotically with $Q_2 = 0$ in the absence of process and measurement noise. In the design of the backup DMPC system, we choose $V_2(x) = V(x)$. The backup DMPC system is the backup control configuration when there is a fault in the actuators associated with u_{12} .

In order to perform FDI for the reactor-separator system, we construct the FDI filters for the states affected directly by the four manipulated inputs as in Eq. 2.13. The states affected directly by the manipulated inputs are T_1 , C_{A2} , C_{B2} , C_{C2} , T_2 and T_3 . In addition, the FDI residuals take the following form:

$$\begin{aligned} r_{T_i}(t) &= |\hat{T}_i(t) - T_i(t)|, \quad i = 1, 2, 3 \\ r_{C_{i2}}(t) &= |\hat{C}_{i2}(t) - C_{i2}(t)|, \quad i = A, B, C. \end{aligned} \tag{2.21}$$

The threshold values used for each residual in the numerical simulations are as follows:

$$\begin{aligned} r_{T_i, \max} &= 5 \ K, \quad i = 1, 2, 3 \\ r_{C_{i2}, \max} &= 0.08 \text{ kmol/m}^3, \quad i = A, B, C. \end{aligned}$$

If a fault affects more than one state directly, more than one residual will be nonzero. However, because of the process dynamics and threshold values, the residuals will not exceed the thresholds at the same time. In order to avoid false isolation of faults, we also use a fault isolation waiting time. That is, when the FDI system detects a fault at t_f , it will not isolate the fault until t_w time later. This waiting time t_w is chosen to make sure that different faults have different fault signatures to avoid false isolation but should also be sensitive enough to isolate faults in a timely manner. The waiting time t_w used in the simulations is $t_w = 0.015hr$.

We consider two different faults in the following simulations. First, we will consider a fault in the heat input/removal actuator to vessel 2, that is a fault in Q_2 . Because Q_2 only affects directly the state T_2 and all the measurements are continuously available, when there is an actuator fault in Q_2 , only the residual corresponding to T_2 will exceed its threshold. The second fault we will consider is a fault in the inlet flow actuator to vessel 2, that is a fault in ΔF_{20} . Because the control action ΔF_{20} affects directly the states T_2 , C_{A2} , C_{B2} and C_{C2} , when there is an actuator fault in F_{20} , more than one residuals will exceed their thresholds. Note that the thresholds and waiting time have been chosen in a way that we can distinguish between faults in Q_2 and F_{20} correctly.

2.4.2 Simulation Results

In the following simulations, the plant is initialized at the target steady state (x_{uss}) and simulated up to $t = 1.0 hr$ with a fault being triggered at time $t = 0.2 hr$ for all the simulations. The measurement and process noise bounds used were $w_m = 0.1w_p$ with

$$w_p = [2.5 \ 0.25 \ 0.25 \ 0.25 \ 2.5 \ 0.25 \ 0.25 \ 0.25 \ 2.5 \ 0.25 \ 0.25 \ 0.25]$$

First, we consider the fault in the heat input/removal actuator to vessel 2 which renders $Q_2 = -10^6 \text{ KJ/hr}$. Figures 2.3 and 2.4 show the temperature and concentration profiles for each vessel when all controlled actuators are completely functional up to time $t = 0.2 hr$, using $u_1 = [Q_1 \ Q_2 \ Q_3]^T$ and $u_2 = \Delta F_{20}$. The dotted lines in the figures represent the target steady-state values. In this example, no FTC is implemented and at time $t = 0.2 hr$ a fault is triggered and the fault is detected at time $t = 0.201 hr$ and correctly isolated to a fault in Q_2 at time $t = 0.216 hr$; we see that the control system cannot stabilize the process at the unstable steady state. Figure 2.5 shows the corresponding residuals with no FTC.

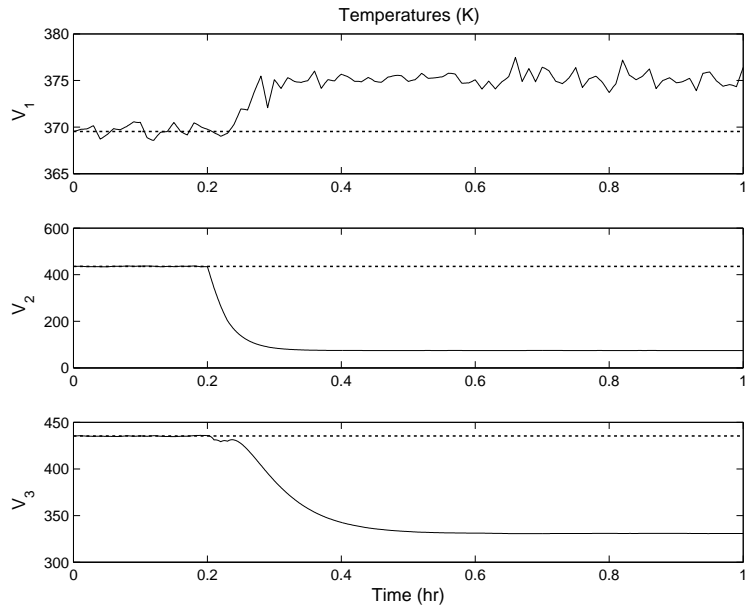


Figure 2.3: Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. No FTC is implemented.

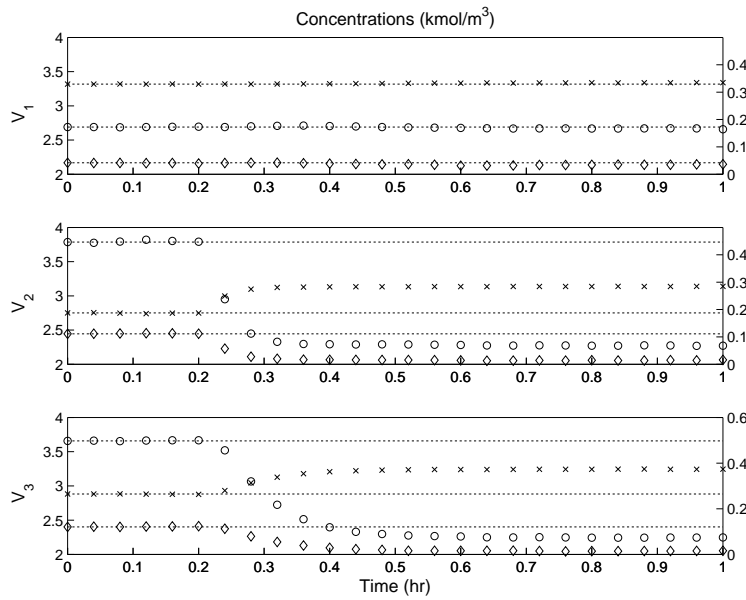


Figure 2.4: Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. No FTC is implemented.

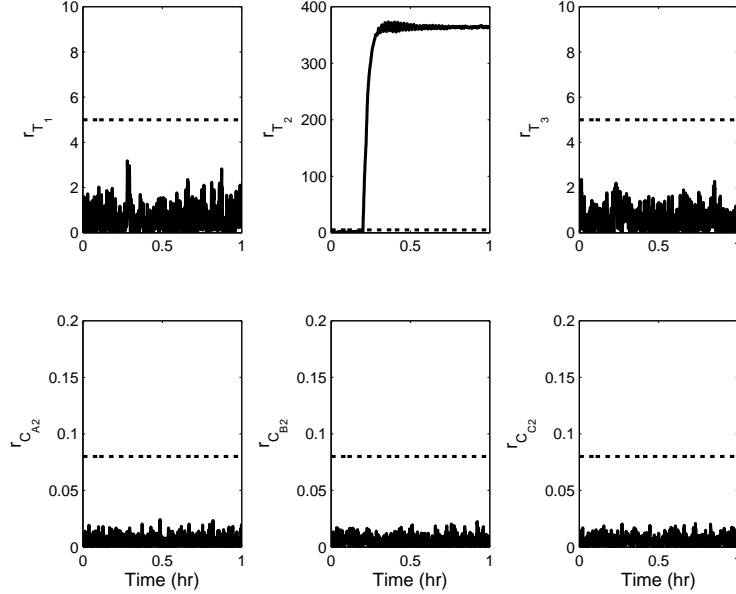


Figure 2.5: FDI filter residuals for temperatures (T_1 , T_2 , T_3) and concentration (C_{A2} , C_{B2} , C_{C2}) with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. No FTC is implemented.

Similar to the above scenario, the same simulation is considered, but upon isolation of the fault in Q_2 , the control system is reconfigured as follows: LMPC 1 is updated to only optimize $u_{11} = [Q_1 \ Q_3]^T$ (i.e., from $u_1 = u_1^L$ to $u_1 = [u_{11}^b \ 0]^T$) and shut down the input for Q_2 (i.e., $u_{12} = Q_2 = 0$) while maintaining the identical LMPC 2 for u_2 (i.e., $u_2 = u_2^L$). This reconfiguration implies that only the FTC switching rule of Eq. 2.15a is implemented and LMPC 2 for u_2 is operating on the assumption that LMPC 1 is using all three heat input/removal actuators. The temperature and concentration profiles of the closed-loop system under this reconfiguration are shown in Figs. 2.6 and 2.7. From Figs. 2.6 and 2.7, we see that the system cannot be stabilized using only the switching rule of Eq. 2.15a. As shown in Fig. 2.8, using only the FTC switching rule of Eq. 2.15a, the control action for $u_2 = \Delta F_{20}$ is not as large as required for stabilization since u_2 expects the control action of Q_2 to help stabilize the system; please also see Fig. 2.12 for the profile of u_2 when the complete FTC switching rule of Eq. 2.15 is implemented for comparison.

The next setup is identical to the conditions tested previously, where we consider a fault in Q_2 , but the FTC will now use the complete switching rule of Eq. 2.15 where the LMPC control law of u_2 is also updated to account for the complimentary controller $u_{11} = [Q_1 \ Q_3]^T$ controlling only

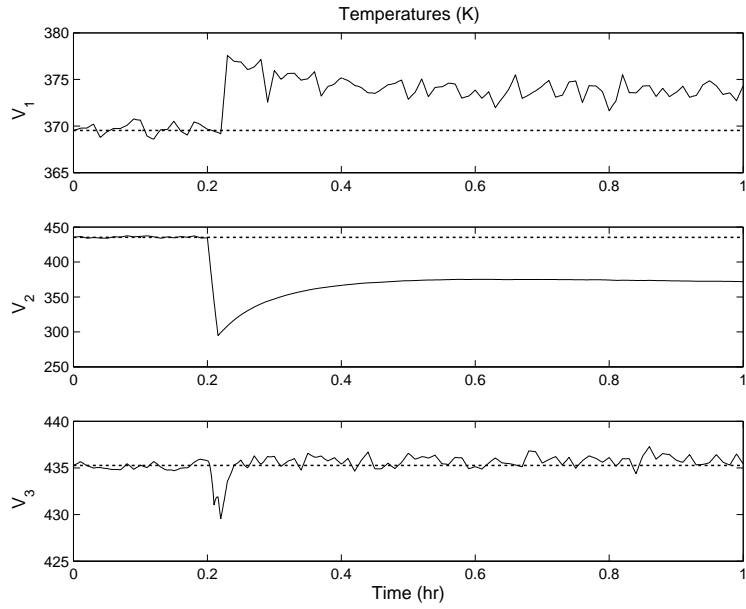


Figure 2.6: Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. The FTC switching rule of Eq. 2.15a is implemented.

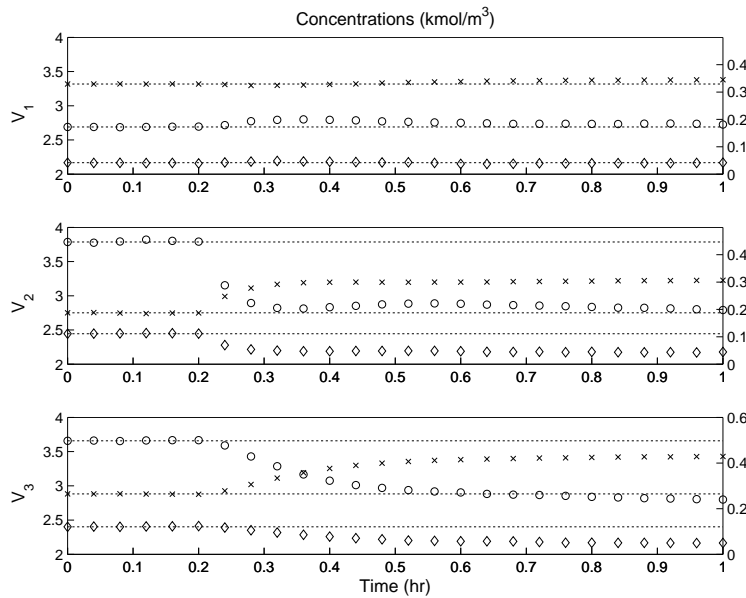


Figure 2.7: Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. The FTC switching rule of Eq. 2.15a is implemented, but cannot stabilize T_2 and T_3 to the desired steady-state.

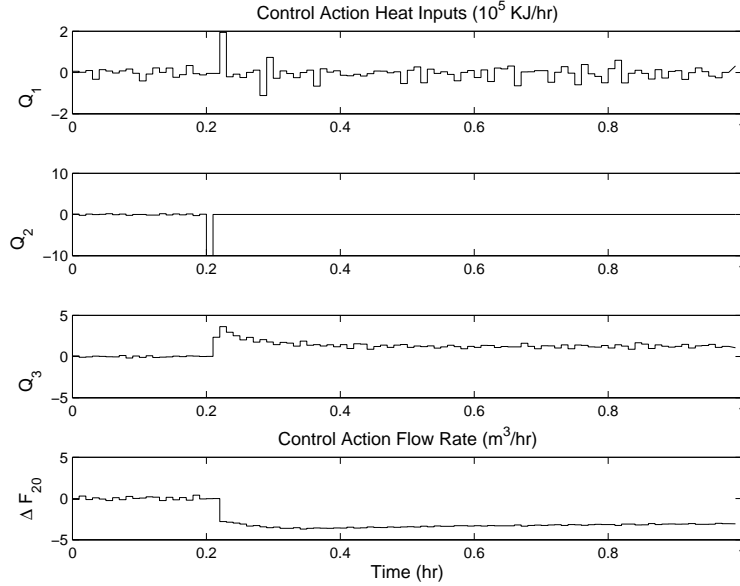


Figure 2.8: Control actions with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. The FTC switching rule of Eq. 2.15a is implemented.

two heat input/removal actuators. Figures 2.9 and 2.10 show the temperature and concentration profiles for each vessel when the fault in Q_2 is triggered at $t = 0.2 \text{ hr}$ and FTC is carried out when the fault is isolated. Figure 2.11 shows the corresponding residuals, where the fault is detected at time $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. We see from these figures that when there is a fault in Q_2 , the state of the closed-loop system deviates from the required steady state, and upon isolation of the fault, the FTC switching rule of Eq. 2.15 is carried out and the reconfigured DMPC is able to drive the state of the system back to the desired steady state. The temperature and concentration trajectories return near the steady state at $t = 0.60 \text{ hr}$ and then minimal control action is required to further maintain system stability. The reconfiguration of u_2 allows the system to be stabilized with an appropriately strong control action from u_2^b . The difference in control action can clearly be seen by comparing Fig. 2.12, where both u_1 and u_2 controllers are reconfigured, to Fig. 2.8, where only u_1 is reconfigured.

Next, we consider the fault in the inlet flow control actuator of vessel 2, F_{20} which renders $\Delta F_{20} = 5 \text{ m}^3/\text{hr}$. Figures 2.13 and 2.14 show the temperature and concentration profiles for each vessel when the fault in F_{20} is triggered at $t = 0.2 \text{ hr}$ and no FTC is implemented; we see that

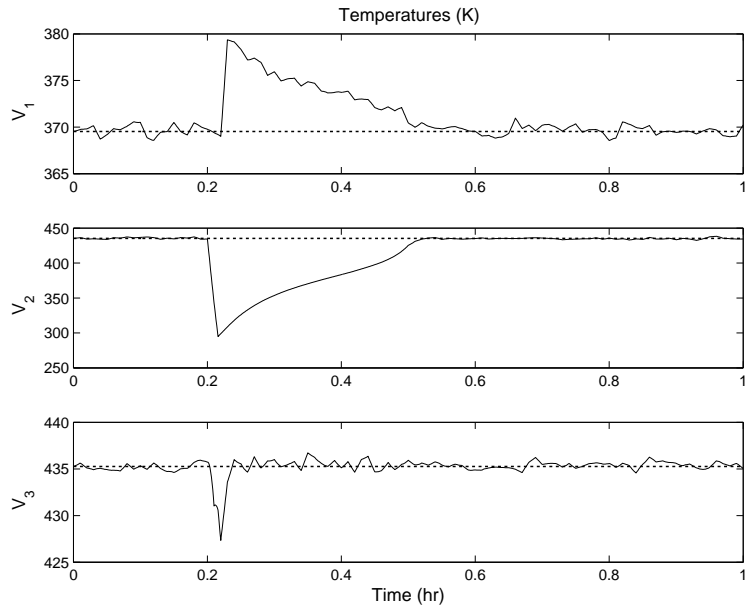


Figure 2.9: Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. The FTC switching rule of Eq. 2.15 is implemented.

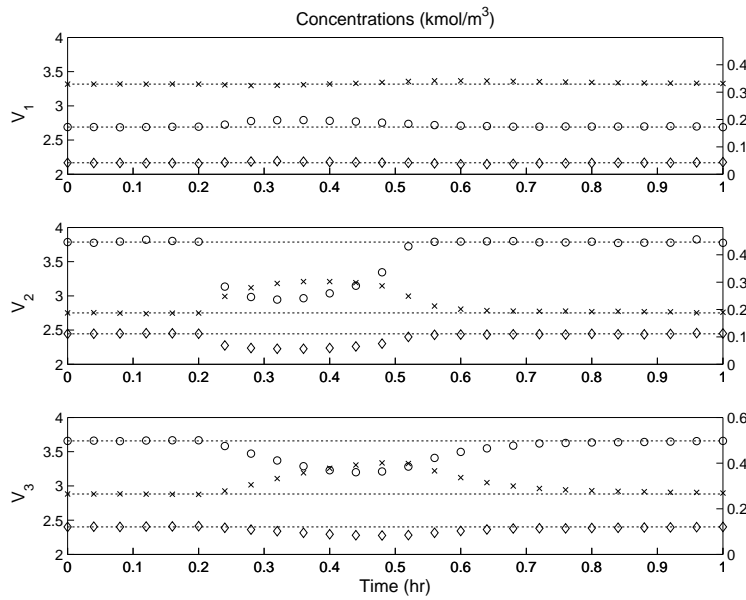


Figure 2.10: Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.201 \text{ hr}$ and isolated at $t = 0.216 \text{ hr}$. The FTC switching rule of Eq. 2.15 is implemented.

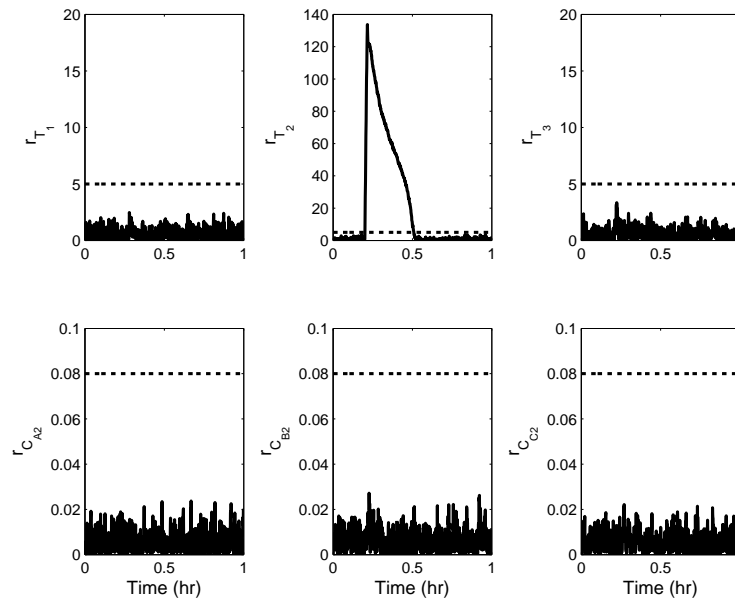


Figure 2.11: FDI filter residuals for temperatures (T_1 , T_2 , T_3) and concentration (C_{A2} , C_{B2} , C_{C2}) with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15 is implemented.

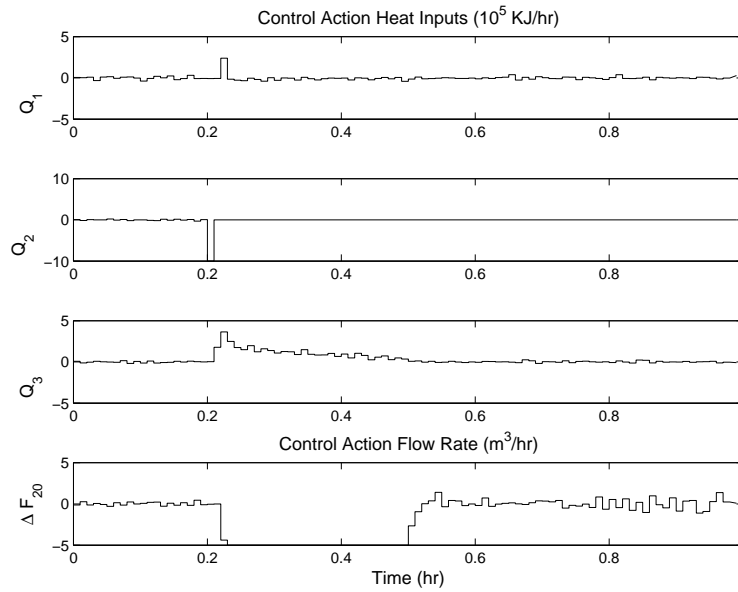


Figure 2.12: Control actions with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.201$ hr and isolated at $t = 0.216$ hr. The FTC switching rule of Eq. 2.15 is implemented.

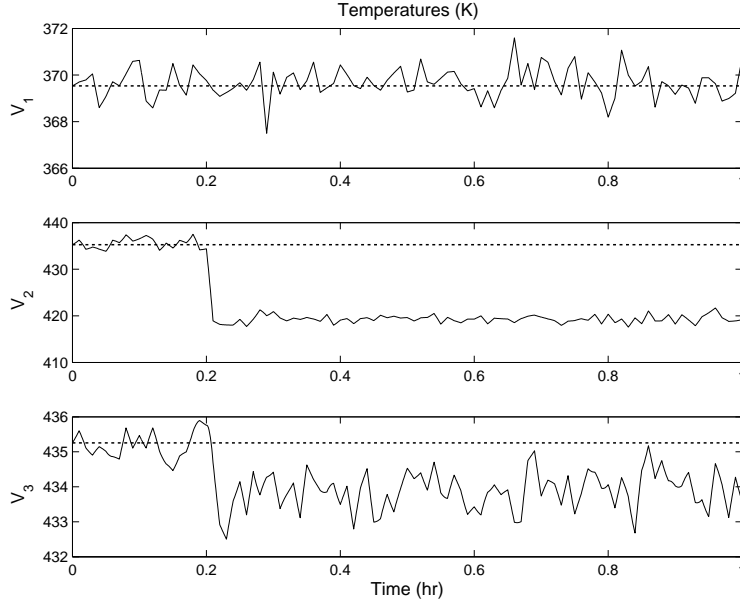


Figure 2.13: Temperature profiles for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.204 \text{ hr}$ and isolated at $t = 0.219 \text{ hr}$. No FTC is implemented.

the control system cannot stabilize the process at the desired steady state. Figure 2.15 shows the corresponding residuals, from which we see that the fault is detected at $t = 0.204 \text{ hr}$ when the residual of T_2 exceeds its threshold and the fault can be isolated at $t = 0.219 \text{ hr}$ when the residuals corresponding to T_2 and C_{B_2} exceed their thresholds, respectively.

In the last simulation scenario, we consider the same fault in F_{20} , but upon detection at $t = 0.204 \text{ hr}$ and isolation of the fault at $t = 0.219 \text{ hr}$, we carry out the switching rule of Eq. 2.14 and the input F_{20} is shut down and separated from the plant. In this particular example the FTC system only reconfigures one controller by switching off the u_2 controller and resetting $u_2 = \Delta F_{20} = 0$, while maintaining the LMPC controller u_1 the same. We know from Section 2.3.2 that u_1 with $u_2 = 0$ can asymptotically stabilize the trajectories toward the set-point. Figures 2.16 and 2.17 show the temperature and concentration profiles for each vessel when the F_{20} fault is triggered at $t = 0.2 \text{ hr}$ and FTC is carried out after the waiting time t_w . We see from these figures that when there is a fault in F_{20} , the state of the closed-loop system deviates from the desired steady state, and upon the isolation of the fault, the FTC switching rule of Eq. 2.14 is carried out and the reconfigured DMPC is able to drive the state of the closed-loop system back to the desired steady state. The corresponding residuals are shown in Fig. 2.18, where we see that an actuator fault in

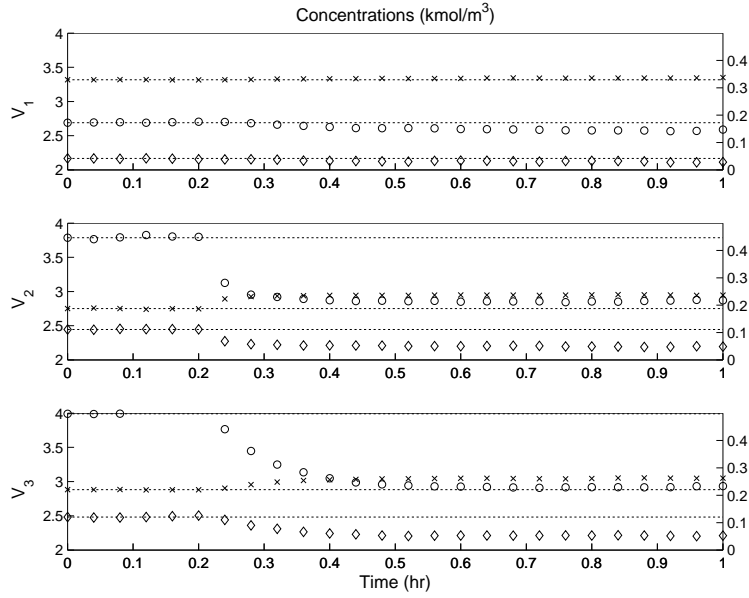


Figure 2.14: Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.204 \text{ hr}$ and isolated at $t = 0.219 \text{ hr}$. No FTC is implemented.

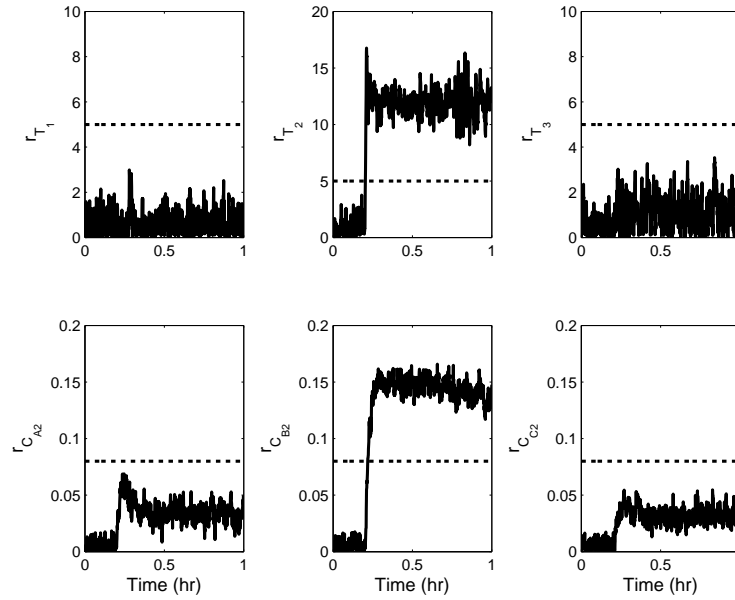


Figure 2.15: FDI filter residuals for temperatures (T_1 , T_2 , T_3) and concentration (C_{A2} , C_{B2} , C_{C2}) with a fault in the inlet flow actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.204 \text{ hr}$ and isolated at $t = 0.219 \text{ hr}$. No FTC is implemented.

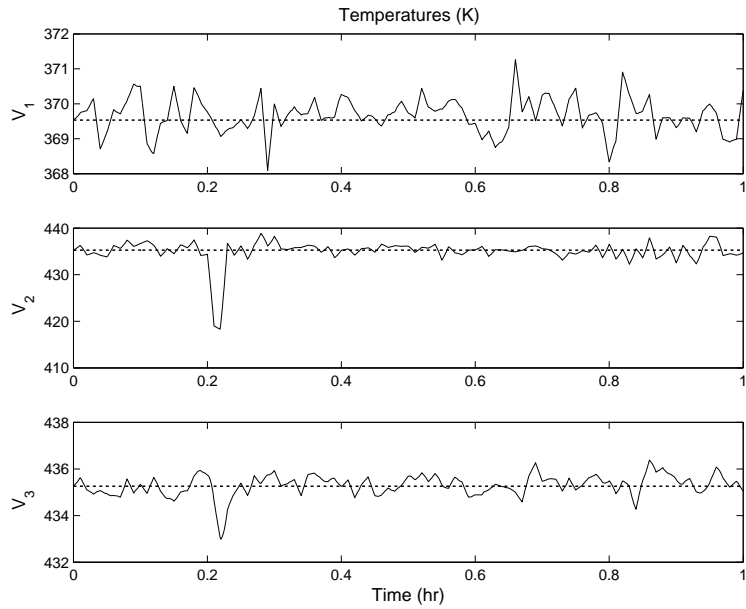


Figure 2.16: Temperature profiles for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.204 \text{ hr}$ and isolated at $t = 0.219 \text{ hr}$. The FTC switching rule of Eq. 2.14 is implemented.

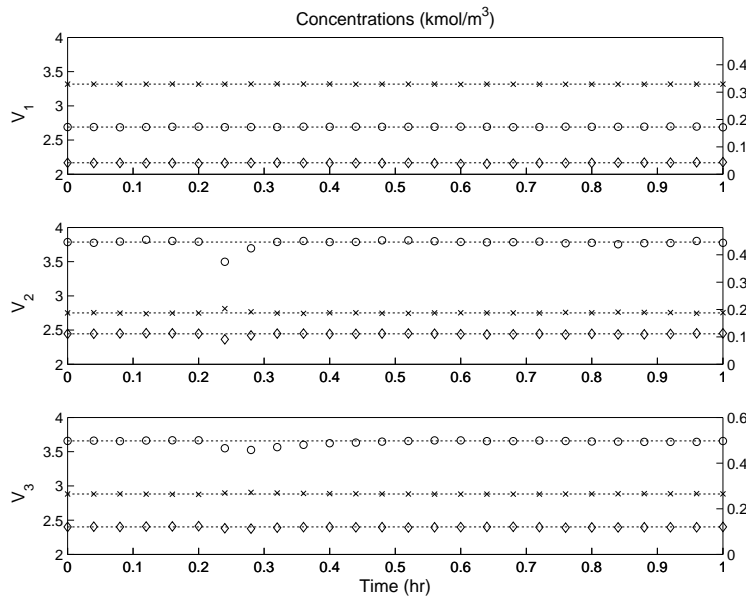


Figure 2.17: Concentration profiles ($C_A = x$, $C_B = o$, $C_C = \diamond$) for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2 \text{ hr}$. Fault is detected at $t = 0.204 \text{ hr}$ and isolated at $t = 0.219 \text{ hr}$. The FTC switching rule of Eq. 2.14 is implemented.

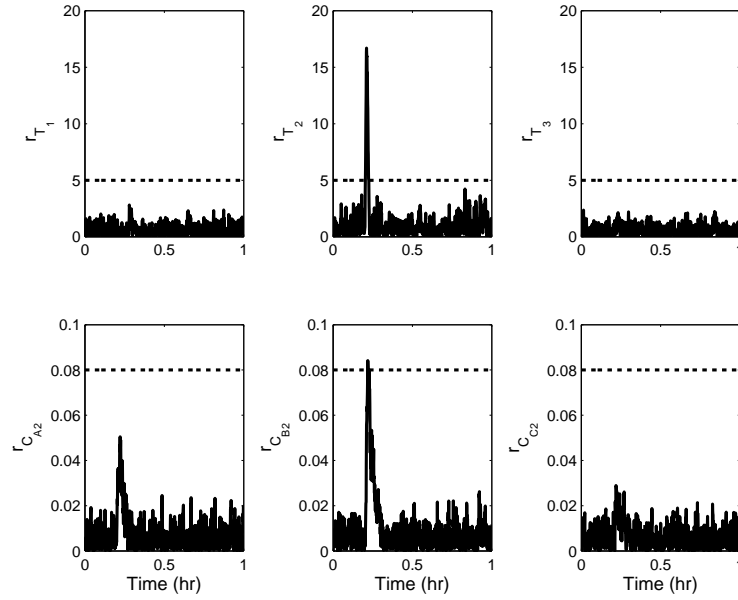


Figure 2.18: FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ hr. Fault is detected at $t = 0.204$ hr and isolated at $t = 0.219$ hr. The FTC switching rule of Eq. 2.14 is implemented.

F_{20} significantly affects the residuals corresponding to T_2 and C_{B2} .

2.5 Conclusions

In this chapter, a model-based FDI and FTC system was designed for the monitoring and reconfiguration of a DMPC system applied to general nonlinear processes in the presence of control actuator faults, and specific FTC switching rules were developed to guide the control system reconfiguration. The applicability and effectiveness of the proposed design was demonstrated via a chemical process example which consists of two CSTRs and a flash tank separator with a recycle stream.

Chapter 3

Data-Based Fault Detection and Isolation Using Adaptive Isolation Windows

3.1 Introduction

In this chapter, we develop a data-based monitoring and reconfiguration system for a distributed model predictive control system in the presence of control actuator faults. Specifically, we first design fault detection filters and filter residuals, which are computed via exponentially weighted moving average, to effectively detect faults. Then, we propose a fault isolation approach which uses adaptive fault isolation time windows whose length depends on the rate of change of the fault residuals to quickly and accurately isolate actuator faults. Simultaneously, we estimate the magnitudes of the faults using a least-squares method and based on the estimated fault values, we design appropriate control system reconfiguration (fault-tolerant control) strategies to handle the actuator faults and maintain the closed-loop system state within a desired operating region. A nonlinear chemical process example is used to demonstrate the approach.

3.2 Notation

The operator $|\cdot|$ is used to denote the absolute value of a scalar and the operator $\|\cdot\|$ is used to denote Euclidean norm of a vector, while $\|\cdot\|_Q$ refers to the square of the weighted Euclidean norm, defined by $\|x\|_Q = x^T Q x$ for all $x \in R^n$. The symbol $diag(v)$ denotes a square diagonal matrix whose diagonal elements are the elements of the vector v .

3.3 Problem formulation and preliminaries

3.3.1 Class of nonlinear systems

We consider nonlinear processes described by the following state-space model:

$$\dot{x}(t) = f(x) + \sum_{i=1}^2 g_i(x)(u_i(t) + \tilde{u}_i(t)) \quad (3.1)$$

where $x \in R^n$ denotes the set of state variables, and $u_1 \in R^{m_1}$ and $u_2 \in R^{m_2}$ denote two sets of manipulated inputs, and $\tilde{u}_1 \in R^{m_1}$ and $\tilde{u}_2 \in R^{m_2}$ denote the unknown fault vectors for u_1 and u_2 , respectively. We consider that $u_1 + \tilde{u}_1$ and $u_2 + \tilde{u}_2$ take values in non-empty convex sets $U_1 \in R^{m_1}$ and $U_2 \in R^{m_2}$, respectively. The convex sets U_1 and U_2 are defined as follows:

$$\begin{aligned} U_1 &= \{u_1 + \tilde{u}_1 \in R^{m_1} : \|u_1 + \tilde{u}_1\| \leq u_1^{\max}\} \\ U_2 &= \{u_2 + \tilde{u}_2 \in R^{m_2} : \|u_2 + \tilde{u}_2\| \leq u_2^{\max}\} \end{aligned}$$

where u_1^{\max} and u_2^{\max} are the magnitudes of the input constraints. The system of Eq. 3.1 can be re-written in a compact form as follows:

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

where $g(x) = [g_1(x) \ g_2(x)]$, $u(t) = [u_1(t)^T \ u_2(t)^T]^T$ and $\tilde{u}(t) = [\tilde{u}_1(t)^T \ \tilde{u}_2(t)^T]^T$. We also assume that U is a suitable composition of U_1 and U_2 such that $u + \tilde{u} \in U$ is equivalent to $u_1 + \tilde{u}_1 \in U_1$ and $u_2 + \tilde{u}_2 \in U_2$.

We use the variable $\tilde{u}_{f,j}$, $j = 1, \dots, m_1 + m_2$, to model the possible faults associated with the j^{th}

element in the manipulated input vector u . Under fault-free operating conditions, we have $\tilde{u} = 0$, and hence, $\tilde{u}_{f,j} = 0$ for all $j = 1, \dots, m_1 + m_2$. When fault j occurs, $\tilde{u}_{f,j}$ takes a non-zero value. We assume that f and g are locally Lipschitz vector functions and that $f(0) = 0$. This means that the origin is an equilibrium point for the fault-free system ($\tilde{u} = 0$ for all t) with $u = 0$. We also assume that the state x of the system is available synchronously and continuously at each sampling time.

Remark 3.1. *Note that the use of two sets of manipulated inputs is adopted because of the implementation of a distributed model predictive control system to regulate the process; please see Subsection 3.3.2 for the control system design. An example of a chemical process system described by Eq. 3.1 is given in Section 3.5.2*

3.3.2 Fault-free control system design

We assume that there exists a nonlinear control law $h(x)$ which determines u_1 (i.e., $u_1(t) = h(x(t))$) and renders the origin of the fault-free closed-loop system asymptotically stable with $u_2(t) = 0$. This assumption is essentially a standard stabilizability requirement made in all linear/nonlinear control methods and implies that there exists a Lyapunov function $V(x)$ of the system whose time derivative is always negative when $u_1 = h(x)$ is applied to the fault-free closed-loop system [23, 8].

We adopt the DMPC architecture introduced in [25] to design the fault-free control system. In this DMPC architecture, one LMPC is designed to determine u_1 and is responsible for the closed-loop stability; and another LMPC is designed to compute u_2 and to coordinate with u_1 to improve the closed-loop performance. We will refer to the two LMPCs computing u_1 and u_2 as LMPC 1 and LMPC 2, respectively. The two LMPCs are evaluated in a sequential fashion (i.e., LMPC 2 is first evaluated and then LMPC 1 is evaluated) at discrete time instants $\{t_{k \geq 0}\}$ with $t_k = t_0 + k\Delta$, $k = 0, 1, \dots$ where t_0 is the initial time and Δ is a sampling time.

Specifically, the optimization problem of LMPC 2 at time t_k depends on the state measurement

$x(t_k)$ and is formulated as follows:

$$\min_{u_2 \in S(\Delta)} \int_{t_k}^{t_k+N} L(\tilde{x}(\tau), u_1(\tau), u_2(\tau)) d\tau \quad (3.2a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + \sum_{i=1}^2 g_i(\tilde{x}(t)) u_i(t) \quad (3.2b)$$

$$\tilde{x}(t_k) = x(t_k) \quad (3.2c)$$

$$u_1(t) = h(\tilde{x}(t_{k+j})), \forall t \in [t_{k+j}, t_{k+j+1}) \quad (3.2d)$$

$$u_2(t) \in U_2 \quad (3.2e)$$

$$\frac{\partial V(x)}{\partial x} g_2(x(t_k)) u_2(t_k) \leq 0 \quad (3.2f)$$

with $L(\tilde{x}, u_1, u_2) = \|\tilde{x}(\tau)\|_{Q_c} + \|u_1(\tau)\|_{R_{c1}} + \|u_2(\tau)\|_{R_{c2}}$ where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ , N is the prediction horizon, Q_c , R_{c1} and R_{c2} are positive definite weighting matrices, $j = 0, \dots, N-1$, \tilde{x} is the predicted trajectory of the fault-free system with u_2 being the input trajectory computed by LMPC 2 of Eq. 3.2 and u_1 being the nonlinear controller $h(x)$ applied in a sample-and-hold fashion. The optimal solution to this optimization problem is denoted $u_2^*(t|t_k)$. This information is sent to LMPC 1. The optimization problem of LMPC 1 depends on $x(t_k)$ and the decision made by LMPC 2 (i.e., $u_2^*(t|t_k)$). Specifically, LMPC 1 is based on the following optimization problem:

$$\min_{u_1 \in S(\Delta)} \int_{t_k}^{t_k+N} L(\tilde{x}(\tau), u_1(\tau), u_2(\tau)) d\tau \quad (3.3a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + \sum_{i=1}^2 g_i(\tilde{x}(t)) u_i(t) \quad (3.3b)$$

$$\tilde{x}(t) = x(t_k) \quad (3.3c)$$

$$u_1(t) \in U_1 \quad (3.3d)$$

$$u_2(t) = u_2^*(t|t_k) \quad (3.3e)$$

$$\frac{\partial V(x)}{\partial x} g_1(x(t_k)) u_1(t_k) \leq \frac{\partial V(x)}{\partial x} g_1(x(t_k)) h(x(t_k)). \quad (3.3f)$$

The optimal solution to this optimization problem is denoted by $u_1^*(t|t_k)$.

Once both optimization problems are solved, the manipulated inputs of the DMPC system

based on LMPC 1 and LMPC 2 are defined as follows:

$$\begin{aligned} u_1^L(t) &= u_1^*(t|t_k), \forall t \in [t_k, t_{k+1}) \\ u_2^L(t) &= u_2^*(t|t_k), \forall t \in [t_k, t_{k+1}). \end{aligned}$$

The fault-free closed-loop system of Eq. 3.1 under this DMPC scheme with inputs defined by $u_1 = u_1^L$ and $u_2 = u_2^L$ maintains practical stability because of the two Lyapunov-based constraints of Eqs. 3.2f and 3.3f [25].

3.3.3 FTC considerations

The presence of the control action u_2 brings extra control flexibility to the closed-loop system which can be used to carry out FTC. Specifically, we further assume that the control input u_1 can be decomposed into two subsets (i.e., $u_1 = [u_{11}^T \ u_{12}^T]^T$) and that there exists a nonlinear control law $h_2(x) = [h_{21}(x)^T \ h_{22}(x)^T]^T$ which determines u_{11} and u_2 (i.e., $u_{11} = h_{21}(x)$ and $u_2 = h_{22}(x)$) and is able to asymptotically stabilize the fault-free closed-loop system with $u_{12} = 0$. This assumption implies that there exist a Lyapunov function $V_2(x)$ of the system whose time derivative is always negative when $u_{11} = h_{21}(x)$, $u_{12} = 0$ and $u_2 = h_{22}(x)$ are applied.

Based on $h_2(x)$, we can design a backup DMPC system (i.e., DMPC with LMPC 1 of Eq. 3.5 and LMPC 2 of Eq. 3.4 below) to manipulate u_{11} and u_2 to stabilize the closed-loop system following the results developed in [24]. We still design two LMPC controllers in the backup DMPC system. One LMPC is used to manipulate u_{11} and the other one is used to manipulate u_2 . In this backup DMPC system, the two LMPCs coordinate their actions to maintain the closed-loop stability. We refer to the LMPC manipulating u_{11} as the backup LMPC 1 and the LMPC manipulating u_2 as the backup LMPC 2. The two backup LMPCs are also evaluated in sequence.

The backup LMPC 2 optimizes u_2 and is designed as follows:

$$\min_{u_2 \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(\tau), u_1(\tau), u_2(\tau)) d\tau \quad (3.4a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g_1(\tilde{x}(t))[u_{11}(t)^T u_{12}(t)^T]^T + g_2(\tilde{x}(t))u_2(t) \quad (3.4b)$$

$$\tilde{x}(t_k) = x(t_k) \quad (3.4c)$$

$$u_{11}(t) = h_{21}(\tilde{x}(t_{k+j})), \forall t \in [t_{k+j}, t_{k+j+1}) \quad (3.4d)$$

$$u_{12}(t) = 0 \quad (3.4e)$$

$$u_2(t) \in U_2 \quad (3.4f)$$

$$\frac{\partial V_2(x)}{\partial x} g_2(x(t_k)) u_2(t_k) \leq \frac{\partial V_2(x)}{\partial x} g_2(x(t_k)) h_{22}(x(t_k)). \quad (3.4g)$$

The solution to the optimization problem of Eq. 3.4 is denoted $u_2^{b,*}(t|t_k)$. The backup LMPC 1 optimizes u_{11} and is designed as follows:

$$\min_{u_{11} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(\tau), u_1(\tau), u_2(\tau)) d\tau \quad (3.5a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g_1(\tilde{x}(t))[u_{11}(t)^T u_{12}(t)^T]^T + g_2(\tilde{x}(t))u_2(t) \quad (3.5b)$$

$$\tilde{x}(t_k) = x(t_k) \quad (3.5c)$$

$$u_{11}(t) \in U_1 \quad (3.5d)$$

$$u_{12}(t) = 0 \quad (3.5e)$$

$$u_2 = u_2^{b,*}(t|t_k) \quad (3.5f)$$

$$\frac{\partial V_2(x)}{\partial x} g_1(x(t_k))[u_{11}(t)^T \ 0^T]^T \leq \frac{\partial V_2(x)}{\partial x} g_1(x(t_k))[h_{21}(x(t_k))^T \ 0^T]^T. \quad (3.5g)$$

The solution to the optimization problem of Eq. 3.5 is denoted $u_{11}^{b,*}(t|t_k)$. The control inputs of the backup DMPC are defined as follows:

$$u_{11}^b(t) = u_{11}^{b,*}(t|t_k), \forall t \in [t_k, t_{k+1})$$

$$u_{12}^b(t) = 0, \forall t$$

$$u_2^b(t) = u_2^{b,*}(t|t_k), \forall t \in [t_k, t_{k+1})$$

The fault-free closed-loop system of Eq. 3.1 under the backup DMPC control with inputs defined

by $u_{11} = u_{11}^b$, $u_{12} = 0$ and $u_2 = u_2^b$ maintains practical stability of the closed-loop system because of the Lyapunov-based constraints of Eqs. 3.4g and 3.5g [24].

To present the proposed method, in this work, we consider control actuator faults that can be detected by appropriate nonlinear dynamic fault filters via observing the evolution of the closed-loop system state. In order to isolate the occurrence of a fault, it is further required to assume that the control actuator in question is the only one influencing the observed “faulty” states (i.e., each fault has a unique fault signature). For more discussions on systems having verifiable isolable structures, please see [34, 36].

3.4 FDI and FTC system design

In this section, we develop a combined model-based and data-based FDI and FTC method for the closed-loop system of Eq. 3.1 under the DMPC of Eqs. 3.2-3.3.

3.4.1 Design of fault detection filters and residuals

The fault detection and isolation filter is designed on the basis of the process model of Eq. 3.1 under the DMPC system of Eqs. 3.2-3.3 used to control the process and it is used to predict the expected process dynamic state response in the absence of faults. These expected state values are compared with the corresponding real-time measured process states, forming the residuals (i.e., $r_p(t)$ defined below). Then, the residuals are compared with threshold values computed from closed-loop data under normal operation and a fault is declared (detected) when the residual values exceed the thresholds. Fault isolation subsequently is carried out by comparing the fault signature (i.e., what residuals exceed their thresholds) with the signature of process/fault interaction of the various explicitly-modeled faults computed from the process model. The DMPC system of Eqs. 3.2-3.3 is the control system for the fault-free closed-loop system. We first design an FDI scheme to detect faults in this control system. In this FDI scheme, a filter is designed for each state and the design of the filter for the p^{th} , $p = 1, \dots, n$, state in the system state vector x is as follows [34]:

$$\hat{x}_p(t) = f_p(X_p) + g_{1p}(X_p)u_1^L(t) + g_{2p}(X_p)u_2^L(t) \quad (3.6)$$

where \hat{x}_p is the filter output for the p^{th} state, f_p , g_{1p} and g_{2p} are the p^{th} components of the vector functions f , g_1 and g_2 , respectively. The state X_p is obtained from both the actual state measurements, x , and the filter output, \hat{x}_p , as follows:

$$X_p(t) = [x_1(t), \dots, x_{p-1}(t), \hat{x}_p(t), x_{p+1}(t), \dots, x_n(t)]^T.$$

Note that in the filter of Eq. 3.6, the control inputs $u_1^L(t)$ and $u_2^L(t)$ are determined by LMPC 1 of Eq. 3.3 and LMPC 2 of Eq. 3.2 as applied to the actual process based on the state X_p , and are updated every control sampling time Δ (i.e., the sampling time instants $\{t_{k \geq 0}\}$).

The FDI filters are only initialized at $t = 0$ such that $\hat{x}_p(0) = x_p(0)$. The information generated by the filters provides a fault-free estimate of the actual system state at any time t and allows easy detection of the actual system state deviations due to faults. For each state associated with a filter, an FDI residual is defined as follows:

$$r_p(t) = \|\hat{x}_p(t) - x_p(t)\|$$

with $p = 1, \dots, n$. The residual r_p is computed continuously because $\hat{x}_p(t)$ is known for all t and the state measurement, x , is also available for all t . If no fault occurs, the filter states track the system states. In this case, the dynamics of the system states and the FDI filter states are identical, so $r_p(t) = 0$ for all times.

In the practical case where sensor measurement noise and process noise are present, the residual will be nonzero even without an actuator fault. In order to reduce the influence of process noise on fault detection, we define a weighted residual $r_{E,p}$, $p = 1, \dots, n$, for each residual r_p , calculated at discrete time instants $\{t_{i \geq 0}\}$ with $t_i = t_0 + i\Delta_r$, $i = 0, 1, 2, \dots$. The weighted residual is calculated using an exponentially weighted moving average (EWMA) method as follows [27]:

$$r_{E,p}(t_i) = \lambda r_p(t_i) + (1 - \lambda)r_{E,p}(t_{i-1}) \quad (3.7)$$

with $r_{E,p}(t_0) = r_p(t_0)$ and the weighting factor $\lambda \in (0, 1]$. The parameter λ determines the rate at which previous data enter into the calculations of the weighted residual. When $\lambda = 1$, $r_{E,p}$ is equivalent to r_p . The benefit of using EWMA residuals is their ability to better capture smaller

drifts in the system and to provide protection against occasional spikes. The value of λ is typically set between 0.2 and 0.5 depending on the sensitivity and responsiveness desired [27]. All further mention of residuals will be in reference to the EWMA residuals.

Also due to sensor measurement and process noise, fault detection thresholds are necessary so that a fault is declared only when a residual exceeds its specific threshold value. The thresholds are based on historical process variance data under no fault (normal) operating conditions and chosen to the desired degree of confidence to quickly detect possible faults. In some cases, the residual may deviate temporarily due to normal process variance and should not be interpreted as a fault. In these cases, it is important to properly confirm that the residual is deviating because of a fault by waiting a specified amount of time. By waiting, we improve our confidence that a fault has occurred and reduce the probability of false alarms. In the detection of a fault, three threshold values for each EWMA residual are used. The threshold values for the EWMA residual, $r_{E,p}$, are calculated as discussed in [27] using the following formula:

$$\sigma_{p,k} = \bar{r}_p + ks_p \sqrt{\frac{\lambda}{2-\lambda}} \quad (3.8)$$

where $k = 3, 4, 5$ are the weighting factors used, \bar{r}_p and s_p are the mean value and standard deviation of the p^{th} residual (r_p) based on historical fault-free operation data of the closed-loop system, respectively. Specifically, the historical process operation data in the application discussed in the next section were obtained by running the closed-loop system under fault-free conditions for a simulated period of 5 hours and collecting the system state and residual values in order to compute the average value and standard deviation for each state and each residual. These values were then used in conjunction with Eq. 3.8 to compute the threshold values ($\sigma_{p,k}$) for each EWMA residual.

3.4.2 Fault detection and isolation using adaptive windows

In this subsection, we augment our previous FDI system [34] to include an adjustable time window based on the rate of change of the residual with the goals of reducing the probability of false alarms and false isolation, and achieving a quicker fault recovery response.

On the occurrence of a fault, certain residuals directly associated with the fault will immediately

become nonzero at different rates (or in the case where process noise and measurement noise are present their thresholds will be exceeded at different times depending on the fault's magnitude). An improvement over previous work is the use of EWMA residuals in combination with adjustable fault isolation time windows.

When there is a residual that exceeds its second threshold and stays above it for a time period Δt_d , then a fault is declared. For example, if $r_{E,p}$ exceeds $\sigma_{p,4}$ at time $t_{\sigma_{p,4}}$ and stays above $\sigma_{p,4}$ from $t_{\sigma_{p,4}}$ to $t_{\sigma_{p,4}} + \Delta t_d$, then a fault is declared. The waiting time Δt_d is used to reduce the incidence of false alarms and, in particular, intermittent spikes.

Fault isolation is carried out simultaneously with fault detection. We define a fault signature as a set $I_p = [i_1, i_2, \dots, i_n]$ where $i_p = 1$ if the residual $r_{E,p} \geq \sigma_{p,3}$, otherwise $i_p = 0$ for a particular fault $\tilde{u}_{f,p}$. We consider that the system has an isolable structure which implies that each of the possible faults has a unique fault signature; that is, $I_p \neq I_q$ with $p \neq q$. Based on the rate of change of the first residual which exceeds its second threshold, a time window over which a fault may be isolated is calculated. If there is no residual that exceeds its third threshold within the time window, the fault is isolated at the end of the time window. The isolated fault has a signature composed of all the residuals that exceed their second thresholds. If there is at least one residual that exceeds its third threshold within the fault isolation time window, a new fault isolation time window is calculated and the fault is isolated at the end of this new time window. For example, if $r_{E,p}$ is the first residual that exceeds its threshold $\sigma_{p,4}$, a time window, Δt_p , is calculated as follows:

$$\Delta t_p = w(t_{\sigma_{p,4}} - t_{\sigma_{p,3}}) \quad (3.9)$$

where w is a constant or a complex function of the model and its current state, and $t_{\sigma_{p,4}}$ and $t_{\sigma_{p,3}}$ are the time instants the residual $r_{E,p}$ exceeds $\sigma_{p,4}$ and $\sigma_{p,3}$, respectively. If from $t_{\sigma_{p,4}}$ to $t_{\sigma_{p,4}} + \Delta t_p$, there is no residual that exceeds its third threshold, the fault is isolated at time $t_{\sigma_{p,4}} + \Delta t_p$ with a signature composed of all the residuals whose values exceed their second thresholds. If from $t_{\sigma_{p,4}}$ to $t_{\sigma_{p,4}} + \Delta t_p$, there is at least one residual that exceed its third threshold, for example, $r_{E,q}$ exceeds $\sigma_{q,5}$ at time $t_{\sigma_{q,5}}$, then a new fault isolation time window, Δt_q , is calculated following Eq. 3.9 based on the change rate of $r_{E,q}$ from $t_{\sigma_{q,4}}$ to $t_{\sigma_{q,5}}$, and the fault is isolated at $\min\{t_{\sigma_{p,4}} + \Delta t_p, t_{\sigma_{q,5}} + \Delta t_q\}$.

An example with one residual is presented in Fig. 3.1. In Fig. 3.1, a fault occurs at time t_f ,

which drives a residual $r_{E,p}$ to go up. The residual exceeds its first and second thresholds at time $t_{\sigma_{p,3}}$ and $t_{\sigma_{p,4}}$, respectively. Once the residual exceeds its second threshold $\sigma_{p,4}$ and stays above it for a short waiting time Δt_d , a fault is declared at time t_d ($t_d = t_{\sigma_{p,4}} + \Delta t_d$). At the time $t_{\sigma_{p,4}}$, a fault isolation time window Δp is also calculated. Because the residual continues to increase and exceeds its third threshold, $\sigma_{p,5}$, at time $t_{\sigma_{p,5}}$, a new isolation window Δt_{p2} is calculated. The fault is isolated at time $t_{s1} = \min\{t_{s1}, t_{s2}\}$. The isolated fault has a signature composed of all the residuals that exceed their second threshold.

3.4.3 Fault parameter estimation

After a fault has been isolated, the FTC system must know the magnitude of the fault in order to target the corresponding new operating point and properly stabilize the system in the presence of the fault. To simplify the description of the proposed method, we consider faults of constant magnitudes in this work; however, faults with slowly time-varying values can be handled using the proposed method in a straightforward manner.

When a residual ($r_{E,p}$) exceeds its first threshold ($\sigma_{p,3}$), we begin to collect the sampled system states as well as the actual control inputs applied to the system. When the fault is confirmed and isolated, a least square optimization problem is solved to get an estimate of the magnitude of the fault based on the sampled system states and the actual control inputs. Specifically, we collect the sampled system states, $x(t)$, and record the actual control inputs (i.e., $u_1(t) = u_1^L(t)$ and $u_2(t) = u_2^L(t)$) applied to the system for $t_{\sigma_{p,3}}$ to the fault isolation time $t_{isolate}$ with a sampling time Δ_e . The magnitude of the fault $\tilde{u}_{f,j}$ is estimated by solving the following optimization problem:

$$\min_{\tilde{u}_{f,j}} \sum_{i=0}^M (x(t_f + i\Delta_e) - \tilde{x}(t_f + i\Delta_e))^2 \quad (3.10a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g(\tilde{x}(t))(u^L(t) + d) \quad (3.10b)$$

$$\tilde{x}(t_f) = x(t_f) \quad (3.10c)$$

where $u^L(t) = [u_1^L(t)^T \ u_2^L(t)^T]^T$ is the actual control inputs that have been applied to the closed-loop system from $t_{\sigma_{p,3}}$ to $t_{isolate}$, M is the maximum integer satisfying $M\Delta_e \leq t_{isolate} - t_{\sigma_{p,3}}$, $d = [0 \ \cdots \ \tilde{u}_{f,j} \ \cdots \ 0]^T$ is the fault vector, and $x(t_f)$ is the system state at the fault detection

time. The solution to the optimization problem of Eq. 3.10 is denoted by $\tilde{u}_{f,j}^*$, which is an optimal estimate of the actual fault $\tilde{u}_{f,j}$ from a least-square point of view.

3.4.4 FTC strategies

When a fault is detected, isolated and the magnitude of the fault is estimated, suitable FTC strategies can be carried out to keep the closed-loop system state within a desired operating region. Because of the fault, the origin (the operating point of the fault-free system) may not be achievable because of the input constraints and the system structure. In this case, we may operate the system at a new operating point within the desired operating region. To determine the new operating point x_s , we propose to solve an optimization problem. Specifically, when the fault is $\tilde{u}_{f,j}^*$, the new operating point, x_s , is obtained by solving the following optimization problem:

$$\min_{x_s, u_s} \|x_s\|_S \quad (3.11a)$$

$$\text{s.t. } f(x_s) + g(x_s)(u_s + d) = 0 \quad (3.11b)$$

$$u_s + d \in U \quad (3.11c)$$

$$x_s \in X \quad (3.11d)$$

where S is a positive weighting matrix, $d = [0 \ \cdots \ \tilde{u}_{f,j}^* \ \cdots \ 0]^T$ and X denotes the desired operating state region. The objective of the above optimization problem is to find an operating point within the desired operating state region such that the distance (measured by weighted Euclidean norm) between the new operating point and the origin is minimized. We assume that the optimization problem of Eq. 3.11 is always feasible which implies that we can always find the new operating point x_s and the corresponding new steady-state control input values $u_s = [u_{1s}^T \ u_{2s}^T]^T$.

Note that the proposed method is only one of many possible approaches to determine the new operating point in the case of a fault. The basic idea of the proposed method is to find a new operating point that stays as close as possible to the original operating point (i.e., the origin $x = 0$).

Once we find the new operating point x_s , we proceed to design the FTC strategies for the fault-free DMPC system (see Eqs. 3.2-3.3) in the presence of actuator faults. In general, when there is a

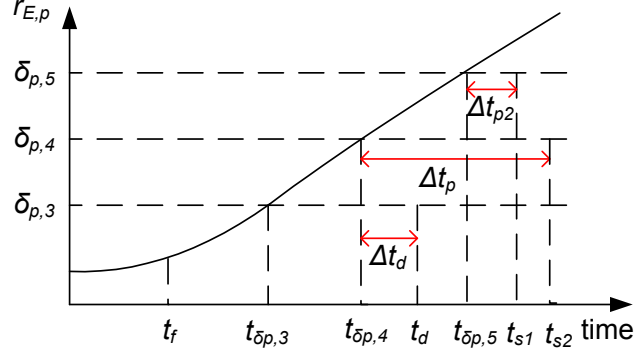


Figure 3.1: An example of residual evolution after the occurrence of a fault at t_f that affects residual $r_{E,p}$.

fault in the control system, it is impossible to carry out FTC unless there is another backup control loop. However, in the fault-free DMPC system, because of the extra control flexibility brought into the whole system by u_2 (LMPC 2), it is possible in some cases to carry out FTC without activating new control actuators.

When there is a persistent fault in the loop of u_2 which is denoted d_2 , and the fault can be detected and isolated in a reasonable time frame, it is possible to switch off the controller LMPC 2 and only use u_1 in the control system. When LMPC 2 is switched off from the closed-loop system, u_2 is set by the fault value (i.e., $u_2 = d_2$); and in the DMPC scheme of Eqs. 3.2-3.3, only LMPC 1 is evaluated at each sampling time. In order to maintain the stability of the closed-loop system, the design of LMPC 1 will need to be updated with the new operating point and its corresponding new steady-state control input values (i.e., the cost function $L(x, u_1, u_2)$ needs to be updated with x_s and u_s in a way such that $L(x_s, u_{1s}, u_{2s}) = 0$), and updated with the fault magnitude information (i.e., $u_2 = d_2$); the design of $h(x)$ also needs to be updated with the new steady-state information. The control inputs determined by the updated LMPC 1 will be referred to as $u'_1(x)$. This FTC strategy will maintain the closed-loop stability if implemented quickly such that the state of the closed-loop system is still within the stability region of the backup controllers and parameter estimation is sufficiently accurate, however, the performance of the closed-loop system may degrade to some extent.

When there is a fault in the subset u_{12} which is denoted d_1 , the FTC strategy would shut down the control action of u_{12} and reconfigure the DMPC algorithms to the backup DMPC of

Eqs. 3.4-3.5 to manipulate u_{11} and u_2 to control the process. In order to maintain the stability of the closed-loop system, the designs of the two backup LMPCs and the design of $h_2(x)$ needs to be updated with the new operating point and the corresponding new steady-state control input values; as well as being updated with the fault magnitude information. The control inputs determined by the updated designs will be referred to as $u_1''(x)$ and $u_2''(x)$.

However, when there is a fault in the subset u_{11} , it is impossible to successfully carry out FTC without activating backup actuators within the DMPC systems for the class of nonlinear systems considered in this work.

The FTC switching rules for the system of Eq. 3.1 within the DMPC system of Eqs. 3.2-3.3 are described as follows:

1. When a fault in the actuator associated with u_2 is isolated at t_f , the FTC switching rule is:

$$u_1(t) = \begin{cases} u_1^L(x), & t \leq t_f \\ u_1', & t > t_f \end{cases} \quad (3.12a)$$

$$u_2(t) = \begin{cases} u_2^L(x), & t \leq t_f \\ d_2, & t > t_f \end{cases} \quad (3.12b)$$

2. When a fault in the actuator associated with u_{12} is detected at t_f , the FTC switching rule is:

$$u_1(t) = \begin{cases} u_1^L(x), & t \leq t_f \\ \begin{bmatrix} u_{11}''(x) \\ d_1 \end{bmatrix}, & t > t_f \end{cases} \quad (3.13a)$$

$$u_2(t) = \begin{cases} u_2^L(x), & t \leq t_f \\ u_2''(x), & t > t_f \end{cases} \quad (3.13b)$$

Remark 3.2. *While in the present work, the full process state is assumed to be measured in real-time, it would be possible to implement the proposed approach using partial measurements of the full process state vector, provided that the available measurements in such a case allow to detect, isolate and estimate the magnitude of the fault.*

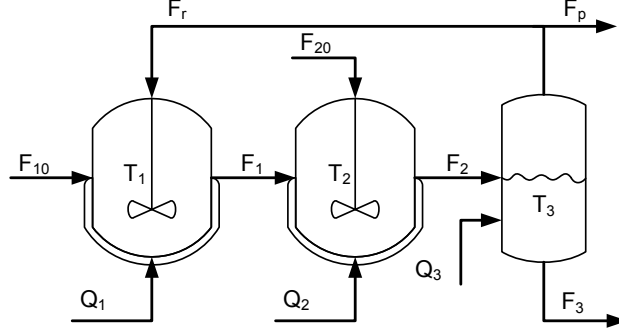


Figure 3.2: Two CSTRs and a flash tank with recycle stream.

Remark 3.3. *In the present work, a DMPC system involving two distributed LMPCs which are solved in sequence is adopted to control the process of Eq. 3.1. The benefit of solving the two LMPCs sequentially is that a single communication (i.e., LMPC 2 is solved first and sends its optimal trajectory to LMPC 1 which then calculates its own trajectory) leads to control actions that guarantee closed-loop stability due to the Lyapunov constraints while simultaneously solving LMPC 1 and LMPC 2 may require multiple communications between the two controllers to achieve a similar performance level.*

Remark 3.4. *Because of the structure of the system considered, it is possible in general that the origin is outside the accessible region of the system at the time of DMPC reconfiguration after fault detection and isolation has occurred (i.e., the reconfigured DMPC cannot stabilize the closed-loop system at the origin). What can be done and is done in this case in the present work (see next section), is to reconfigure the backup control system to try to maintain the closed-loop system within a region as close to the origin as possible.*

3.5 Application to a reactor-separator process

3.5.1 Process description and modeling

The process considered in this study is a three vessel, reactor-separator system consisting of two CSTRs and a flash tank separator as shown in Fig. 3.2. Its detailed description and modeling can be found in [6]. Sensor and process noise were added to the simulations. The desired operating steady-state is the unstable steady state, x_s , whose values are shown in Table 3.1.

For this process, we have two sets of manipulated inputs. The first set of manipulated inputs is the heat injected to or removed from the three vessels, that is $u_1 = [Q_1 - Q_{1s} \ Q_2 - Q_{2s} \ Q_3 - Q_{3s}]^T$; the second set includes the inlet flow rate to vessel 2, that is $u_2 = F_{20} - F_{20s}$. The variables Q_{1s} , Q_{2s} , Q_{3s} and F_{20s} denote the steady-state input values of the inputs whose values are shown in Table 3.2. The control inputs are subject to the constraints $|Q_i - Q_{is}| \leq u_1^{\max} = 10^6 \text{ KJ/hr}$, ($i = 1, 2, 3$) and $|F_{20} - F_{20s}| \leq u_2^{\max} = 5 \text{ m}^3/\text{hr}$.

In the design of the fault free DMPC system for the process, we consider a quadratic Lyapunov function $V(x) = x^T P x$ with $P = \text{diag}([20 \ 10^3 \ 10^3 \ 10^3 \ 10 \ 10^3 \ 10^3 \ 10^3 \ 10 \ 10^3 \ 10^3 \ 10^3])$ and design the controller $h(x)$ as three PI controllers with proportional gains $K_{p1} = K_{p2} = K_{p3} = 8000$ and integral time constants $\tau_{I1} = \tau_{I2} = \tau_{I3} = 10$ based on the measurements of T_1 , T_2 and T_3 , respectively. Note that, in the absence of process noise and measurement noise, this design of $h(x)$ manipulating u_1 can stabilize the closed-loop system asymptotically without the use of u_2 . Based on $h(x)$ and $V(x)$, we design LMPC 1 following Eq. 3.3 to determine u_1 and LMPC 2 following Eq. 3.2 to determine u_2 . In the design of the LMPCs, the weighting matrices are chosen to be $Q_c = P$, $R_1 = \text{diag}([(5 \ 5 \ 5) \cdot 10^{-12}])$ and $R_2 = 100$. The horizon for the optimization problem is $N = 4$ with a time step of $\Delta = 0.05 \text{ hr}$.

In addition, the set of control inputs u_1 can be divided into two subsets, $u_{11} = [Q_1 - Q_{1s} \ Q_3 - Q_{3s}]^T$ and $u_{12} = Q_2 - Q_{2s}$. The input combination u_{11} and u_2 is able to stabilize the closed-loop system which can be used as the input configuration of the backup DMPC system of Eqs. 3.4-3.5. In order to design the backup DMPC, we need to design a second Lyapunov-based controller $h_2(x)$ which manipulates u_{11} and u_2 . We also design h_2 through PI control law with proportional gains $K_{p1}^b = K_{p2}^b = 8000$, $K_{p3}^b = -0.3$ and integral time constants $\tau_{I1}^b = \tau_{I2}^b = \tau_{I3}^b = 10$ based on the measurements of T_1 , T_3 and T_2 , respectively. The control design h_2 can stabilize the closed-loop system asymptotically with $Q_2 = 0$ in the absence of process noise and measurement noise. In the design of the backup DMPC system, we choose $V_2(x) = V(x)$.

In order to perform FDI for the reactor-separator system, we construct the FDI filters for the states affected directly by the four manipulated inputs as in Eq. 3.6. The states affected directly by the manipulated inputs are T_1 , C_{A2} , C_{B2} , C_{C2} , T_2 and T_3 . The FDI residuals take the following

form:

$$\begin{aligned} r_{T_i}(t) &= |\hat{T}_i(t) - T_i(t)|, \quad i = 1, 2, 3 \\ r_{C_{i2}}(t) &= |\hat{C}_{i2}(t) - C_{i2}(t)|, \quad i = A, B, C. \end{aligned} \tag{3.14}$$

Based on these residuals, we design the EWMA residuals with $\lambda = 0.5$ and the sampling time $\Delta_r = 0.005$. The mean values and standard deviations of the EWMA residuals are shown in Table 3.3.

We consider two different faults in the following simulations. First, we consider a fault in the heat input/removal actuator to vessel 2, that is a fault in Q_2 . Because Q_2 only affects the state T_2 directly and all the measurements are continuously available, when there is an actuator fault in Q_2 , only the residual corresponding to T_2 exceeds its threshold. The second fault we consider is a fault in the inlet flow actuator to vessel 2, that is a fault in F_{20} . Because the control action F_{20} affects directly the states T_2 , C_{A2} , C_{B2} and C_{C2} , when there is an actuator fault in F_{20} , more than one of the residuals will exceed their thresholds. In the simulations, $\Delta t_d = 36 \text{ s} = 0.01 \text{ hr}$, $w = [4 \ 3 \ 3; 2 \ 2 \ 2]$ and $\Delta_e = 0.005 \text{ hr}$.

3.5.2 Simulation Results

Four different simulation sets are presented to demonstrate the merits of isolating by using the adaptive windows based on EWMA residuals. For each simulation, the plant is initialized at the desired steady-state x_s (see Table 3.1) and simulated to 5.0 *hr* with a fault being triggered at 1.050*hr*. Process and measurement noise is applied to the plant.

The first case considered triggers a small magnitude Q_2 fault that will demonstrate longer isolation windows to minimize false alarms along with a quicker response since fault detection/isolation begins tracking the potential fault sooner. The second and third case will demonstrate the quick detection and isolation of a large magnitude Q_2 fault using adaptive windows in comparison to an FDI scheme using fixed isolation times. The fourth case will demonstrate an F_{20} small magnitude fault.

In the first set of simulations, a Q_2 fault with a magnitude of 15% of u_1^{\max} is triggered at 1.050 *hr* (we will refer to it as “small” Q_2 fault). From the design of the system, the Q_2 fault directly affects the temperature in vessel 2 where we expect only the residual for T_2 to deviate. When the residual

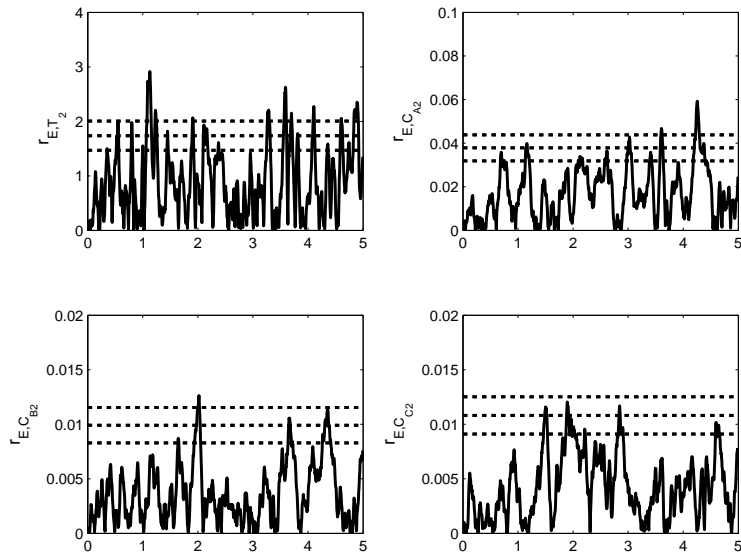


Figure 3.3: Case 1: Q_2 “small” fault is isolated using longer waiting time calculated from the residual change of T_2 . r_{E,T_2} (top left plot) exceeds $\sigma_{T_2,3}$ at 1.065 *hr* and then exceeds $\sigma_{T_2,4}$ where the fault isolation time is set to 3.6 *min*. When r_{E,T_2} further exceeds $\sigma_{T_2,5}$ at 1.080 *hr*, the waiting time is updated to 36 *sec*. The fault is isolated and estimated as 18.5 *KJ/hr* (actual 20.0 *KJ/hr*) at 1.090 *hr*

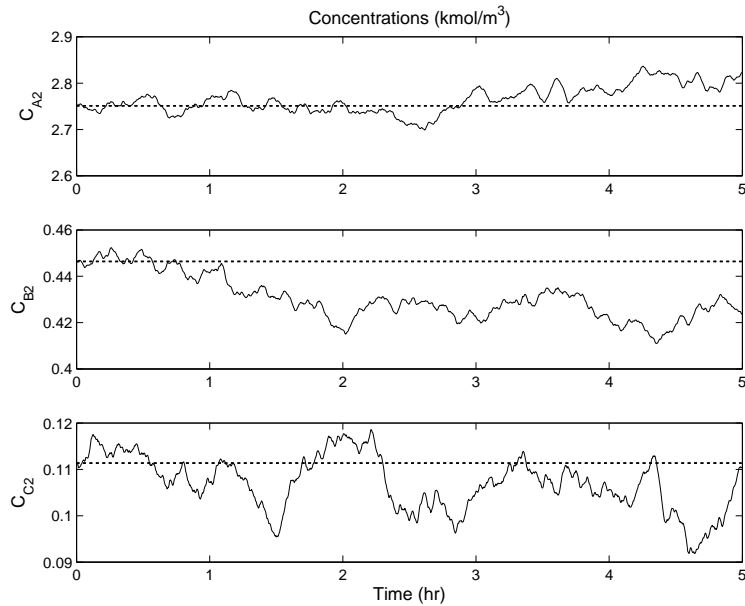


Figure 3.4: Case 1: Q_2 “small” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Concentrations. Note the new steady state values and scale.

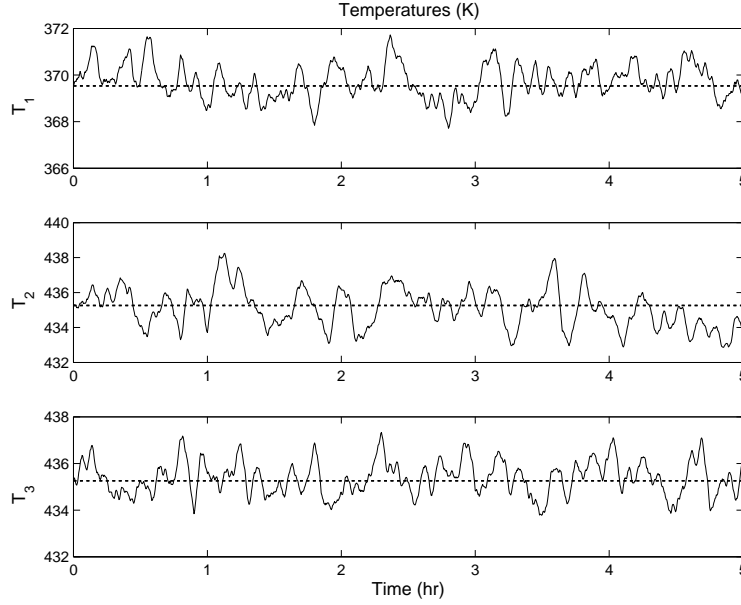


Figure 3.5: Case 1: Q_2 “small” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Temperatures.

for T_2 , r_{E,T_2} , (see top left plot of Fig. 3.3) exceeds a chosen confidence level (i.e., its first threshold $\sigma_{T_2,3}$) at 1.065 *hr*, the FDI system begins monitoring the rate of change of the T_2 residual. The residual r_{E,T_2} exceeds its second threshold $\sigma_{T_2,4}$ at 1.075 *hr*. This fault is confirmed and declared at time 1.085 *hr* after the waiting time $\Delta t_d = 0.01$ *hr*. At the same time 1.075 *hr*, a fault isolation window of 3.6 *min* = 0.1 *hr* is calculated based on the rate of change of r_{E,T_2} to ensure proper isolation of the fault. However, because r_{E,T_2} deviates quickly and exceeds $\sigma_{T_2,5}$ at 1.080 *hr*, the isolation window is updated to 36 *sec* = 0.01 *hr* and the Q_2 fault is isolated at 1.090 *hr*. The fault is estimated at 18.5 *KJ/hr* (actual fault value is 20 *KJ/hr*). The fault tolerant control system reconfigures the control system, which is able to stabilize the closed-loop system near the target steady state by 1.500 *hr* as shown in the concentration profiles in Fig. 3.4 and the temperature profiles in Fig. 3.5.

In the second case, a Q_2 fault is set to a magnitude of 80% of u_1^{\max} and is triggered at 1.050 *hr* (we will refer to it as the “large” Q_2 fault). The larger Q_2 fault will demonstrate the FDI system’s quicker response and improved robustness when used in conjunction with fault tolerant control. In Fig. 3.7, the “large” fault compared to a “small” fault of case 1 (Fig. 3.3) causes the residual to deviate much quicker with the FDI system beginning to monitor at 1.060 *hr* when r_{T_2} immediately

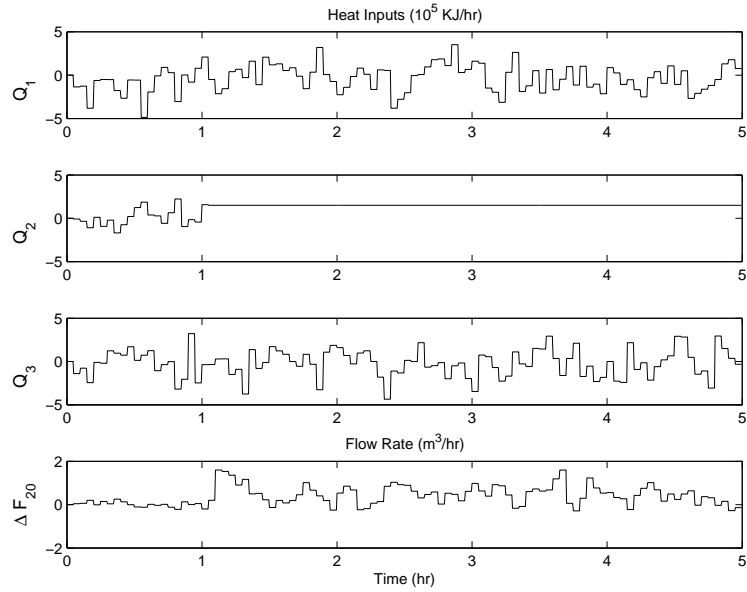


Figure 3.6: Case 1: Q_2 “small” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Control actions.

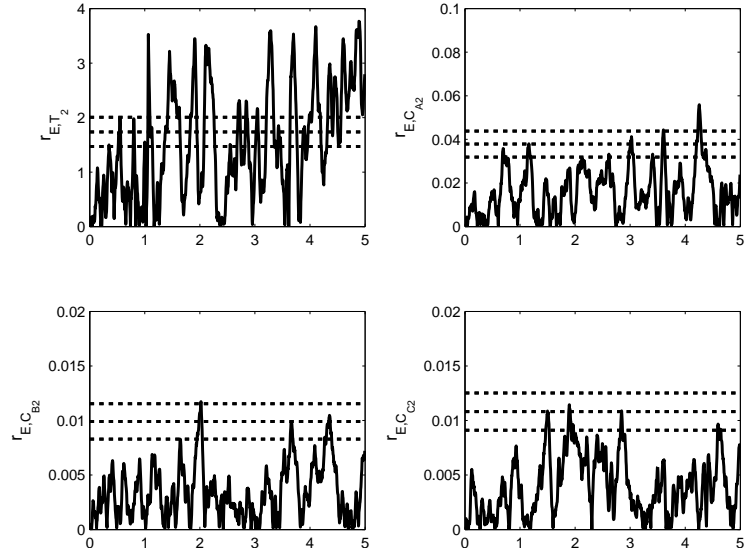


Figure 3.7: Case 2: Q_2 “large” fault is isolated using a shorter waiting time based on residual change of T_2 . r_{E,T_2} (top left plot) immediately exceeds $\sigma_{T_{2,5}}$ in a single measurement update at 1.060 hr. The calculated waiting time is 36 sec. The fault is estimated as 88 KJ/hr (actual 80 KJ/hr) and FTC is implemented at 1.070 hr

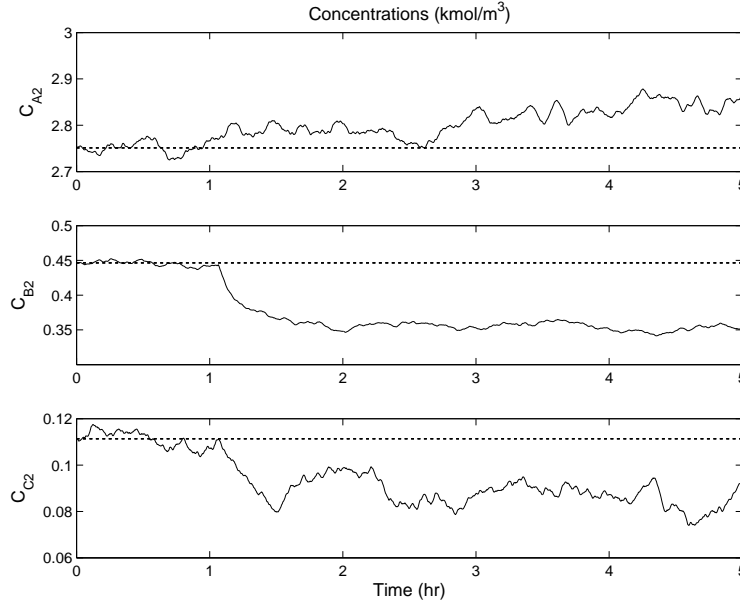


Figure 3.8: Case 2: Q_2 “large” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Concentrations. Note the new steady state values and scale.

exceeds $\sigma_{T_2,5}$ and an isolation window of 36 s is calculated. The fault is declared and isolated at 1.070 hr soon after the r_{E,T_2} exceeded $\sigma_{T_2,5}$, and the fault is estimated as 88 KJ/hr (actual 80 KJ/hr). Figures 3.8 and 3.9 show that the FTC system is able to stabilize the system at a new steady-state after reconfiguration at 1.070 hr.

The purpose of the third case is to better illustrate the need for variable windows and minimum waiting times for proper isolation. In the third case we trigger an identical Q_2 fault as in case 2 with the exception that the FDI system uses fixed isolation windows. Similarly, when r_{E,T_2} first exceeds $\sigma_{T_2,5}$ at 1.060 hr, the isolation system monitors the remaining residuals for a matching fault signature over a fixed window of 4.8 min. At the end of the fixed window at 1.135 hr, the FTC system (identical to that of case 2) reconfigures the DMPC system with a Q_2 fault estimate of 89 KJ/hr (actual 80 KJ/hr) and is unable to stabilize the system due to the plant state having left the stability region of the reconfigured control system. Note that after fault isolation the residuals are no longer used.

The fourth case involves an F_{20} fault whose fault signature includes r_{E,T_2} and at least one other concentration residual ($r_{E,CA_2}, r_{E,CB_2}, r_{E,CC_2}$). In this case, an F_{20} fault occurs with a magnitude of 17% of u_2^{\max} . In Fig. 3.14, the T_2 residual exceeds $\sigma_{T_2,3}$ at 1.100 hr while the residual for

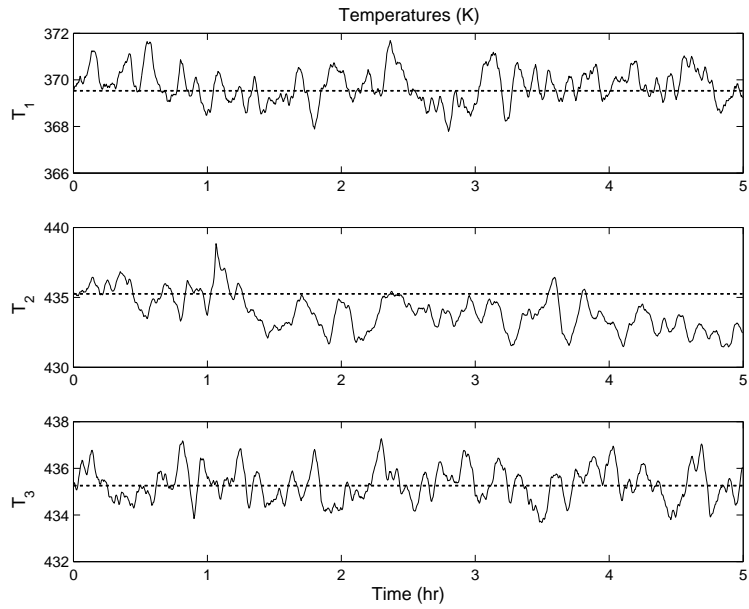


Figure 3.9: Q_2 “large” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Temperatures.

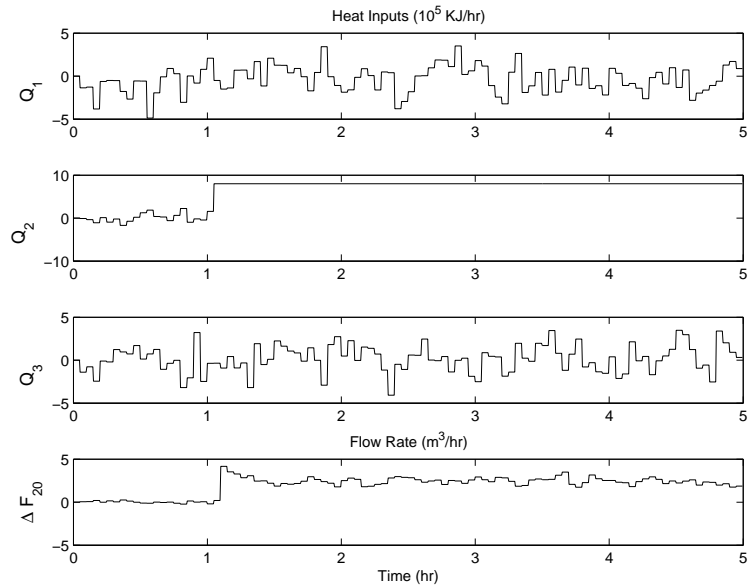


Figure 3.10: Q_2 “large” fault is isolated and control system is reconfigured to stabilize the closed-loop system - Control actions.

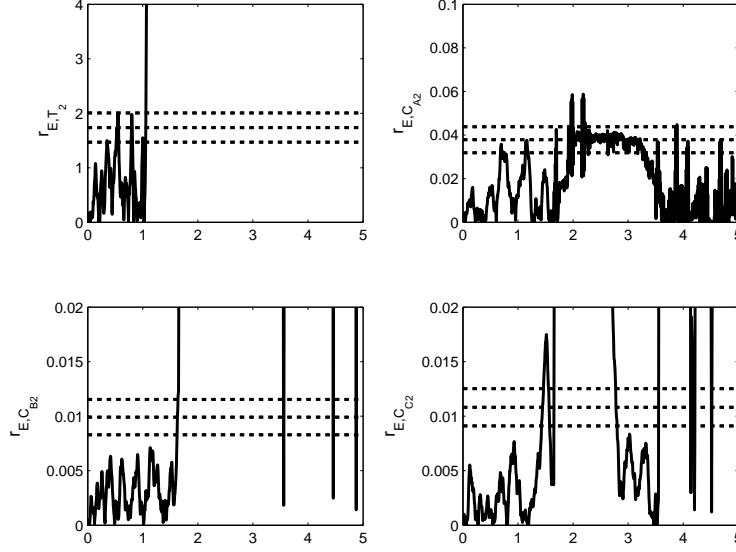


Figure 3.11: Case 3: Same fault conditions as in case 2, i.e., Q_2 “large” fault, using fixed isolation time. r_{E,T_2} immediately exceeding $\sigma_{T_{2,5}}$ at 1.060 hr, but the isolation windows is set to worst case condition of 4.8 min. The fault is isolated at 1.135 hr and estimated as 89 KJ/hr (actual 80 KJ/hr) but the FTC is unable to stabilize the closed-loop system.

Table 3.1: The desired operating steady-state x_s .

T_1	C_{A1}	C_{B1}	C_{C1}
370 [K]	3.32 [kmol/m ³]	0.17 [kmol/m ³]	0.04 [kmol/m ³]
T_2	C_{A2}	C_{B2}	C_{C2}
435 [K]	2.75 [kmol/m ³]	0.45 [kmol/m ³]	0.11 [kmol/m ³]
T_3	C_{A3}	C_{B3}	C_{C3}
435 [K]	2.88 [kmol/m ³]	0.50 [kmol/m ³]	0.12 [kmol/m ³]

concentration of component B in the second tank exceeds $\sigma_{C_{B2,4}}$ at 1.105 hr. A fault is declared at 1.115 hr when $r_{E,C_{B2}}$ stays above $\sigma_{C_{B2,4}}$ for 0.01 hr. A fault isolation window of 4.8 min is calculated at 1.105 hr. However, within the isolation window, $r_{E,C_{B2}}$ exceeds $\sigma_{C_{B2,5}}$ at 1.120 hr and a new isolation window of 36 sec is calculated. At the end of the new isolation window (i.e., $t = 1.130$ hr), no matching fault signature is found and the FDI system continues monitoring the residuals until 1.150 hr a matching fault signature is found when r_{E,T_2} exceeds $\sigma_{T_{2,5}}$ at 1.140 hr and stays above it for 0.01 hr. The FTC is implemented once the fault is isolated with a fault estimate 1.08 m³/hr (actual 0.85 m³/hr)

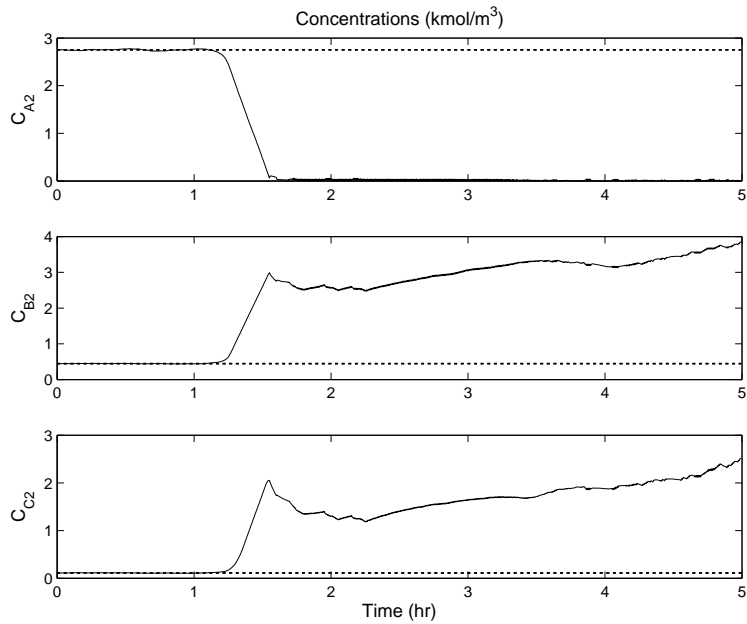


Figure 3.12: Case 3: Same fault conditions as in case 2, i.e., Q_2 “large” fault, using fixed isolation time. The control system is reconfigured at 1.135 *hr* and is unable to stabilize the closed-loop system - Concentrations.

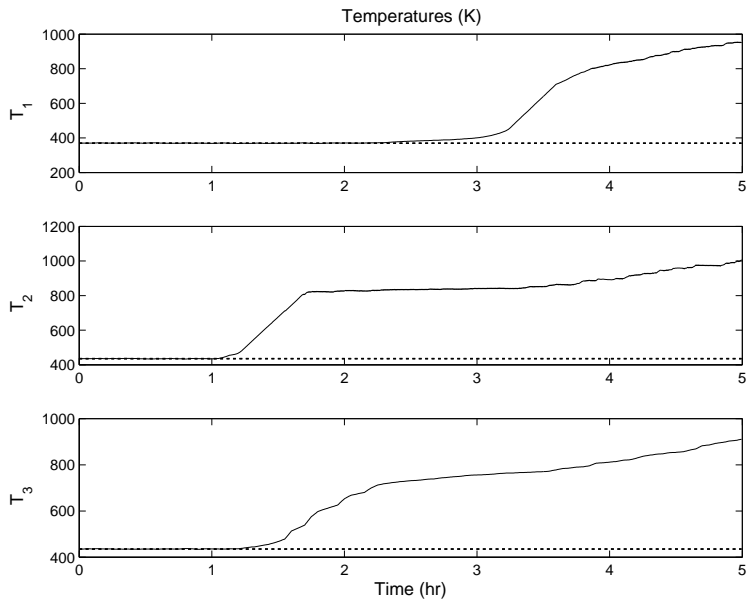


Figure 3.13: Case 3: Same fault conditions as in case 2, i.e., Q_2 “large” fault, using a fixed isolation time. The control system is reconfigured at 1.135 *hr* and is unable to stabilize the closed-loop system - Temperatures.

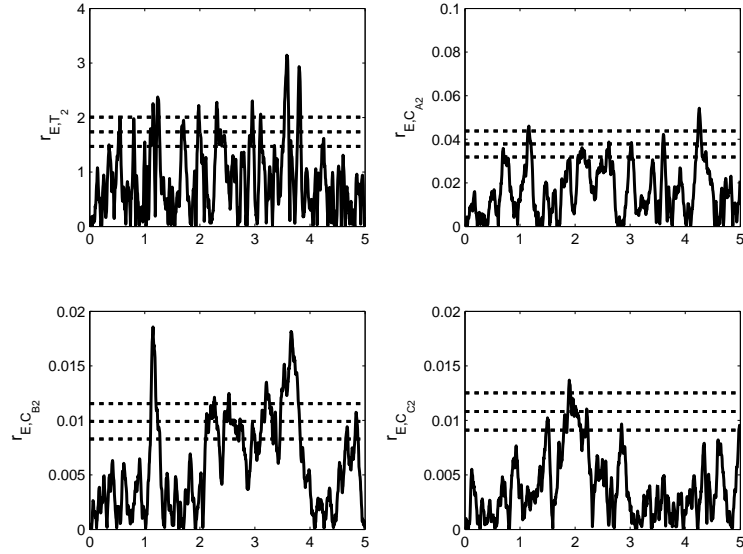


Figure 3.14: Case 4: F_{20} fault demonstrates FDI system with multiple residuals exceeding the thresholds under small magnitude fault. The residual r_{T_2} first exceeds $\sigma_{T_2,3}$ at 1.100 *hr* and $r_{C_{B2}}$ exceeds $\sigma_{C_{B2},4}$ at 1.105 *hr*. A fault is declared at 1.115 *hr* and an isolation time window of 4.8 *min* is calculated. A new fault isolation window of 36 *sec* is calculated when $r_{C_{B2}}$ exceeds $\sigma_{C_{B2},5}$ at 1.120 *hr* and at the end of the new isolation window, no matching fault signature is found. The FDI system continues monitoring the residuals until 1.150 *hr* a matching fault signature is found. The FTC is implemented once the fault is isolated with a fault estimate 1.08 m^3/hr (actual 0.85 m^3/hr).

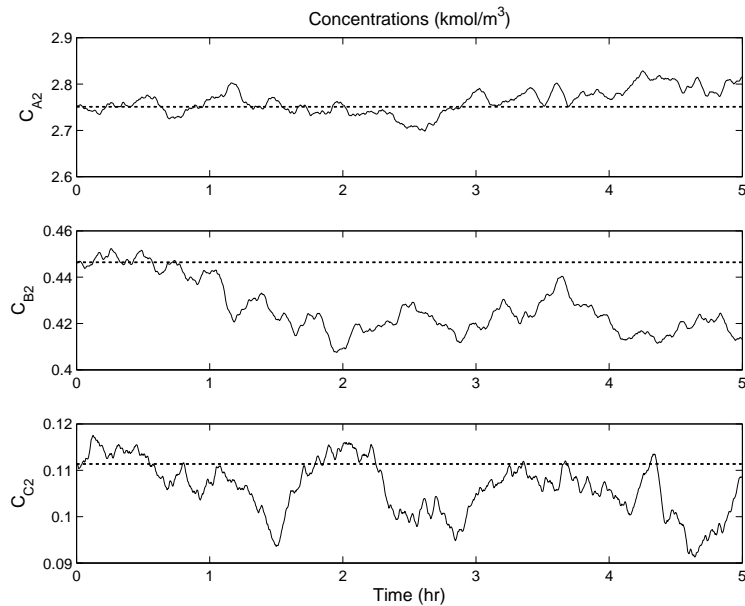


Figure 3.15: Case 4: F_{20} fault is isolated and control system is reconfigured to stabilize the closed-loop system - Concentrations.

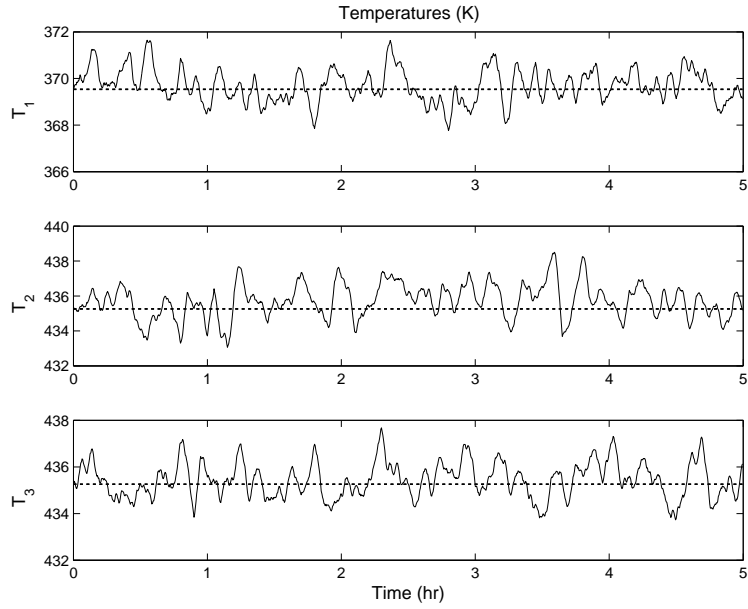


Figure 3.16: Case 4: F_{20} fault is isolated and control system is reconfigured to stabilize the closed-loop system - Temperatures.

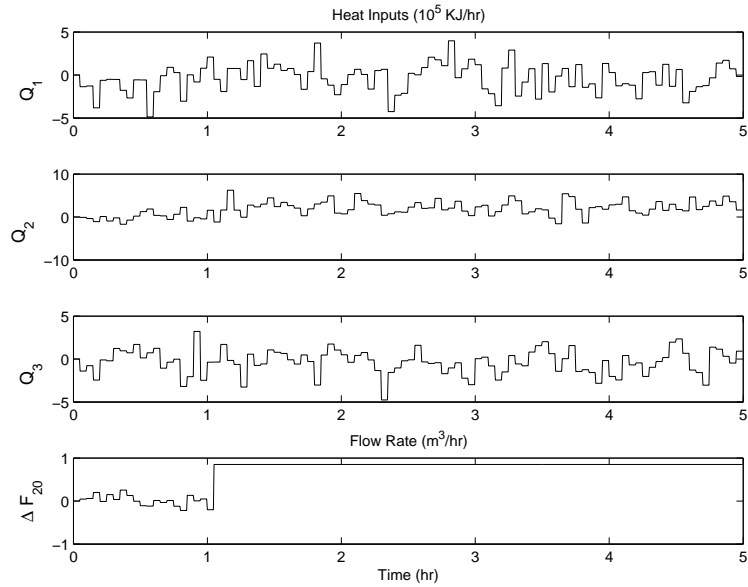


Figure 3.17: Case 4: F_{20} fault is isolated and control system is reconfigured to stabilize the closed-loop system - Control actions.

Table 3.2: The steady-state input values.

Q_{1s}	Q_{2s}	Q_{3s}	F_{20s}
0 [KJ/hr]	0 [KJ/hr]	0 [KJ/hr]	5 [m^3/hr]

Table 3.3: EWMA Residual Means and Standard Deviations.

\bar{r}_{T_2}	$\bar{r}_{C_{A2}}$	$\bar{r}_{C_{B2}}$	$\bar{r}_{C_{C2}}$
0.664900	0.013944	0.003421	0.003980
s_{T_2}	$s_{C_{A2}}$	$s_{C_{B2}}$	$s_{C_{C2}}$
0.464139	0.010351	0.002810	0.002960

3.6 Conclusions

In the chapter, we developed a monitoring and reconfiguration system for a DMPC system in the presence of control actuator faults taking advantage of both process models and process measurements. Specifically, we first designed fault detection filters and corresponding filter residuals, which are computed via EWMA method, to effectively detect actuator faults. Then, we proposed a fault isolation approach which uses adaptive fault isolation time windows to quickly and accurately isolate actuator faults and reduce the probability of false alarms. Subsequently, we designed appropriate FTC strategies to handle the actuator faults by reconfiguring the DMPC system and maintain the closed-loop system state within a desired operating region. The applicability and effectiveness of the proposed approach were illustrated via extensive simulations based on a nonlinear reactor-separator process example.

Chapter 4

Fault Detection and Isolation and Fault Tolerant Control of a Catalytic Alkylation of Benzene Process

4.1 Introduction

In this chapter, we focus on the application of an integrated fault detection, isolation and fault tolerant control (FDIFTC) framework to a catalytic alkylation of benzene process. We consider that the catalytic alkylation of benzene process is controlled by a distributed model predictive control (DMPC) system and is subjected to unknown, persistent actuator faults. The FDIFTC system monitors closed-loop process residuals in order to detect and isolate a faulty actuator. After isolation of an actuator fault, the FDIFTC system estimates the fault magnitude, recalculates a new optimal operating point, and ultimately reconfigures the DMPC system to maintain stability of the process in an optimal manner. Extensive simulations are carried out to demonstrate the performance of the FDIFTC system from closed-loop stability and performance points of view.

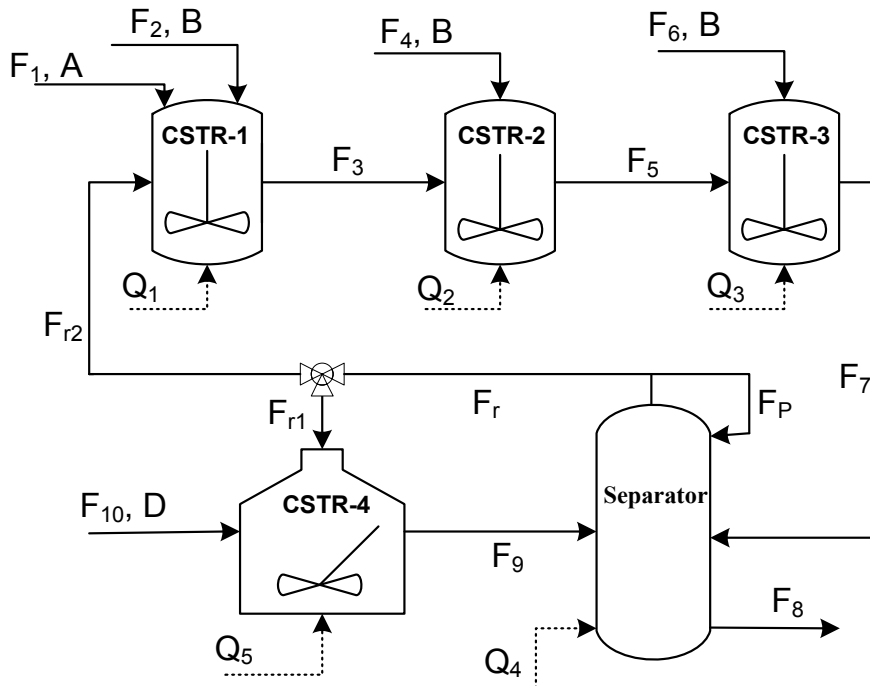


Figure 4.1: Process flow diagram of alkylation of benzene.

4.2 Description of the alkylation of benzene process

The process of alkylation of benzene with ethylene to produce ethylbenzene is widely used in the petrochemical industry. Dehydration of the product produces styrene, which is the precursor to polystyrene and many copolymers. The process model developed in this section is based on these references [16, 21, 38, 1, 56] and details can be found in [24]. In the remainder, we review this model for completeness of the presentation and of the results of this work. More specifically, the process considered in this work consists of four continuously stirred tank reactors (CSTRs) and a flash tank separator, as shown in Fig. 4.1. The CSTR-1, CSTR-2 and CSTR-3 are in series and involve the alkylation of benzene with ethylene. Pure benzene is fed from stream F_1 and pure ethylene is fed from streams F_2 , F_4 and F_6 . Two catalytic reactions take place in CSTR-1, CSTR-2 and CSTR-3. Benzene (A) reacts with ethylene (B) and produces the required product ethylbenzene (C) (reaction 1); ethylbenzene can further react with ethylene to form 1,3-diethylbenzene (D) (reaction 2) which is the byproduct. The effluent of CSTR-3, including the products and leftover reactants, is fed to a flash tank separator, in which most of benzene is separated overhead by

vaporization and condensation techniques and recycled back to the plant and the bottom product stream is removed. A portion of the recycle stream F_{r2} is fed back to CSTR-1 and another portion of the recycle stream F_{r1} is fed to CSTR-4 together with an additional feed stream F_{10} which contains 1,3-diethylbenzene from further distillation process that we do not consider in this example. In CSTR-4, reaction 2 and catalyzed transalkylation reaction in which 1,3-diethylbenzene reacts with benzene to produce ethylbenzene (reaction 3) takes place. All chemicals left from CSTR-4 eventually pass into the separator. All the materials in the reactions are in liquid phase due to high pressure. The dynamic equations describing the behavior of the process, obtained through material and energy balances under standard modeling assumptions, are shown below:

$$\begin{aligned}
\frac{dC_{A1}}{dt} &= \frac{F_1C_{A0} + F_{r2}C_{Ar} - F_3C_{A1}}{V_1} - r_1(T_1, C_{A1}, C_{B1}) \\
\frac{dC_{B1}}{dt} &= \frac{F_2C_{B0} + F_{r2}C_{Br} - F_3C_{B1}}{V_1} - r_1(T_1, C_{A1}, C_{B1}) - r_2(T_1, C_{B1}, C_{C1}) \\
\frac{dC_{C1}}{dt} &= \frac{F_{r2}C_{Cr} - F_3C_{C1}}{V_1} + r_1(T_1, C_{A1}, C_{B1}) - r_2(T_1, C_{B1}, C_{C1}) \\
\frac{dC_{D1}}{dt} &= \frac{F_{r2}C_{Dr} - F_3C_{D1}}{V_1} + r_2(T_1, C_{B1}, C_{C1}) \\
\frac{dT_1}{dt} &= \frac{Q_1 + F_1C_{A0}H_A(T_{A0}) + F_2C_{B0}H_B(T_{B0})}{\sum_i^{A,B,C,D} C_{i1}C_{pi}V_1} \\
&\quad + \frac{\sum_i^{A,B,C,D} (F_{r2}C_{ir}H_i(T_4) - F_3C_{i1}H_i(T_1))}{\sum_i^{A,B,C,D} C_{i1}C_{pi}V_1} \\
&\quad + \frac{(-\Delta H_{r1})r_1(T_1, C_{A1}, C_{B1})(-\Delta H_{r2})r_2(T_1, C_{B1}, C_{C1})}{\sum_i^{A,B,C,D} C_{i1}C_{pi}} \\
\frac{dC_{A2}}{dt} &= \frac{F_3C_{A1} - F_5C_{A2}}{V_2} - r_1(T_2, C_{A2}, C_{B2})
\end{aligned} \tag{4.1a}$$

$$\begin{aligned}
\frac{dC_{B2}}{dt} &= \frac{F_3C_{B1} + F_4C_{B0} - F_5C_{B2}}{V_2} - r_1(T_2, C_{A2}, C_{B2}) - r_2(T_2, C_{B2}, C_{C2}) \\
\frac{dC_{C2}}{dt} &= \frac{F_3C_{C1} - F_5C_{C2}}{V_2} + r_1(T_2, C_{A2}, C_{B2}) - r_2(T_2, C_{B2}, C_{C2}) \\
\frac{dC_{D2}}{dt} &= \frac{F_3C_{D1} - F_5C_{D2}}{V_2} + r_2(T_2, C_{B2}, C_{C2}) \\
\frac{dT_2}{dt} &= \frac{Q_2 + F_4C_{B0}H_B(T_{B0})}{\sum_i^{A,B,C,D} C_{i2}C_{pi}V_2} \\
&\quad + \frac{\sum_i^{A,B,C,D} (F_3C_{i1}H_i(T_1) - F_5C_{i2}H_i(T_2))}{\sum_i^{A,B,C,D} C_{i2}C_{pi}V_2} \\
&\quad + \frac{(-\Delta H_{r1})r_1(T_2, C_{A2}, C_{B2})(-\Delta H_{r2})r_2(T_2, C_{A2}, C_{B2})}{\sum_i^{A,B,C,D} C_{i2}C_{pi}} \\
\frac{dC_{A3}}{dt} &= \frac{F_5C_{A2} - F_7C_{A3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) \\
\frac{dC_{B3}}{dt} &= \frac{F_5C_{B2} + F_6C_{B0} - F_7C_{B3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}) \\
\frac{dC_{C3}}{dt} &= \frac{F_5C_{C2} - F_7C_{C3}}{V_3} + r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}) \\
\frac{dC_{D3}}{dt} &= \frac{F_5C_{D2} - F_7C_{D3}}{V_3} + r_2(T_3, C_{B3}, C_{C3}) \\
\frac{dT_3}{dt} &= \frac{Q_3 + F_6C_{B0}H_B(T_{B0})}{\sum_i^{A,B,C,D} C_{i3}C_{pi}V_3} \\
&\quad + \frac{\sum_i^{A,B,C,D} (F_5C_{i2}H_i(T_2) - F_7C_{i3}H_i(T_3))}{\sum_i^{A,B,C,D} C_{i3}C_{pi}V_3} \\
&\quad + \frac{(-\Delta H_{r1})r_1(T_3, C_{A3}, C_{B3})(-\Delta H_{r2})r_2(T_3, C_{B3}, C_{C3})}{\sum_i^{A,B,C,D} C_{i3}C_{pi}}
\end{aligned}$$

$$\begin{aligned}
\frac{dC_{A4}}{dt} &= \frac{F_7C_{A3} + F_9C_{A5} - F_rC_{Ar} - F_8C_{A4}}{V_4} \\
\frac{dC_{B4}}{dt} &= \frac{F_7C_{B3} + F_9C_{B5} - F_rC_{Br} - F_8C_{B4}}{V_4} \\
\frac{dC_{C4}}{dt} &= \frac{F_7C_{C3} + F_9C_{C5} - F_rC_{Cr} - F_8C_{C4}}{V_4} \\
\frac{dC_{D4}}{dt} &= \frac{F_7C_{D3} + F_9C_{D5} - F_rC_{Dr} - F_8C_{D4}}{V_4} \\
\frac{dT_4}{dt} &= \frac{Q_4 + \sum_i^{A,B,C,D} (F_7C_{i3}H_i(T_3) + F_9C_{i5}H_i(T_5))}{\sum_i^{A,B,C,D} C_{i4}C_{pi}V_4} \\
&\quad + \frac{\sum_i^{A,B,C,D} (-M_iH_i(T_4) - F_8C_{i4}H_i(T_4) - M_iH_{vapi})}{\sum_i^{A,B,C,D} C_{i4}C_{pi}V_4} \\
\frac{dC_{A5}}{dt} &= \frac{F_{r1}C_{Ar} - F_9C_{A5}}{V_5} - r_3(T_5, C_{A5}, C_{D5}) \\
\frac{dC_{B5}}{dt} &= \frac{F_{r1}C_{Br} - F_9C_{B5}}{V_5} - r_2(T_5, C_{B5}, C_{C5}) \\
\frac{dC_{C5}}{dt} &= \frac{F_{r1}C_{Cr} - F_9C_{C5}}{V_5} - r_2(T_5, C_{B5}, C_{C5}) \\
&\quad + 2r_3(T_5, C_{A5}, C_{D5}) \\
\frac{dC_{D5}}{dt} &= \frac{F_{r1}C_{Dr} + F_{10}C_{D0} - F_9C_{D5}}{V_5} \\
&\quad + r_2(T_5, C_{B5}, C_{C5}) - r_3(T_5, C_{A5}, C_{D5}) \\
\frac{dT_5}{dt} &= \frac{Q_5 + F_{10}C_{D0}H_D(T_{D0})}{\sum_i^{A,B,C,D} C_{i5}C_{pi}V_5} \\
&\quad + \frac{\sum_i^{A,B,C,D} (F_{r1}C_{ir}H_i(T_4) - F_9C_{i5}H_i(T_5))}{\sum_i^{A,B,C,D} C_{i5}C_{pi}V_5} \\
&\quad + \frac{(-\Delta H_{r2})r_2(T_5, C_{B5}, C_{C5})(-\Delta H_{r3})r_3(T_5, C_{A5}, C_{D5})}{\sum_i^{A,B,C,D} C_{i5}C_{pi}}
\end{aligned}$$

where r_1 , r_2 and r_3 are the reaction rates of reactions 1, 2 and 3 respectively and H_i , $i = A, B, C, D$, are the enthalpies of the reactants. The reaction rates are related to the concentrations of the reactants and the temperature in each reactor as follows:

$$r_1(T, C_A, C_B) = 0.0840e^{\frac{-9502}{RT}} C_A^{0.32} C_B^{1.5} \quad (4.4)$$

$$r_2(T, C_B, C_C) = \frac{0.0850e^{\frac{-20643}{RT}} C_B^{2.5} C_C^{0.5}}{(1 + k_{EB2} C_D)} \quad (4.5)$$

$$r_3(T, C_A, C_D) = \frac{66.1e^{\frac{-61280}{RT}} C_A^{1.0218} C_D}{(1 + k_{EB3} C_A)} \quad (4.6)$$

where:

$$k_{EB2} = 0.152e^{\frac{-3933}{RT}} \quad (4.7)$$

$$k_{EB3} = 0.490e^{\frac{-50870}{RT}}. \quad (4.8)$$

The heat capacities of the species are assumed to be constants and the molar enthalpies have a linear dependence on temperature as follows:

$$H_i(T) = H_{iref} + C_{pi}(T - T_{ref}), i = A, B, C, D \quad (4.9)$$

where C_{pi} , $i = A, B, C, D$ are heat capacities.

The model of the flash tank separator is developed under the assumption that the relative volatility of each species has a linear correlation with the temperature of the vessel within the operating temperature range of the flash tank, as shown below:

$$\alpha_A = 0.0449T_4 + 10 \quad (4.10)$$

$$\alpha_B = 0.0260T_4 + 10 \quad (4.11)$$

$$\alpha_C = 0.0065T_4 + 0.5 \quad (4.12)$$

$$\alpha_D = 0.0058T_4 + 0.25 \quad (4.13)$$

where α_i , $i = A, B, C, D$, represent relative volatility. It has also been assumed that there is a negligible amount of reaction taking place in the separator and a fraction of the total condensed

overhead flow is recycled back to the reactors. The following algebraic equations model the composition of the overhead stream relative to the composition of the liquid holdup in the flash tank:

$$M_i = k \frac{\alpha_i(F_7C_{i3} + F_9C_{i5}) \sum_j^{A,B,C,D} (F_7C_{j3} + F_9C_{j5})}{\sum_j^{A,B,C,D} \alpha_j(F_7C_{j3} + F_9C_{j5})}, i = A, B, C, D \quad (4.14)$$

where M_i , $i = A, B, C, D$ are the molar flow rates of the overhead reactants and k is the fraction of condensed overhead flow recycled to the reactors. Based on M_i , $i = A, B, C, D$, we can calculate the concentration of the reactants in the recycle streams as follows:

$$C_{ir} = \frac{M_i}{\sum_j^{A,B,C,D} M_j/C_{j0}}, i = A, B, C, D \quad (4.15)$$

where C_{j0} , $j = A, B, C, D$, are the mole densities of pure reactants. The condensation of vapor takes place overhead, and a portion of the condensed liquid is purged back to separator to keep the flow rate of the recycle stream at a fixed value. The temperature of the condensed liquid is assumed to be the same as the temperature of the vessel.

The definitions for the variables used in the above model can be found in Table 4.1, with the parameter values given in Table 4.2.

Each of the tanks has an external heat/coolant input. The manipulated inputs to the process are the heat injected to or removed from the five vessels, Q_1 , Q_2 , Q_3 , Q_4 and Q_5 , and the feed stream flow rates to CSTR-2 and CSTR-3, F_4 and F_6 .

The states of the process consist of the concentrations of A , B , C , D in each of the five vessels and the temperatures of the vessels. The state of the process is assumed to be available continuously to the controllers. We consider a stable steady state (operating point), x_s , of the process which is defined by the steady-state inputs Q_{1s} , Q_{2s} , Q_{3s} , Q_{4s} , Q_{5s} , F_{4s} and F_{6s} which are shown in Table 4.3. The steady-state temperatures in the five vessels are the following:

$$T_{1s} = 477.2 \text{ K}, T_{2s} = 476.9 \text{ K}, T_{3s} = 473.4 \text{ K}, T_{4s} = 470.6 \text{ K}, T_{5s} = 478.2 \text{ K}.$$

Table 4.1: Process variables

$C_{A1}, C_{B1}, C_{C1}, C_{D1}$	Concentrations of A, B, C, D in CSTR-1
$C_{A2}, C_{B2}, C_{C2}, C_{D2}$	Concentrations of A, B, C, D in CSTR-2
$C_{A3}, C_{B3}, C_{C3}, C_{D3}$	Concentrations of A, B, C, D in CSTR-3
$C_{A4}, C_{B4}, C_{C4}, C_{D4}$	Concentrations of A, B, C, D in separator
$C_{A5}, C_{B5}, C_{C5}, C_{D5}$	Concentrations of A, B, C, D in CSTR-4
$C_{Ar}, C_{Br}, C_{Cr}, C_{Dr}$	Concentrations of A, B, C, D in F_r
T_1, T_2, T_3, T_4, T_5	Temperatures in each vessel
T_{ref}	Reference temperature
F_3, F_5, F_7, F_8, F_9	Effluent flow rates from each vessel
$F_1, F_2, F_4, F_6, F_{10}$	Feed flow rates to each vessel
F_r, F_{r1}, F_{r2}	Recycle flow rates
H_{vapA}, H_{vapB}	Enthalpies of vaporization of A, B
H_{vapC}, H_{vapD}	Enthalpies of vaporization of C, D
H_{Aref}, H_{Bref}	Enthalpies of A, B at T_{ref}
H_{Cref}, H_{Dref}	Enthalpies of C, D at T_{ref}
$\Delta H_{r1}, \Delta H_{r2}, \Delta H_{r3}$	Heat of reactions 1, 2 and 3
V_1, V_2, V_3, V_4, V_5	Volume of each vessel
Q_1, Q_2, Q_3, Q_4, Q_5	External heat/coolant inputs to each vessel
$C_{pA}, C_{pB}, C_{pC}, C_{pD}$	Heat capacity of A, B, C, D
$\alpha_A, \alpha_B, \alpha_C, \alpha_D$	Relative volatility of A, B, C, D
$C_{A0}, C_{B0}, C_{C0}, C_{D0}$	Molar densities of pure A, B, C, D
T_{A0}, T_{B0}, T_{D0}	Feed temperatures of pure A, B, D
k	Fraction of overhead flow recycled to the reactors

Table 4.2: Parameter values

$F_1 = 7.1 \times 10^{-3}$	m^3/s	$F_r = 0.012$	m^3/s
$F_2 = 8.697 \times 10^{-4}$	m^3/s	$F_{r1} = 0.006$	m^3/s
$F_{r2} = 0.006$	m^3/s	$V_1 = 1$	m^3
$F_{10} = 2.31 \times 10^{-3}$	m^3/s	$V_2 = 1$	m^3
$H_{vapA} = 3.073 \times 10^4$	$J/mole$	$V_3 = 1$	m^3
$H_{vapB} = 1.35 \times 10^4$	$J/mole$	$V_4 = 3$	m^3
$H_{vapC} = 4.226 \times 10^4$	$J/mole$	$V_5 = 1$	m^3
$H_{vapD} = 4.55 \times 10^4$	$J/mole$	$C_{pA} = 184.6$	$J/mole \cdot K$
$\Delta H_{r1} = -1.536 \times 10^5$	$J/mole$	$C_{pB} = 59.1$	$J/mole \cdot K$
$\Delta H_{r2} = -1.118 \times 10^5$	$J/mole$	$C_{pC} = 247$	$J/mole \cdot K$
$\Delta H_{r3} = 4.141 \times 10^5$	$J/mole$	$C_{pD} = 301.3$	$J/mole \cdot K$
$C_{A0} = 1.126 \times 10^4$	$mole/m^3$	$T_{ref} = 450$	K
$C_{B0} = 2.028 \times 10^4$	$mole/m^3$	$T_{A0} = 473$	K
$C_{C0} = 8174$	$mole/m^3$	$T_{B0} = 473$	K
$C_{D0} = 6485$	$mole/m^3$	$T_{D0} = 473$	K
k	0.8		

Table 4.3: Steady-state input values for x_s .

Q_{1s}	$-4.4 \times 10^6 [J/s]$	Q_{2s}	$-4.6 \times 10^6 [J/s]$
Q_{3s}	$-4.7 \times 10^6 [J/s]$	Q_{4s}	$9.2 \times 10^6 [J/s]$
Q_{5s}	$5.9 \times 10^6 [J/s]$	F_{4s}, F_{6s}	$8.697 \times 10^{-4} [m^3/s]$

The process will be under the control of three distributed Lyapunov-based model predictive controllers. The first distributed controller (LMPC 1) will control the values of Q_1 , Q_2 and Q_3 , the second distributed controller (LMPC 2) will decide the values of Q_4 and Q_5 , and the third distributed controller (LMPC 3) will decide the values of F_4 and F_6 . The manipulated inputs for LMPC 1, 2, and 3 will use deviation variables and be described by the sets $u_1^T = [u_{11} \ u_{12} \ u_{13}] = [Q_1 - Q_{1s} \ Q_2 - Q_{2s} \ Q_3 - Q_{3s}]$, $u_2^T = [u_{21} \ u_{22}] = [Q_4 - Q_{4s} \ Q_5 - Q_{5s}]$ and $u_3^T = [u_{31} \ u_{32}] = [F_4 - F_{4s} \ F_6 - F_{6s}]$ which are subject to the constraints shown in Table 4.4.

The alkylation of benzene process model of Eq. 4.1 belongs to the following class of nonlinear systems

$$\dot{x}(t) = f(x(t)) + \sum_{i=1}^3 g_i(x(t))u_i(t) \quad (4.16)$$

where $x(t) \in R^{25}$ denotes the vector of process state variables. The explicit expressions of f ,

Table 4.4: Manipulated input constraints.

$ u_{11} \leq 7.5 \times 10^5 [J/s]$	$ u_{1i} \leq 5 \times 10^5 [J/s], (i = 2, 3)$
$ u_{21} \leq 6 \times 10^5 [J/s]$	$ u_{22} \leq 5 \times 10^5 [J/s]$
$ u_{31} \leq 4.93 \times 10^{-5} [m^3/s]$	$ u_{32} \leq 4.93 \times 10^{-5} [m^3/s]$

g_i ($i = 1, 2, 3$) are omitted for brevity. We assume that the state x of the system is sampled synchronously and the time instants at which state measurements are sampled is indicated by the time sequence $\{t_{k \geq 0}\}$ with $t_k = t_0 + k\Delta$, $k = 0, 1, \dots$ where t_0 is the initial time and $\Delta = 15 \text{ sec}$ is the sampling time.

In the control of the process, u_1 and u_2 are necessary to keep the stability of the closed-loop system, while u_3 can be used as an extra manipulated input to improve the closed-loop performance. We can design a Lyapunov-based controller $h(x) = [h_1(x) \ h_2(x) \ h_3(x)]^T$ to stabilize the closed-loop process. Specifically, $h_1(x)$ and $h_2(x)$ are designed as follows [47]:

$$h_i(x) = \begin{cases} -\frac{L_f V + \sqrt{(L_f V)^2 + (L_{g_i} V)^4}}{(L_{g_i} V)^2} L_{g_i} V & \text{if } L_{g_i} V \neq 0 \\ 0 & \text{if } L_{g_i} V = 0 \end{cases}$$

where $i = 1, 2$, $L_f V = \frac{\partial V}{\partial x} f(x)$ and $L_{g_i} V = \frac{\partial V}{\partial x} g_i(x)$ denote the Lie derivatives of the scalar function V with respect to the vector fields f and g_i ($i = 1, 2$), respectively. The controller $h_3(x)$ is chosen to be $h_3(x) = [0 \ 0]^T$ because the input set u_3 is not needed to stabilize the process. We consider a Lyapunov function $V(x) = x^T P x$ with P being the following weight matrix: $P = \text{diag}([1 \ 1 \ 1 \ 1 \ 10 \ 1 \ 1 \ 1 \ 1 \ 10 \ 1 \ 1 \ 1 \ 1 \ 10 \ 1 \ 1 \ 1 \ 1 \ 10 \ 1 \ 1 \ 1 \ 1 \ 10])^*$. The values of the weights in P have been chosen in such a way that the Lyapunov-based controller $h(x)$ stabilizes the closed-loop system asymptotically and provides good closed-loop performance.

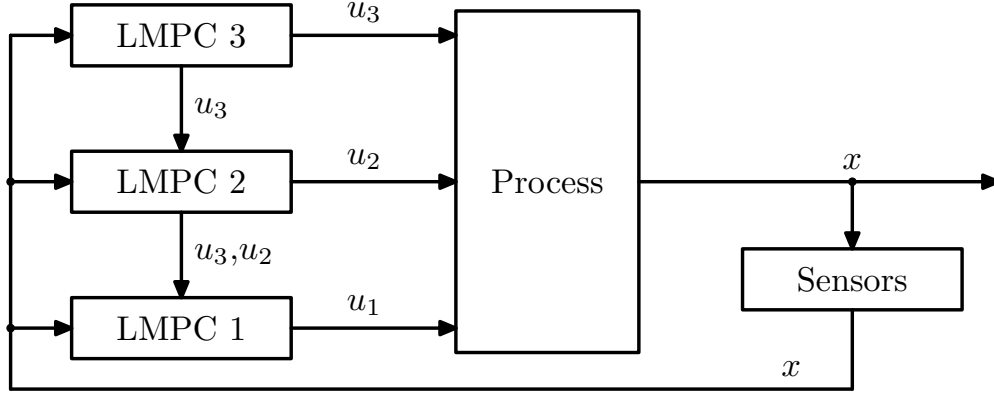


Figure 4.2: Sequential distributed LMPC for the catalytic alkylation of benzene process.

4.3 FDIFTC system design

4.3.1 Fault-free DMPC system design

In this section, we design the fault-free control system for the alkylation process following the sequential distributed Lyapunov-based MPC (LMPC) approach described in [25, 24]. Specifically, for the alkylation process, we design three LMPC controllers to compute u_1 , u_2 , and u_3 , respectively. In the sequential distributed control scheme, the distributed LMPCs communicate in a one-directional manner as shown in Fig. 4.2 in which at each sampling time t_k : 1) all LMPCs receive the state measurement $x(t_k)$ from the sensors; 2) LMPC 3 evaluates the optimal input trajectory of u_3 and sends its future input information to LMPC 2; 3) LMPC 2 evaluates its optimal input trajectory of u_2 and sends its own and LMPC 3's future input information to LMPC 1; 4) LMPC 1 evaluates its optimal input trajectory of u_1 ; and 5) the first step input values of u is sent to its corresponding actuators and the process is repeated at every sampling time.

The sequential DMPC is based on $h(x)$ and the Lyapunov function $V(x)$. Specifically, the

* $diag(v)$ denotes a matrix with its diagonal elements being the elements of vector v and all the other elements being zeros.

distributed LMPCs are based on the following optimization problem:

$$\min_{u_i \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + \sum_{i=1}^3 u_i(\tau)^T R_{ci} u_i(\tau)] d\tau \quad (4.17a)$$

$$\text{s.t. } \dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + \sum_{i=1}^3 g_i(\tilde{x}(\tau)) u_i(\tau) \quad (4.17b)$$

$$u_j(\tau) = h_k(\tilde{x}(b\Delta)), \forall \tau \in [b\Delta, (b+1)\Delta], b = 0, \dots, N-1, \\ j = 1, \dots, i-1 \quad (4.17c)$$

$$u_j(\tau) = u_j^*(\tau|t_k), j = i+1, \dots, 3 \quad (4.17d)$$

$$\tilde{x}(0) = x(t_k) \quad (4.17e)$$

$$u_i(\tau) \in U_i \quad (4.17f)$$

$$\frac{\partial V(x)}{\partial x} g_i(x(t_k)) u_i(0) \leq \frac{\partial V(x)}{\partial x} g_i(x(t_k)) h_i(x(t_k)) \quad (4.17g)$$

where $S(\Delta)$ is the family of piece-wise continuous function with sampling time Δ , the prediction horizon $N = 3$, \tilde{x} is the predicted system trajectory, u_j^* is the future optimal input trajectory calculated by LMPC j , Q_c and R_{ci} are positive definite weighting matrices with the following values: $Q_c = \text{diag}(Q_v)$ with $R_{c1} = \text{diag}([10^{-8} \ 10^{-8} \ 10^{-8}])$, $R_{c2} = \text{diag}([10^{-8} \ 10^{-8}])$, $R_{c3} = \text{diag}([1 \ 1])$ and

$$Q_v = [1 \ 1 \ 1 \ 1 \ 10^3 \ 1 \ 1 \ 1 \ 1 \ 10^3 \ 10 \ 10 \ 10 \ 10 \ 10^4 \ 1 \ 1 \ 1 \ 1 \ 10^3 \ 1 \ 1 \ 1 \ 1 \ 10^3]$$

The optimal solution to this optimization problem is denoted by $u_i^*(\tau|t_k)$, $i = 1, 2, 3$, which is defined for $\tau \in [0, N\Delta]$. Note that in this optimization problem, the constraint of Eq. 4.17c is only active for LMPC 3 and LMPC 2; and the constraint of Eq. 4.17d is only active for LMPC 2 and LMPC 1. The constraint of Eq. 4.17g is used to make sure that each controller has a minimum contribution to the decrease rate of the Lyapunov function which is used to guarantee the closed-loop stability. Once all optimization problems are solved, the manipulated inputs of the distributed LMPC system are defined as follows:

$$u_i^L(t|x) = u_i^*(t - t_k|t_k), i = 1, 2, 3, \forall t \in [t_k, t_{k+1}).$$

Table 4.5: Fault signature shows which residuals are triggered by faults in particular actuators. Note that some signatures overlap (i.e., Q_2 fault signature overlaps with F_4 fault signature and Q_3 fault signature overlaps with F_6 fault signature)

Actuator	Fault Signature
Q_1	T_1
Q_2	T_2
Q_3	T_3
Q_4	T_4
Q_5	T_5
F_4	$T_2, C_{A2}, C_{B2}, C_{C2}$
F_6	$T_3, C_{A3}, C_{B3}, C_{C3}$

The alkylation process under this DMPC scheme with inputs defined by $u_i = u_i^L$, $i = 1, 2, 3$, maintains the same stability region as the Lyapunov-based control law h [25, 24].

4.3.2 Fault Detection and Isolation

We consider control actuator faults that can be detected and isolated by an appropriate nonlinear dynamic filter by observing the evolution of the closed-loop system state. This consideration requires that a fault in a control actuator influences the evolution of at least one of the states. In order to isolate the occurrence of a fault, it is further required that the control actuator in question is the only one influencing a certain set of the system states (i.e., each fault has a unique fault signature), see Table 4.5. For more discussions on systems having isolable structures, see [34, 36].

The DMPC system of Eq. 4.17 is the control configuration for the fault-free system of Eq. 4.16. We first design an FDI scheme to detect faults in this control system. In this FDI scheme, a filter is designed for each state and the design of the filter for the p^{th} , $p = 1, \dots, 25$, state in the system state vector x is as follows:

$$\dot{\hat{x}}_p(t) = f_p(X_p) + \sum_{i=1}^3 g_{ip}(X_p)u_i^L(X_p) \quad (4.18)$$

where \hat{x}_p is the filter output for the p^{th} state, f_p and g_{ip} are the p^{th} components of the vector functions f and g_i , respectively. With a slight abuse of notation, we have dropped the time index in Eq. 4.18 in the control functions and denote $u_i^L(t|x)$ with $u_i^L(x)$, in order to simplify the FDI

definitions. The state X_p is obtained from both the actual state measurements, x , and the filter output, \hat{x}_p , as follows:

$$X_p(t) = [x_1(t), \dots, x_{p-1}(t), \hat{x}_p(t), x_{p+1}(t), \dots, x_{25}(t)]^T.$$

Note that in the filter of Eq. 4.18, the control inputs $u_i^L(X_p)$ are determined by the same LMPC of Eq. 4.17 as applied to the actual process, and are updated at every sampling time (i.e., the sampling time instants $\{t_{k \geq 0}\}$).

The states of the FDI filters are initialized at $t = 0$ to the actual state values; that is, $\hat{x}_p = x_p$. The FDI filters are only initialized at $t = 0$ such that $\hat{x}_p(0) = x_p(0)$. The information generated by the filters provides a fault-free estimate of the process at any time t and allows detection of the faults. For each state associated with a filter, the FDI residual can be defined as:

$$r_p(t) = |\hat{x}_p(t) - x_p(t)|,$$

with $p = 1, \dots, 25$. The residual r_p is computed continuously because $\hat{x}_p(t)$ is known for all t and the state measurement, x , is also available for all t . If no fault occurs, the filter states track the system states. In this case, the dynamics of the system states and the FDI filter states are identical, so $r_p(t) = 0$ for all times. When there is a fault in the system, filter residuals affected directly by the fault will deviate from zero soon after the occurrence of the fault. For more detailed discussion on the properties of the filters, see [34].

Note that due to sensor measurement and process noise, the residuals will be nonzero even without an actuator fault. This necessitates the use of fault detection thresholds so that a fault is declared only when a residual exceeds a specific threshold value, σ_p . This threshold value is chosen to avoid false alarms due to process and sensor measurement noise, but should still be sensitive enough to detect faults in a timely manner so that effective fault-tolerant control can be performed.

The objective of the FDI scheme is to quickly detect an actuator fault when it occurs, and then identify which of the possible different actuator faults has occurred. When a fault occurs, one or more of the filter residuals will become nonzero. Once a residual (r_p) is detected at time t_{σ_p} , the monitoring system will declare a fault alarm. In order to isolate a fault, the system must have an

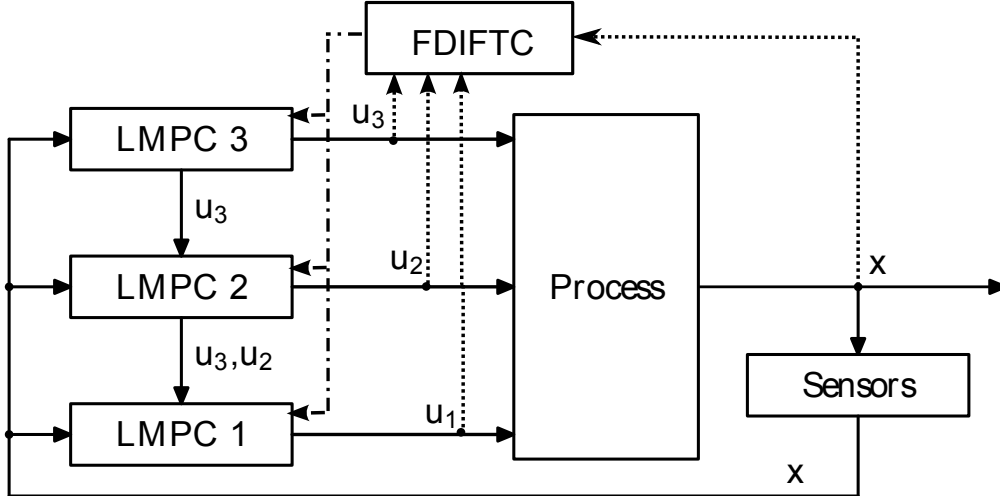


Figure 4.3: Sequential distributed LMPC with FDIFTC system.

isolable structure in which different faults have different fault signatures. In some cases the fault signatures overlap such that a waiting time (Δt_i) is used to confidently distinguish between fault signatures by letting the fault propagate in the system (see Table 4.5 where a Q_2 fault signature overlaps with an F_4 signature). Δt_i is chosen to achieve a trade off between quicker reconfiguration and the need to confidently isolate a fault and is based on the worst case time needed for the slowest actuator fault to develop its fault signature. If a fault is isolated, the FDIFTC system will send the fault information and reconfiguration policy to the distributed controllers to activate the FTC system as shown in Fig. 4.3.

4.3.3 Fault parameter estimation

After a fault has been isolated, the FTC system must know the magnitude of the fault in order to target the corresponding new operating point and properly stabilize the system in the presence of the fault. To simplify the description of the proposed method, we consider faults of constant magnitudes in this work; however, faults with slowly time-varying values can be handled using the proposed FDIFTC method in a straightforward manner.

When a residual (r_p) exceeds its threshold (σ_p), we begin to collect the sampled system states as well as the actual control inputs applied to the system. When the fault is confirmed and isolated, a least square optimization problem is solved to estimate the magnitude of the fault based on the

sampled system states and the actual control inputs. Specifically, we collect the sampled system states, $x(t)$, and record the actual control inputs (i.e., $u_1(t) = u_1^L(t)$, \dots , $u_3(t) = u_3^L(t)$) applied to the system from t_{σ_p} to the fault isolation time ($t_{isolate} = t_{\sigma_p} + \Delta t_i$). The magnitude of the fault (denoted as d) is estimated by solving the following optimization problem:

$$\min_d \sum_{i=0}^M (x(t_f + i\Delta) - \tilde{x}(t_f + i\Delta))^2 \quad (4.19a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g(\tilde{x}(t))(u^L(t) + \tilde{u}) \quad (4.19b)$$

$$\tilde{x}(t_f) = x(t_f) \quad (4.19c)$$

where $\tilde{u} = [0 \ \dots \ d \ \dots \ 0]^T$ is the fault vector, $u^L(t) = [u_1^L(t)^T \ \dots \ u_3^L(t)^T]^T$ is the actual control inputs that have been applied to the closed-loop system from t_{σ_p} to $t_{isolate}$, M is the maximum integer satisfying $M\Delta \leq t_{isolate} - t_{\sigma_p}$, and $x(t_{\sigma_p})$ is the system state at the fault detection time. The solution to the optimization problem of Eq. 4.19 is denoted by d^* , which is the estimate of the actual fault from a least-square point of view.

4.3.4 FTC consideration and strategies

In order to carry out FTC, there must be a backup control configuration for the system under consideration. For the alkylation process, the presence of the control action u_3 brings extra control flexibility to the closed-loop system which can be used to carry out FTC. From extensive simulations, we found that the closed-loop process can also be stabilized using the manipulated input sets $\{u_{11}, u_{12}, u_2, u_3\}$ and $\{u_1, u_2, u_{31}\}$ when the faults in u_{13} and u_{32} are small enough such that the new operating points are close enough to the original operating point. This fact can be taken advantage of to design FTC systems for the alkylation process.

First, we discuss the case that there is a persistent fault d_1 in u_{13} . In this case, we need to design a Lyapunov-based control law $h_2(x)$ which manipulates u_{11} , u_{12} , u_2 and u_3 to stabilize the closed-loop process. The control law $h_2(x)$ in [47] and its expression is omitted for brevity. This control law will be used in the backup distributed LMPC when the fault in u_{13} is detected and isolated. We still design three LMPC controllers in the backup DMPC system. One LMPC is used to manipulate u_{11} and u_{12} , one for u_2 , and the third is used to manipulate u_3 . In this backup

DMPC system, the three LMPCs coordinate their actions to maintain the closed-loop stability. We refer to the LMPC manipulating u_{11} and u_{12} as the backup LMPC 1 and the LMPC manipulating u_2 and u_3 as the backup LMPC 2 and 3, respectively. The three backup LMPCs are also evaluated in sequence. Specifically, the backup LMPC 3 is designed as follows:

$$\min_{u_3 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + \sum_{i=1}^3 u_i(\tau)^T R_{ci} u_i(\tau)] d\tau \quad (4.20a)$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + \sum_{i=1}^3 g_i(\tilde{x}(\tau)) u_i(\tau)$$

$$u_2(\tau) = h_{22}(\tilde{x}(j\Delta)), \quad (4.20b)$$

$$[u_{11}(\tau) \ u_{12}(\tau)]^T = h_{21}(\tilde{x}(j\Delta)), \quad (4.20c)$$

$$\forall \tau \in [j\Delta, (j+1)\Delta), \ j = 0, \dots, N-1$$

$$u_{13}(\tau) = 0 \quad (4.20d)$$

$$\tilde{x}(0) = x(t_k) \quad (4.20e)$$

$$u_3(\tau) \in U_3 \quad (4.20f)$$

$$\frac{\partial V(x)}{\partial x} g_3(x(t_k)) u_3(0) \leq \frac{\partial V(x)}{\partial x} g_3(x(t_k)) h_{23}(x(t_k)). \quad (4.20g)$$

The solution to the optimization problem of Eq. 4.20 is denoted $u_3^{b*}(t|t_k)$. The backup LMPC 2

optimizes u_2 and is designed as follows:

$$\min_{u_2 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + \sum_{i=1}^3 u_i(\tau)^T R_{ci} u_i(\tau)] d\tau \quad (4.21a)$$

$$\begin{aligned} \dot{\tilde{x}}(\tau) &= f(\tilde{x}(\tau)) + \sum_{i=1}^2 g_i(\tilde{x}(\tau)) u_i(\tau) \\ &\quad + g_3(\tilde{x}(\tau)) u_3^{b*}(\tau) \end{aligned} \quad (4.21b)$$

$$[u_{11}(\tau) \ u_{12}(\tau)]^T = h_{21}(\tilde{x}(j\Delta)), \quad \forall \tau \in [j\Delta, (j+1)\Delta), \quad (4.21c)$$

$$j = 0, \dots, N-1 \quad (4.21d)$$

$$u_{13}(\tau) = 0 \quad (4.21e)$$

$$\tilde{x}(0) = x(t_k) \quad (4.21f)$$

$$u_2(\tau) \in U_2 \quad (4.21g)$$

$$\frac{\partial V(x)}{\partial x} g_2(x(t_k)) u_2(0) \leq \frac{\partial V(x)}{\partial x} g_2(x(t_k)) h_{22}(x(t_k)). \quad (4.21h)$$

The solution to the optimization problem of Eq. 4.21 is denoted $u_2^{b*}(t|t_k)$. The backup LMPC 1 optimizes u_{11} and is designed as follows:

$$\min_{u_1 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + \sum_{i=1}^3 u_i(\tau)^T R_{ci} u_i(\tau)] d\tau \quad (4.22a)$$

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t)) + g_1(\tilde{x}(t)) [u_{11}(t) \ u_{12}(t) \ 0]^T + \sum_{i=2}^3 g_i(\tilde{x}(t)) u_i^{b*}(t) \quad (4.22b)$$

$$\tilde{x}(t_k) = x(t_k) \quad (4.22c)$$

$$u_1(t) \in U_1 \quad (4.22d)$$

$$u_{13}(t) = 0 \quad (4.22e)$$

$$\frac{\partial V_2(x)}{\partial x} g_1(x(t_k)) [u_{11}(t) \ u_{12}(t) \ 0]^T \leq \frac{\partial V_2(x)}{\partial x} g_1(x(t_k)) [h_{21}(x(t_k))^T \ 0]^T. \quad (4.22f)$$

The solution to the optimization problem of Eq. 4.22 is denoted $u_{11}^{b*}(t|t_k)$. The control inputs of

the backup DMPC system are defined as follows:

$$\begin{aligned}
[u_{11}^b(t) \ u_{12}^b(t)]^T &= [u_{11}^{b*}(t|t_k) \ u_{12}^{b*}(t|t_k)], \ \forall t \in [t_k, t_{k+1}) \\
u_{13}^b(t) &= 0, \ \forall t \\
u_2^b(t) &= u_2^{b*}(t|t_k), \ \forall t \in [t_k, t_{k+1}) \\
u_3^b(t) &= u_3^{b*}(t|t_k), \ \forall t \in [t_k, t_{k+1})
\end{aligned}$$

The fault-free closed-loop system of Eq. 4.16 under the backup DMPC control with inputs defined by $u_{11} = u_{11}^b$, $u_{12} = 0$, $u_2 = u_2^b$, and $u_3 = u_3^b$ maintains practical stability of the closed-loop system because of the Lyapunov-based constraints of Eqs. 4.20g, 4.21h, and 4.22f [24].

When a fault in u_{13} is detected, isolated and the magnitude of the fault is estimated, suitable FTC strategies can be carried out to keep the closed-loop system state within a desired operating region. Because of the fault, the operating point of the fault-free system may not be achievable because of the input constraints and the system structure. In this case, we may operate the system at a new operating point within the desired operating region. To determine the new operating point x_s , we propose to solve an optimization problem. Specifically, when the fault is d_1^* , the new operating point, x_s , is obtained by solving the following optimization problem:

$$\min_{x_s, u_s} x_s^T W x_s \quad (4.23a)$$

$$\text{s.t. } f(x_s) + g(x_s)(u_s + \tilde{u}) = 0 \quad (4.23b)$$

$$u_s + \tilde{u} \in U \quad (4.23c)$$

$$x_s \in X \quad (4.23d)$$

where W is a positive weighting matrix, $\tilde{u} = [0 \ \cdots \ d_1^* \ \cdots \ 0]^T$ and X denotes the desired operating state-space region. The objective of the above optimization problem is to find an operating point within the desired operating state space region such that the distance (measured by weighted Euclidean norm) between the new operating point and the original (fault-free) operating point is minimized. We assume that the optimization problem of Eq. 4.23 is always feasible which implies that we can always find the new operating point x_s and the corresponding new steady-state control input values $u_s = [u_{1s}^T \ u_{2s}^T \ u_{3s}^T]^T$.

Once the fault is isolated, the FTC strategy would shut down the control action of u_{13} and reconfigure the DMPC algorithms of Eq. 4.17 to the backup DMPC of Eqs. 4.20-4.22 to manipulate u_{11} , u_{12} , u_2 , and u_3 to control the process. In order to maintain the stability of the closed-loop system, the designs of the three backup LMPCs and the design of $h_2(x)$ needs to be updated with the new operating point and the corresponding new steady-state control input values; as well as being updated with the fault magnitude information. Note that the proposed method is only one of many possible approaches to determine the new operating point in the case of a fault. The basic idea of the proposed method is to find a new operating point that stays as close as possible to the original operating point.

Next, we consider the case that there is a persistent fault d_3 in u_{32} . In this case, if the fault is detected and isolated in a reasonable time frame, it is possible to switch off the faulty portion of LMPC 3 and only use u_1 , u_2 , and u_{31} in the control system of Eq. 4.17. When u_{32} is switched off from the closed-loop system, u_{32} is set to the fault value (i.e., $u_{32} = d_3$). In order to maintain the stability of the closed-loop system, the design of LMPC 1, 2, 3, and $h(x)$ will be updated with the new operating point, corresponding to the new steady-state control input values, and updated with the fault magnitude information (i.e., $u_{32} = d_2$). The control inputs determined by the updated LMPC 1,2, and 3 will be referred to as $u'_1(x)$, $u'_2(x)$, and $u'_3(x)$. This FTC strategy will maintain the closed-loop stability if implemented quickly enough such that the state of the closed-loop system is still within the stability region of the backup controllers and parameter estimation is sufficiently accurate, however, the performance of the closed-loop system may degrade to some extent.

However, when there is a fault in u_{11} , or u_{12} or u_2 or u_{31} , it may be impossible to successfully carry out FTC without activating backup actuators within the DMPC systems for the alkylation process considered in this work.

The FTC switching rules for the alkylation process within the DMPC system of Eq. 4.17 are described as follows:

1. When a fault in the actuator associated with u_{32} is isolated at t_f , the FTC switching rule is:

$$u_1(t) = \begin{cases} u_1^L(x), & t \leq t_f \\ u_1''(x), & t > t_f \end{cases} \quad (4.24a)$$

$$u_2(t) = \begin{cases} u_2^L(x), & t \leq t_f \\ u_2''(x), & t > t_f \end{cases} \quad (4.24b)$$

$$u_3(t) = \begin{cases} u_3^L(x), & t \leq t_f \\ \begin{bmatrix} u_{31}''(x) \\ d_3 \end{bmatrix}, & t > t_f \end{cases} \quad (4.24c)$$

$$(4.24d)$$

2. When a fault in the actuator associated with u_{13} is detected at t_f , the FTC switching rule is:

$$u_1(t) = \begin{cases} u_1^L(x), & t \leq t_f \\ \begin{bmatrix} u_{11}^b(x) \\ u_{12}^b(x) \\ d_1 \end{bmatrix}, & t > t_f \end{cases} \quad (4.25a)$$

$$u_2(t) = \begin{cases} u_2^L(x), & t \leq t_f \\ u_2^b(x), & t > t_f \end{cases} \quad (4.25b)$$

$$u_3(t) = \begin{cases} u_3^L(x), & t \leq t_f \\ u_3^b(x), & t > t_f \end{cases} \quad (4.25c)$$

4.4 Simulation results

In this section, various simulations are presented with the goal of showing the abilities of the fault detection/isolation and fault tolerant control system along with its limitations. First, we demonstrate the closed-loop system poor performance upon the triggering of an actuator fault with no fault tolerant control implemented. In the second simulation we again trigger the same

fault and demonstrate the timely fault detection and isolation of the fault and triggering of the fault tolerant control system to reconfigure the control system to maintain stability of the plant with a persistent actuator fault present. Varying levels of recovery are possible after isolation of an actuator fault depending on the robustness of the remaining control structure, and the speed and flexibility of the FDI/FTC system.

4.4.1 No Fault Tolerant Control Implementation

The first two plots presented, show the trajectory of the plant under no-fault conditions. In Figs. 4.4 and 4.5 we see the plant's temperatures (T) and ethylene concentrations (C_B) begin near the steady state (dotted line) and are considered stabilized around the steady state by the 200 *min* mark. We found that besides vessel temperature, focusing only on the ethylene concentration provided the necessary confidence in demonstrating and isolating actuator faults for this particular plant. Since in this particular process (see Table 4.5) the Q_2 actuator and the F_4 actuator partially overlap in terms of their fault signatures since they both trigger the vessel 2 temperature residual (r_{T_2}) and the difference being that the F_4 also triggers the concentration residuals. Similarly a Q_3 fault overlaps with a F_6 fault. In simulations ethylene (C_b) was consistently the first of the concentration residuals to respond from a flow actuator fault (F_2 and F_4). As such it is sufficient to monitor the temperatures and each vessel's ethylene concentration in order to properly isolate an actuator fault. In simulations where a fault is considered, the unknown actuator fault is triggered at the 200 *min* mark and the fault is set to +50% of its maximum actuation, unless written otherwise. Noise was introduced to the closed-loop system as process noise and measurement noise.

The first simulation considered has a fault triggered in the heat actuator of vessel 3 (Q_3) that shows the closed-loop system moving quickly away from the target steady-states in Fig. 4.6. The Q_3 actuator fault is triggered at 200 *min* and increased the heat delivered to vessel 3 where the first residual to consistently exceed its threshold for Δt_i is vessel 3 temperature (r_{T_3}) at 202 *min* shown in Fig. 4.7. The residual response is consistent with the plant model filter design where only the filter states directly associated with the fault will show an immediate deviation as shown in Eq. 4.1. Figure 4.6 shows the temperature in vessel 3 (T_3) increasing beyond its target shortly after initiation of the heat actuator fault and the fault manifesting in vessel 4 temperature after the 310 *min* mark when no fault tolerant control is implemented. The final cost of the simulation

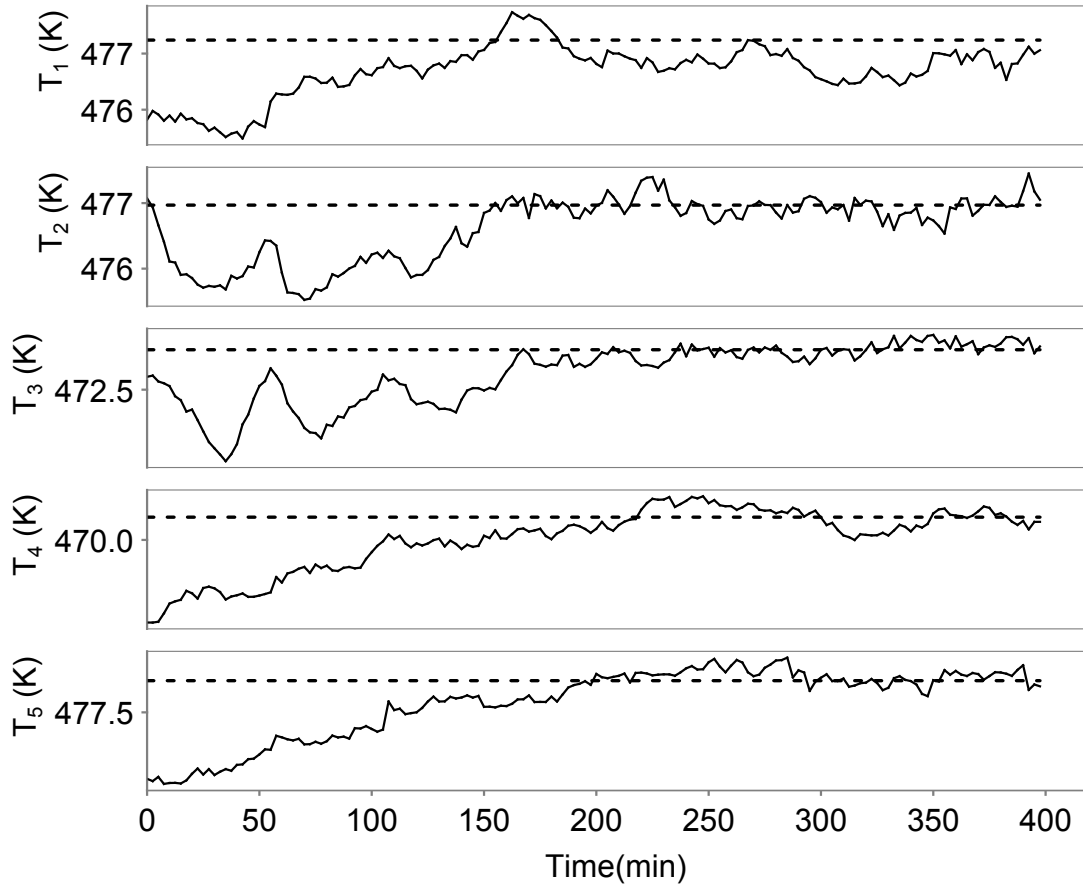


Figure 4.4: Temperatures trajectories for the five vessels under normal fault free operation. Dotted line represents target operating point. The process reaches steady-state conditions around 200 *min*.

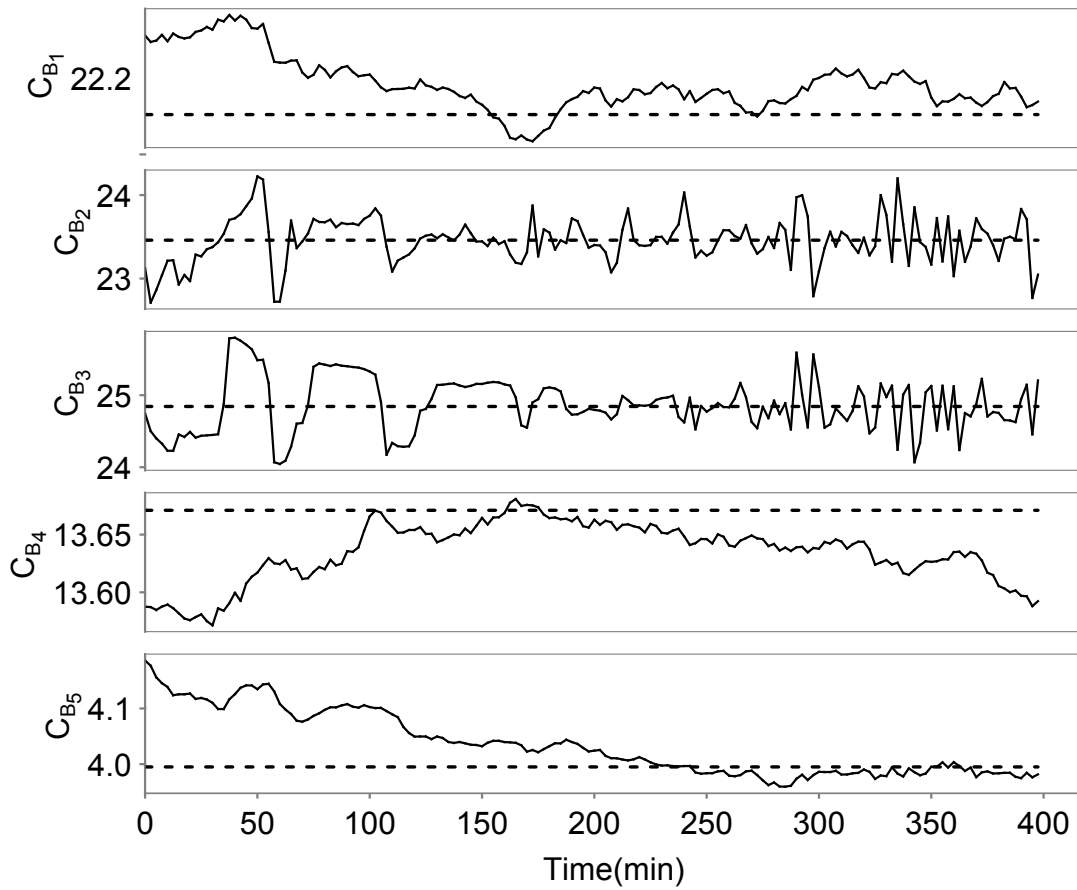


Figure 4.5: Trajectories of ethylene concentration ($mole/m^3$) for the five vessels under normal fault free operation. Dotted line represents target operating point. The process reaches steady-state after 200 *min*.

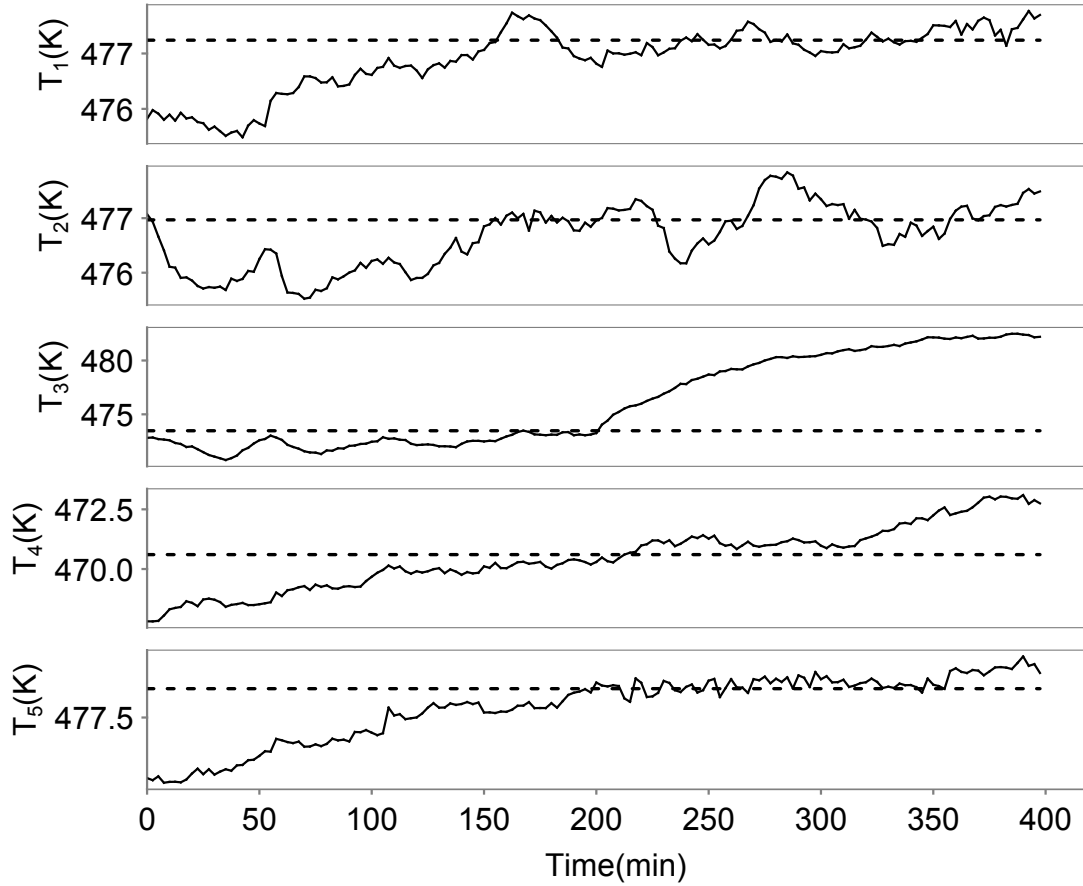


Figure 4.6: Temperature trajectories of the five vessels after triggering a Q_3 fault at time 200 min with no fault tolerant control. Note that the vessel 3 disturbance eventually propagates downstream to vessel 4 and 5 after 310 min and 360 min , respectively.

without fault tolerant control is $9.9 \times 10^7 \text{ units}$.

4.4.2 FTC of a Q_3 heat actuator fault

In the next example we look at how the fault tolerant control system responds to the same heat actuator (Q_3) fault at 200 min . The fault's first appearance is most evident in the residual plot in Fig. 4.8 where vessel 3 temperature residual spikes upward after 202 min (t_{σ_p}). At this time the fault isolation system performs two actions, first it begins monitoring the residuals for a consistent fault signature where the appropriate residuals exceed their thresholds for a specified amount of

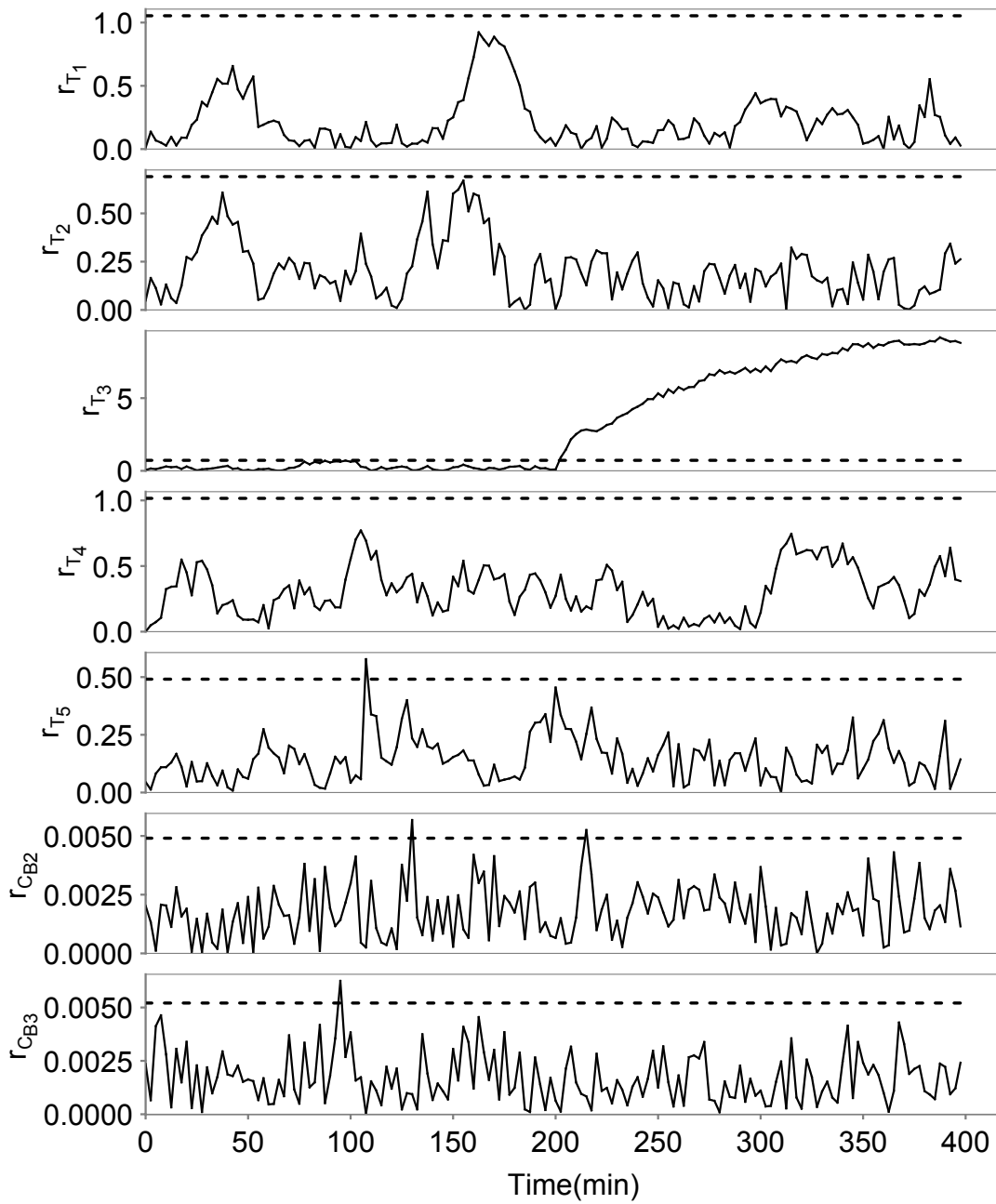


Figure 4.7: Residual plots of key isolation residuals showing residual pattern upon triggering a fault in the heat actuator to the third vessel (Q_3) with no fault tolerant control. Note how only the residual associated with the temperature of vessel three (r_{T_3}) is severely affected.

time (Δt_i) and the fault is isolated. The second action after detecting a possible fault is to begin logging plant states and controller action in order for successful fault estimation to be achieved as presented in Section 4.3.3. The isolation time ($\Delta t_i = 10 \text{ plantsteps} = 150 \text{ sec}$) was determined from various simulations by initiating a low magnitude fault and recording the necessary time for a consistent fault signature. The value was chosen long enough to confidently isolate to a certain degree of certainty and short enough so that the fault tolerant control system can stabilize the plant by reconfiguring the control system while the plant remains in the stability region of the reconfigured control system. A low magnitude fault was used as these typically have the slowest propagation within the system and represents a worst case in terms of isolation time. In the event of losing a fault signature (i.e., the corresponding residual recedes below the threshold) within the isolation time Δt_i , the fault isolation process is reset.

At the end of the isolation time ($t_{\sigma_p} + \Delta t_i = 202 \text{ min} + 150 \text{ sec} = 205 \text{ min}$) the Q_3 fault is isolated and the magnitude is correctly estimated at 50% of maximum actuation. This information is used to reconfigure the control system to account for the persistent disturbance. The successful reconfiguration is obvious at the 205 *min* mark in Figs. 4.8 and 4.9 where the vessel three temperature residual (r_{T_3}) dives down below the threshold and the temperature (T_3) returns to its steady state. But note that after isolation the residuals do not provide useful information unless further reconfiguration strategies are built into the fault tolerant control system. The size of the small spike in T_3 is directly related to the isolation time but it is required in order to confidently isolate when fault signatures have overlapping residual patterns. Reconfiguration in this case is due to the flexibility in the control system to ramp up the F_6 flow actuator to compensate for the problem with Q_3 . The final cost of the simulation with fault tolerant control is $2.2 \times 10^7 \text{ units}$.

4.4.3 FTC of an F_4 flow actuator fault

In the very last example we look at a fault in the flow actuator to vessel two (F_4) at 100% maximum actuation which introduces pure ethylene (C_b) into vessel 2, with no fault tolerant control. Because of the structure of the plant we expect the fault to affect more than one residual, in fact all residuals associated with vessel 2. In Fig. 4.10, we see that the pattern of the residual for concentration of ethylene in vessel 2 changes shortly after the fault is triggered in F_4 . In this example with no fault tolerant control, the fault propagates and we see that the temperatures for tank 2 and 3 begin to

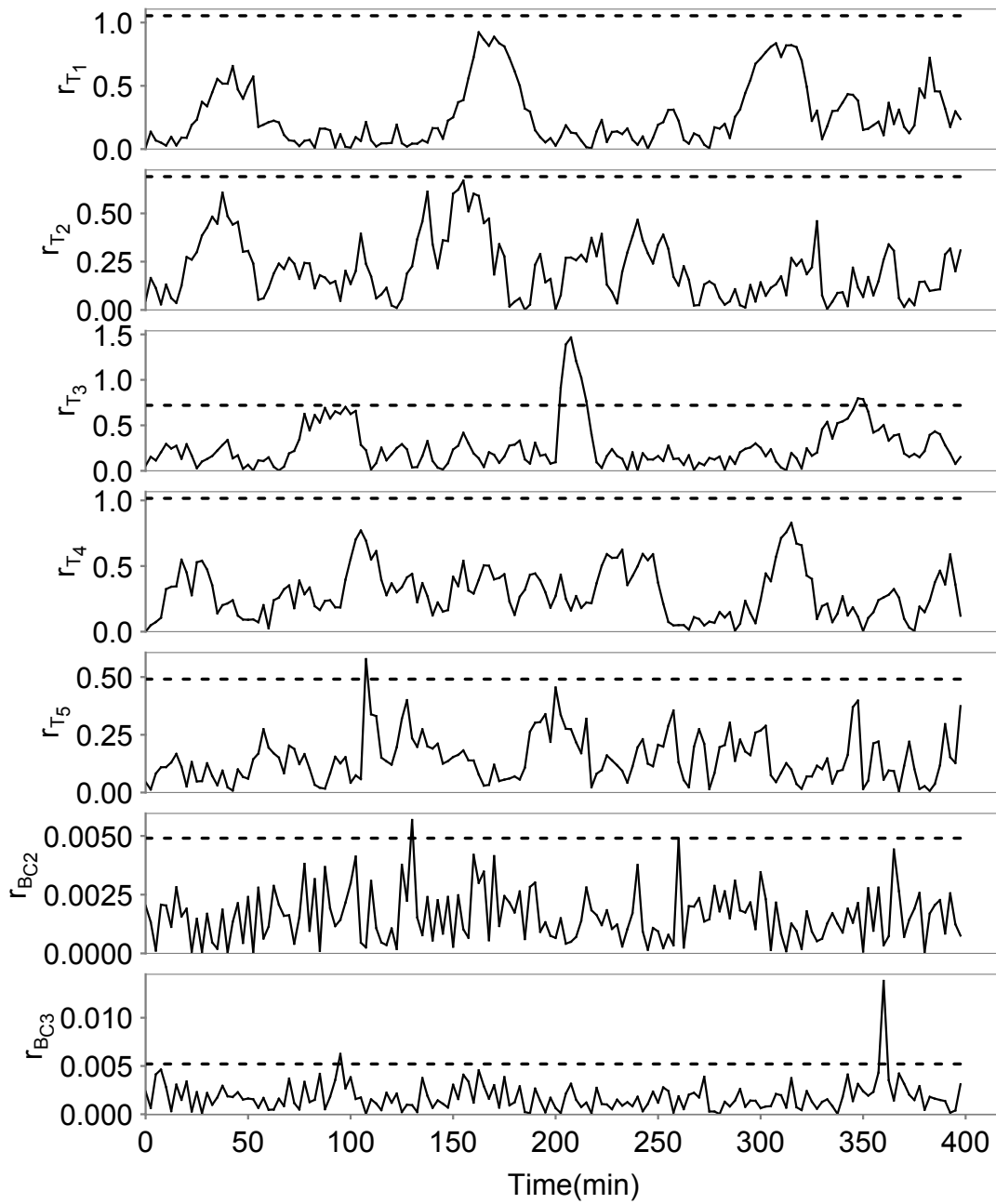


Figure 4.8: Residual plots of key isolation residuals showing residual pattern upon triggering a fault in the heat actuator to the third vessel (Q_3) and using FTC at time 205 *min*. Note how residual r_{T3} trajectory changes immediately after reconfiguration.

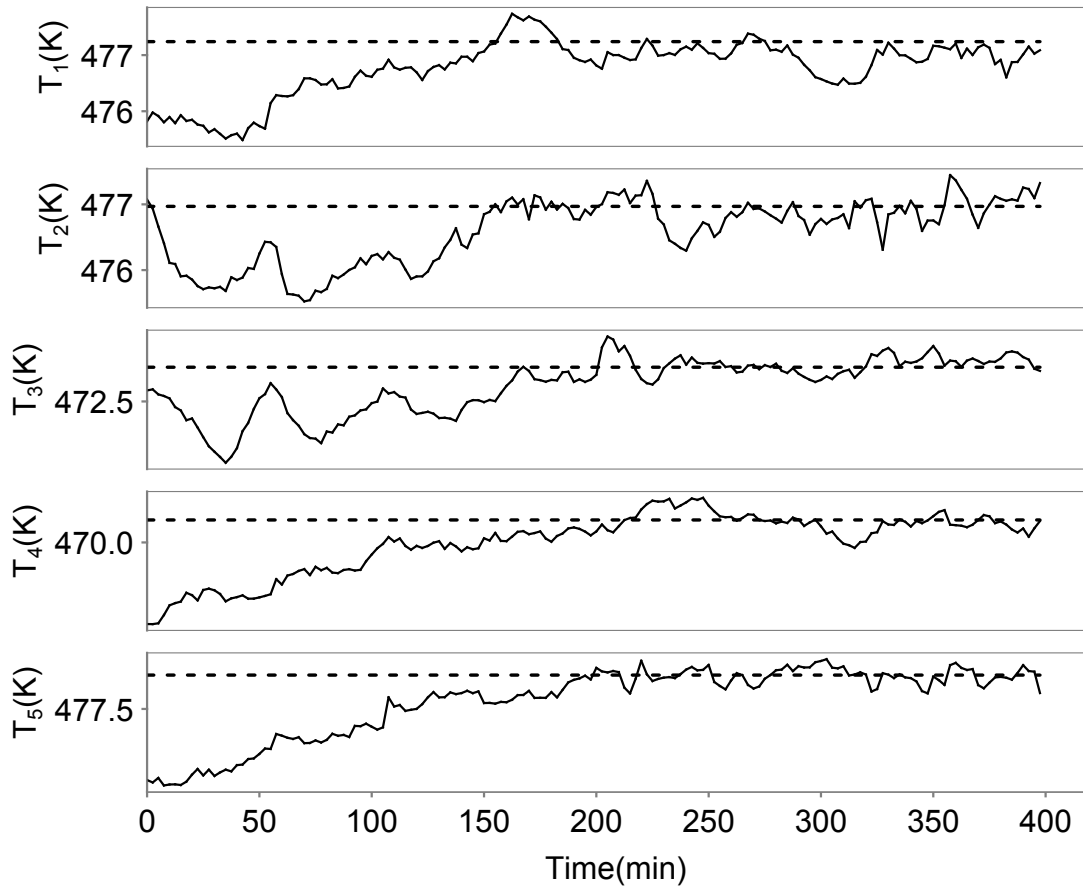


Figure 4.9: Temperature trajectories of the five vessels after triggering a Q_3 actuator fault at 200 min and achieving fault isolation at 205 min . The small peak above the threshold in T_3 from $200 - 210 \text{ min}$ is the result of the actuator fault.

change after 200 *min* in Fig. 4.11.

In the last set of figures we implement the appropriate FTC strategy after isolation of a flow actuator fault in vessel 2 (F_4). After the fault is triggered at 200 *min* the temperature in vessel 2 moves away from the target steady-state (Fig. 4.14). The first residual to trigger monitoring is the ethylene concentration in tank 2 (r_{CB_2}) at 200 *min* (Fig. 4.15). The fault is isolated and FTC reconfiguration is initiated at time 203 *min* where we see that the vessel 2 temperature plot begin to shift back towards the original steady-state as it did in Fig. 4.11. Comparing the temperatures and ethylene concentration plots under fault tolerant control and no fault tolerant control, the difference is minor. But comparing the cost for the no fault tolerant control simulation (fig. 4.12) with a cost of 6.3×10^7 *units* and the fault tolerant control simulation (fig. 4.13) with a final cost of 5.3×10^7 *units* shows a significant gain and is partly due to a reduction in wasted control action (please compare Figs. 4.12 and 4.13).

In the case where an actuator fault occurs in vessel one, four, or five, the fault will be properly isolated and estimated, but due to the plant structure, there does not exist a way to compensate for the lost actuation and persistent disturbance. Also due to the persistent fault and the structure of the process, the original target operating point is not accessible anymore, and the new target steady-state is chosen as to remain as close to the original target with the persistent fault present.

The simulations were carried out using Java programming language on a Pentium 3.20 *GHz* computer. The optimization problems were solved using the open source interior point optimizer Ipopt [53].

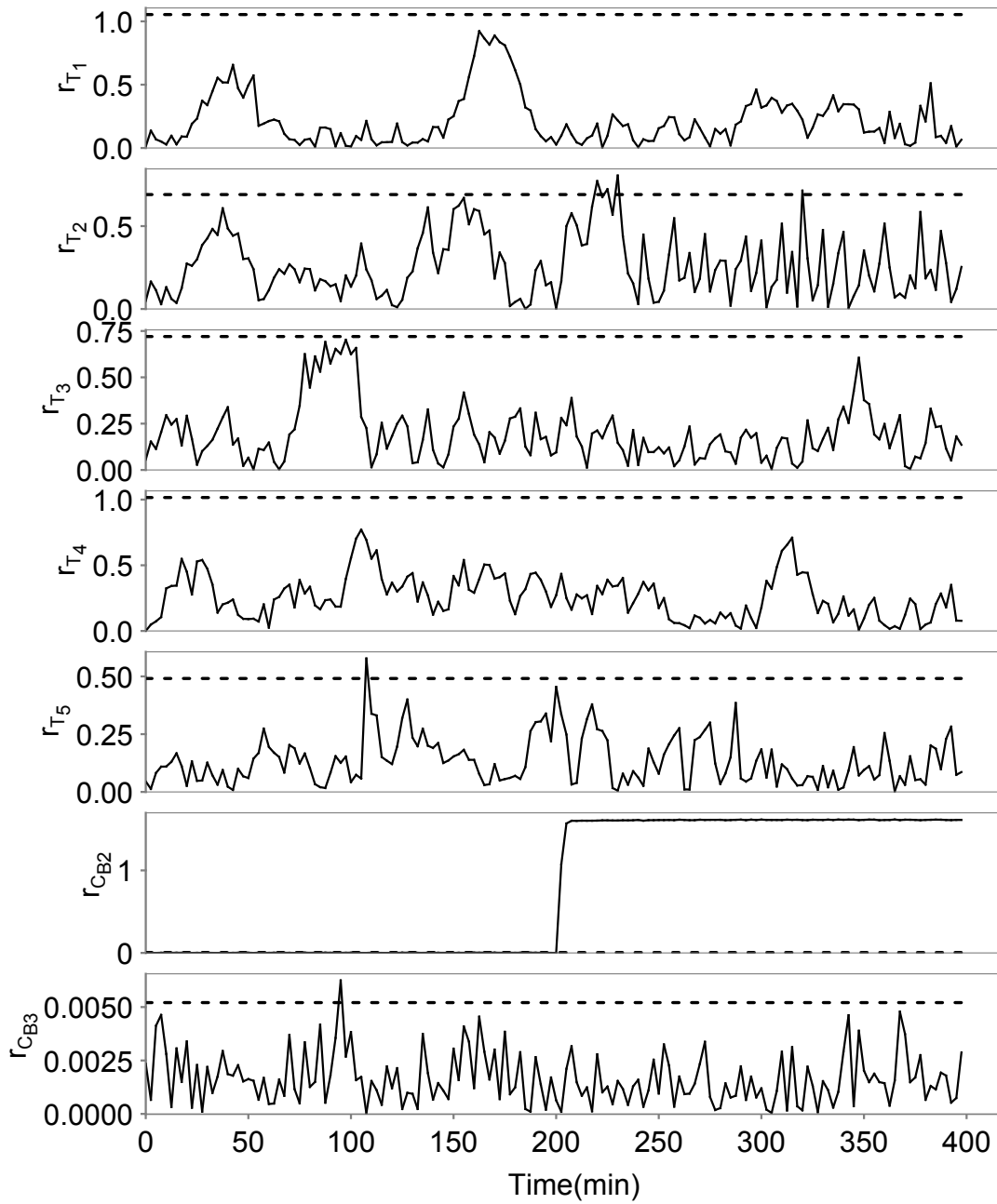


Figure 4.10: Residual plots after initiating a flow actuator fault in tank 2 (F_4) with no fault tolerant control. Note that this fault causes a shift in the residuals for tank 2 ethylene concentration at 200 min.

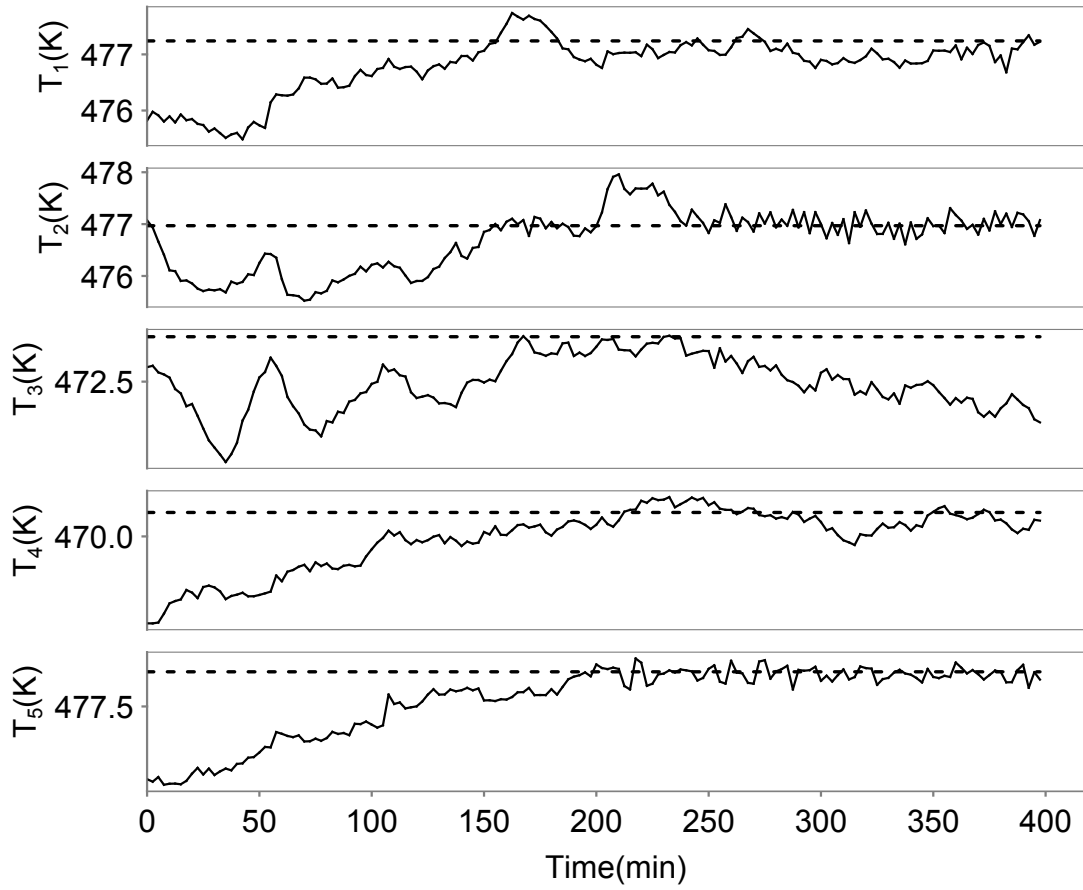


Figure 4.11: Temperature trajectories of the five vessels after triggering a F_4 fault at time 200 *min* with no fault tolerant control.

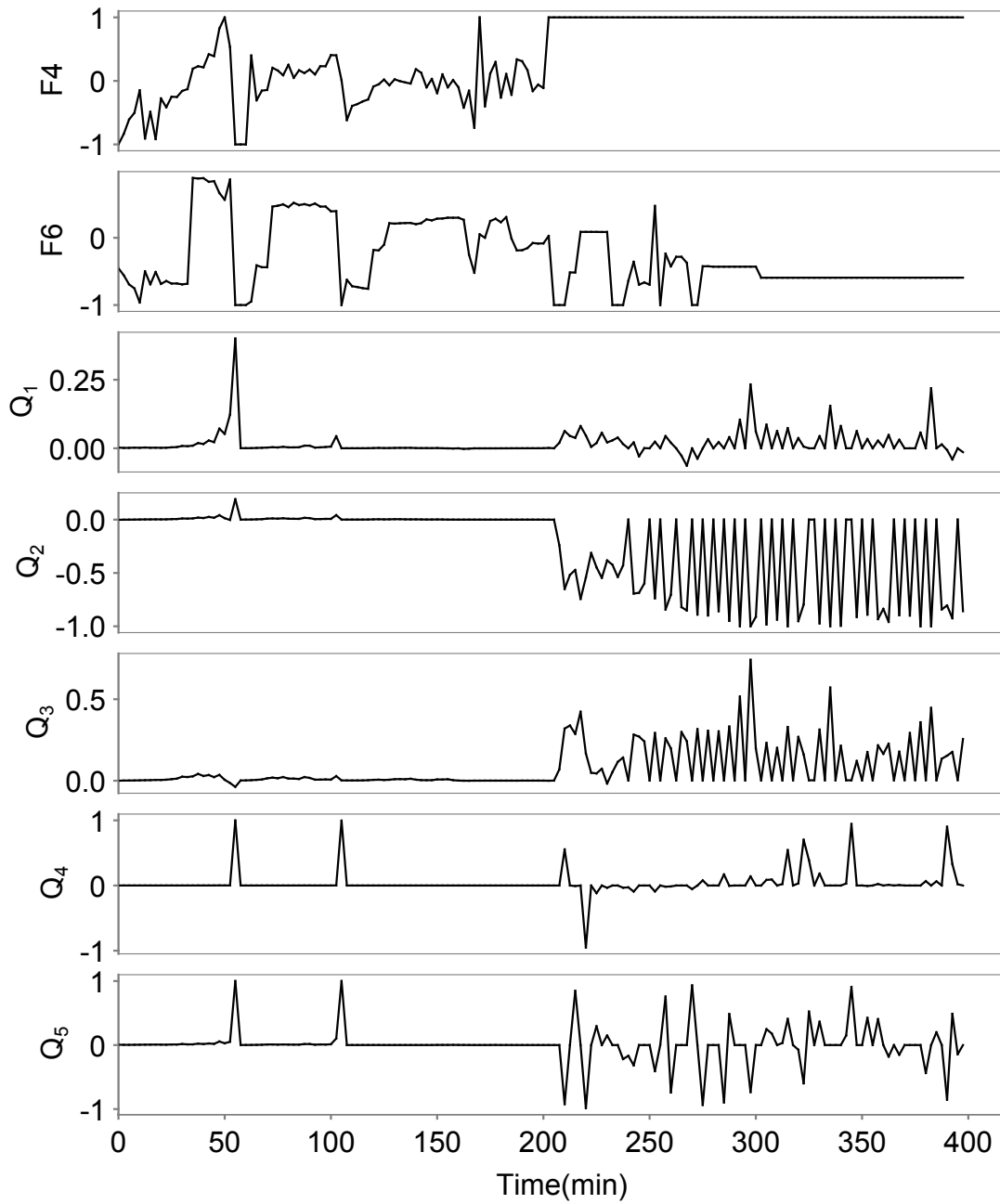


Figure 4.12: Manipulated input trajectories after initiating a flow actuator fault in tank 2 (F_4) with no fault tolerant control. Final cost 6.7×10^7 units. Units of F_4 are m^3/s and Q_1, \dots, Q_5 are J/s ; all inputs are scaled to be in the range of $[-1, 1]$ using the values of Table 4.4.

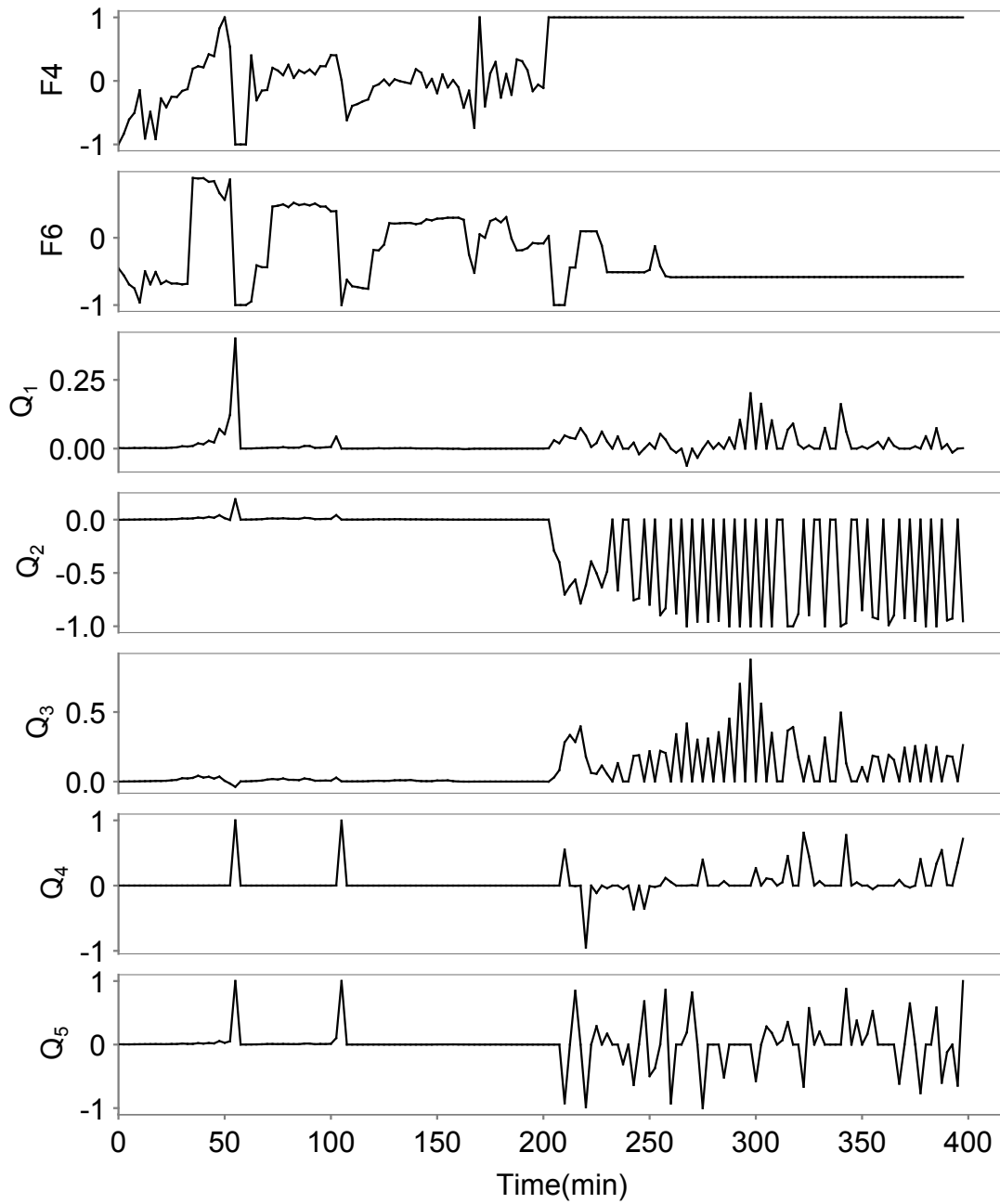


Figure 4.13: Manipulated input trajectories after initiating a flow actuator fault in tank 2 (F_4) with fault tolerant control. Final cost 5.3×10^7 units. Units of F_4 are m^3/s and Q_1, \dots, Q_5 are J/s ; all inputs are scaled to be in the range of $[-1, 1]$ using the values of Table 4.4.

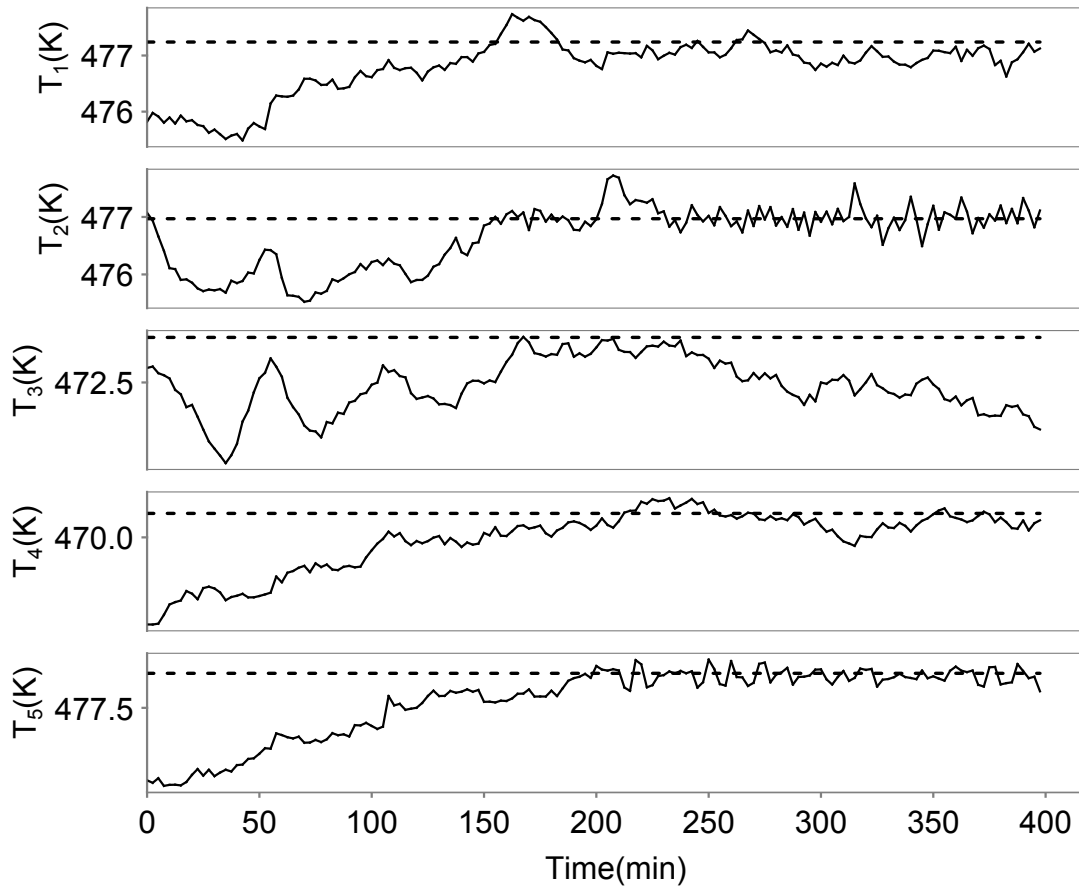


Figure 4.14: Temperature trajectories of the five vessels after triggering a F_4 fault at time 200 *min* with FTC reconfiguration. Note smaller deviation peak in T_2 compared to no FTC implementation in fig. 4.11

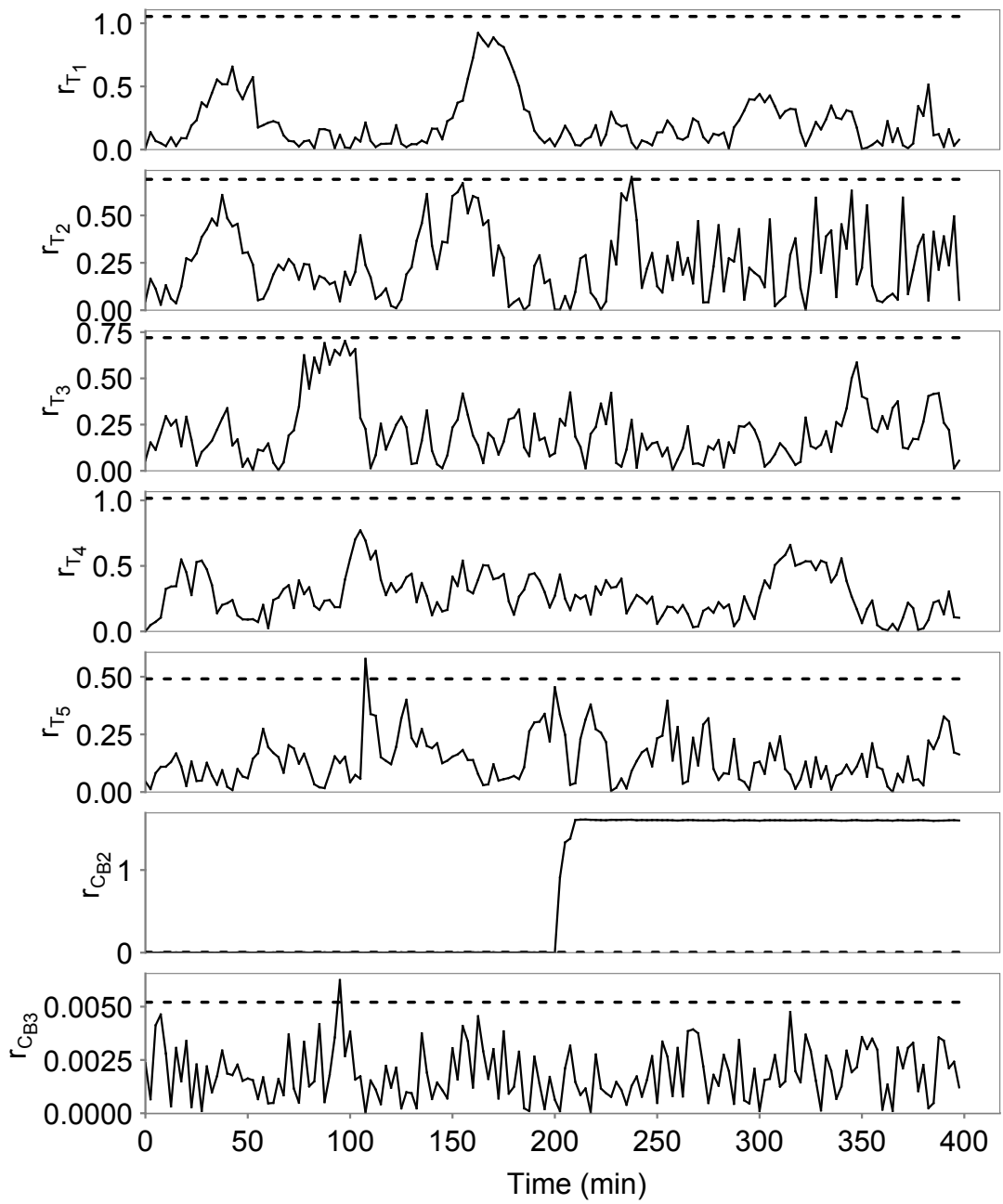


Figure 4.15: Residual plots after initiating a flow actuator fault in tank 2 (F_4) with fault tolerant control.

4.5 Conclusion

In this chapter, we focused on fault detection, isolation and fault tolerant control of an alkylation with benzene process under distributed model predictive control in the presence of an unknown actuator fault. In order to achieve the objectives of closed-loop stability and optimal plant operation, methods for quick fault detection and isolation were necessary such that the faults perturbation had not yet pushed the plant state outside the reconfigured control system's stability region. In addition accurate fault estimation and optimal recalculation of state and input targets was necessary to maintain optimal plant operation in terms of cost. We demonstrated that FTC reconfiguration benefits were most visible in the total operating cost, where controller action no longer wasted energy by under- and over- compensating for an unresponsive and disruptive actuator.

Chapter 5

Monitoring and Retuning of Low-Level PID Control Loops

5.1 Introduction

In general, the calculation of the optimal input trajectories for the manipulated inputs via MPC, the dynamics of the corresponding control actuators that will implement the control actions computed by the MPC are neglected and the MPC-computed control actions are assumed to be directly implemented by the control actuators. However, in practice, these control actuators have their own specific dynamics. As a result of this, there are always discrepancies (i.e., time lags, magnitude differences, etc.) between the actual control actions applied to the process by the control actuators and the control actions requested by the MPC. To mitigate the influence of these discrepancies in closed-loop performance, PID controllers (typically called “low-level” PID controllers) are usually implemented on the control actuators to regulate the outputs of the actuators at the values requested by the MPC [2]. The representation of this added extra layer of the PID controllers around the control actuators is shown in Fig. 5.1. In this case, the tuning of the PID controllers is critical for the overall control actuator and closed-loop system performance. An actuator with a well-tuned PID controller can effectively implement the actions requested by the MPC; whereas, an actuator with a poorly-tuned PID controller may reduce the performance of the closed-loop system dramatically or may even cause instability of the closed-loop system.

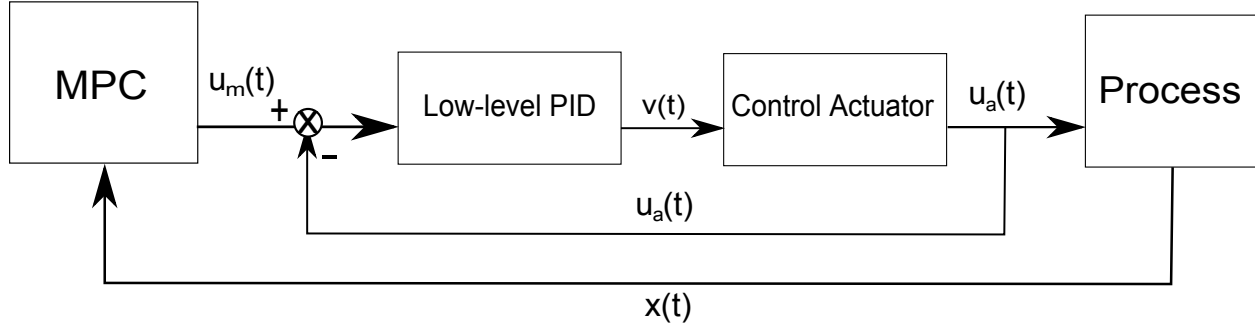


Figure 5.1: Closed-loop system with MPC as advanced model-based controller and low-level PID controller implemented to regulate the control actuators.

Monitoring the performance of low-level PID loops provides the motivation for this work. With respect to previous works on the subject, there is indeed a plethora of techniques discussed in the literature on monitoring of the performance and tuning of PID controller parameters. With respect to tuning, methods such as Ziegler-Nichols [59], Cohen-Coon [9], internal model control [46], pole placement [54], and others have been widely used to tune PID controller parameters based on either the estimated plant's transfer function or experimentally-obtained step response and/or frequency response curves. Gain scheduling [43, 58] has also been developed to allow PID controllers to be able to self-tune to accommodate changing operating conditions. Multiple works have also been published on automatic retuning of PID parameters based on the current performance of the PID controller and on-line system identification [48, 4, 49]. On the monitoring front, [13] and [39] provide a survey of available monitoring techniques. Specifically, minimum variance control [17] has been developed as a tool to assess PID performance, while [51], and [50] utilize statistical process control (SPC) to monitor and provide performance criteria to assess the performance of PID controllers. In another work [45], a monitoring scheme was proposed to determine poor tuning/faults using principal component analysis (PCA) and neural networks. One common feature in all of the works in the PID monitoring field mentioned above is the assumption that measurements of the output of the PID-controlled loop are available, this leaves isolation as a trivial exercise as the badly tuned actuator response can be easily discerned from the expected response.

Motivated by the above considerations, we address the problem of real-time monitoring and retuning of low-level PID controllers in the case where the measurement of the actual control action implemented on the process is unavailable. Specifically, we present a method for monitoring the PID performance via a model-based FDI method [32, 34] coupled with real-time process measurements.

Using an estimated transfer function model of the control actuators, model-based FDI can be used to detect the discrepancies between the expected actuation level and the actual actuation level performed by the control actuators. Based on the patterns of the residuals, a poorly-tuned actuator can be isolated and retuned accordingly. An example of a nonlinear reactor-separator process under MPC control with low-level PID controllers around the control actuators is used to demonstrate the approach.

5.2 Preliminaries

5.2.1 Class of Nonlinear Systems

In this work, we consider non-linear process systems with constraints on the inputs described by the following state-space model:

$$\dot{x}(t) = f(x(t)) + G(x(t))u_a(t) + w(t) \quad (5.1)$$

where $x(t) \in R^{n_x}$ is an n_x -element column vector representing n_x states of the system, $u_a(t) \in U \subseteq R^{m_u}$ is an m_u -element column vector representing m_u inputs to the system, and $w(t) \in W \subseteq R^{n_x}$ is an n_x -element column vector representing the process noise to the system. U is a convex set, $f(\cdot)$ is a non-linear sufficiently smooth vector function, and $G(\cdot)$ is a $n_x \times m_u$ matrix whose elements are sufficiently smooth functions that relate the j^{th} input to the i^{th} state with $1 \leq j \leq m_u$ and $1 \leq i \leq n_x$. Without loss of generality, $x = 0$ is assumed to be the equilibrium of the unforced system, i.e., $\dot{x}(t) = 0$ when $x = 0$, $u_a = 0$, and $w = 0$. The operator $|\cdot|$ is used to denote the absolute value of a scalar. The operator $\|\cdot\|$ is used to denote Euclidean norm of a vector.

Since the central focus of this work is on the difference between the requested actuation computed by the model-based controller and the actual actuation level applied to the process by the control actuators, we shall distinguish the two elements by calling the requested actuation $u_m(t)$ and the actual actuation $u_a(t)$.

5.2.2 Lyapunov-based MPC

The model-based controller that is used to determine the set-points for each actuator is a Lyapunov-based Model Predictive Controller (LMPC) [31]. One assumption about the design of the model-based control system used in this work is that it does not explicitly account for the dynamics of the control actuators and the presence of the process noise. Therefore, the model used for the design of the model-based control system assumes the following dynamics for the process:

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t)) + G(\tilde{x}(t))u_m(t) \quad (5.2)$$

where u_m is the commanded actuation by the high-level MPC. We make the following assumptions regarding the stability of the closed-loop system. We assume that there exists a Lyapunov-based controller $h(\tilde{x})$ such that the origin of the nominal closed-loop system under this controller, i.e., system of Eq. 5.2 with $u_m(t) = h(\tilde{x}) \forall t$, is asymptotically stable. Using converse Lyapunov theorems, this implies that there exist class \mathcal{K} functions* $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$ and a continuously differentiable Lyapunov function $V(\tilde{x})$ for the nominal closed-loop system that satisfy the following inequalities:

$$\alpha_1(\|\tilde{x}\|) \leq V(\tilde{x}) \leq \alpha_2(\|\tilde{x}\|), \quad \left\| \frac{\partial V(\tilde{x})}{\partial x} \right\| \leq \alpha_3(\|\tilde{x}\|) \quad (5.3a)$$

$$\frac{\partial V(\tilde{x})}{\partial x}(f(\tilde{x}) + G(\tilde{x})h(\tilde{x})) \leq -\alpha_4(\|\tilde{x}\|) \quad (5.3b)$$

for all $\tilde{x} \in D \subseteq \mathbb{R}^{n_x}$ where D is an open neighborhood of the origin. We denote the region $\Omega_\rho^\dagger \subseteq D$ as the stability region of the nominal closed-loop system, i.e., Eq. 5.2, under the control $u_m(t) = h(\tilde{x})$.

The existence of the controller $h(\tilde{x})$ allows us to formulate an MPC that inherits the stability

*A continuous function $\alpha : [0, a) \rightarrow [0, \infty)$ is said to belong to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$.

†We use Ω_ρ to denote the set $\Omega_\rho := \{x \in \mathbb{R}^{n_x} | V(x) \leq \rho\}$.

properties of $h(\tilde{x})$ [31], and it is described by the following optimization problem:

$$\min_{u_c \in S(\Delta)} \int_0^{N_c \Delta} [\hat{x}^T(\tau) Q \hat{x}(\tau) + u_c^T(\tau) R u_c(\tau)] d\tau \quad (5.4a)$$

$$\dot{\hat{x}}(\tau) = f(\hat{x}(\tau)) + G(\hat{x}(\tau)) u_c(\tau) \quad (5.4b)$$

$$\hat{x}(0) = x(t_k) \quad (5.4c)$$

$$u_c(\tau) \in U \quad (5.4d)$$

$$\frac{\partial V(x(t_k))}{\partial x} G(x(t_k)) u_c(0) \leq \frac{\partial V(x(t_k))}{\partial x} G(x(t_k)) h(x(t_k)) \quad (5.4e)$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ , Q and R are strictly positive definite symmetric weighting matrices, $x(t_k)$ is the process state measurement obtained at t_k , \hat{x} is the predicted trajectory of the system under the MPC, N_c is the number of steps in the prediction horizon, and V is the Lyapunov function corresponding to the controller $h(\tilde{x})$.

The optimal solution to this optimization problem is denoted by $u_c^*(\tau|t_k)$. The LMPC is implemented following a receding horizon strategy; at each sampling time t_k , a new state measurement $x(t_k)$ is received from the sensors and the optimization problem of Eq. 5.4 is solved, and $u_c^*(0|t_k)$ is sent to the actuators and it is implemented for $t \in [t_k, t_{k+1}]$.

The constraint of Eq. 5.4e guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time of the LMPC is less than or equal to the value obtained if only the Lyapunov-based control $u_m = h(\tilde{x})$ is implemented. This constraint allows the LMPC to inherit the stability properties of the Lyapunov-based control $h(\tilde{x})$ for sufficiently small sampling period Δ ; in particular, practical stability of the closed-loop system can be proven for sufficiently small Δ . For detailed results on Lyapunov-based MPC, please see [31].

Remark 5.1. *Note that in the design of the LMPC of Eq. 5.4 and its closed-loop stability analysis, one assumption is that the requested actuation $u_m(t)$ is applied directly to the process by the control actuators. In a practical setting, however, $u_m(t)$ has to go through the dynamics of the PID-controlled actuators before the system is actuated with $u_a(t)$. The central focus of this work is on how to bring $u_a(t)$ to be as close as possible to $u_m(t)$. The relationship between $u_a(t)$ and $u_m(t)$ will be discussed in detail in the next section.*

Remark 5.2. *Though a Lyapunov-based MPC is used as the model-based control system to demonstrate how the problem of low-level PID monitoring and retuning based on process state measurements can be approached, the monitoring and retuning methods presented here can be applied to any type of model-based control system (i.e., geometric control, distributed MPC [25], etc.). Specifically, as long as the requested actuation level $u_m(t)$ and the process state measurements are available to the monitoring and retuning system at all times, the same method presented in this work can be applied to detect the deviation of the actual actuation level $u_a(t)$ from the requested actuation level $u_m(t)$.*

5.2.3 Low-level PID Loops

As depicted in Fig. 5.1, $u_m(t)$ is sent from the model-based controller as the set-point to the control actuators. PID controllers are installed around these control actuators to help accelerate the actuator's response so that $u_a(t)$ can approach the value of $u_m(t)$ faster. Eq. 5.5 below shows the relationship between u_m and u_a in the Laplace domain:

$$u_a(s) = \frac{G_p G_c}{1 + G_p G_c} u_m(s) \quad (5.5)$$

where G_p is the actuator's transfer function and G_c is the PID controller's transfer function. G_c contains 3 parameters: K_c (proportional gain), τ_I (integral time constant), and τ_D (derivative time constant) and takes the following form:

$$G_c = K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s \right) \quad (5.6)$$

The transfer function of the actuator's dynamics, G_p , on the other hand, can be approximated as a first-order transfer function with dead time G'_p as follows:

$$G'_p = K_p \frac{e^{-\tau_d s}}{\tau_p s + 1} \quad (5.7)$$

where K_p is the actuator's gain, τ_d is the actuator dead time, and τ_p is the actuator's time constant.

The estimation of the actuator's transfer function (G'_p) will be needed by the FDI algorithm below when the actuator's expected behavior is calculated and also at the retuning step when a

new set of PID parameters is calculated. The expected actuation level (denoted by $u'_a(t)$) will be used as the benchmark upper limit of how well the control actuators can perform.

5.3 Monitoring and Retuning of Low-Level PID Loops

We consider the case where there is no access by the monitoring system to the measurements of the actual actuation levels $u_a(t)$ implemented by the control actuators on the process. Therefore, the detection of poor PID tunings must be performed based on the measurements of the states of the process. To this end, an FDI method is used as the main tool to extract actuator behavior from the process state measurements [6]. We use exponentially-weighted-moving-average (EWMA) residuals to detect and isolate poorly-tuned PID loops. Once a poorly-tuned actuator is isolated, a model-based tuning rule such as Cohen-Coon or internal model control is applied to the PID controller that regulates the poorly-tuned actuator.

The residuals are constructed from the difference between the expected behavior and the actual behavior of the plant. This is done by comparing the evolution of the actual system obtained from the state measurements against the evolution of the ideal filtered states based on the plant model. The actual closed-loop system state ($x(t)$) evolves in the following manner:

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + G(x(t))u_a(t) + w(t) \\ u_a(s) &= \frac{G_p G_c}{1 + G_p G_c} u_m(s)\end{aligned}\tag{5.8}$$

where $u_m(t)$ is the control action computed by the MPC and $u_a(t)$ is the actual actuation performed by the actuators. The filter state ($\check{x}(t)$), on the other hand, evolves as follows:

$$\begin{aligned}\dot{\check{x}}_i(t) &= f_i(\hat{x}_i(t)) + G_i(\hat{x}_i(t))u'_a(t) \\ \hat{x}_i &= [x_1 \cdots x_{i-1}, \check{x}_i, x_{i+1} \cdots x_{n_x}]^T \\ u'_a(s) &= \frac{G'_p G'_c}{1 + G'_p G'_c} u_m(s) \\ \check{x}(N\Delta_m) &= x(N\Delta_m), \forall N = 0, 1, 2, \dots\end{aligned}\tag{5.9}$$

where Δ_m is the MPC sampling time; G'_p is the estimated transfer function matrix of the control

actuators; G'_c is a well-tuned PID controller transfer function matrix based on the estimated model of the actuator G'_p . This makes $u'_a(t)$ the expected actuation level of $u_a(t)$.

Using Eqs. 5.8 and 5.9, the real-time measurements of $x(t)$ can be compared against the evolution of $\check{x}(t)$. The residual, or the difference between $x_i(t)$ and $\check{x}_i(t)$ denoted by $r_i(t)$, is expressed in the following manner:

$$r_i(t) = |\check{x}_i(t) - x_i(t)| \quad (5.10)$$

In the absence of noise and if $G'_p = G_p$, whenever the j^{th} element of u_a deviates from its expected behavior u'_{aj} , and the i^{th} -row- j^{th} -column element of the $G(x)$ matrix is non-zero, the i^{th} residual (r_i) would instantaneously become non-zero. In other words, r_i is non-zero only when there is a problem with the actuators that directly affect the i^{th} state of the system (relative degree of 1) [32, 34].

In practice however, model mismatch, process noise, and measurement noise are always present to some degree. Therefore, in a practical setting, the residuals will be non-zero regardless of the accuracy of the process model used in Eq. 5.9. Thus, before the model-based FDI method can be used in practice, the effects of process and measurement noise levels must first be recorded from fault-free closed-loop process operation data (with both the PID controllers and the MPC being well-tuned). On the basis of these noisy closed-loop system states, the mean and the standard deviation of the residuals are calculated and the thresholds are determined.

Occasional noise spikes can make the residuals exceed the thresholds for a brief period of time even when the actuators are functioning well; this can lead to the common problem of false alarms. To reduce the incidence of false alarms, we define a modified residual $r_{E,i}, i = 1, \dots, n_x$, for each residual r_i , calculated at discrete time instants t_k with $t_k = t_0 + k\Delta_r$, $k = 0, 1, 2, \dots$ and Δ_r being the interval between two consecutive state measurements. The weighted residual is calculated using an EWMA method as follows [6, 5]:

$$r_{E,i}(t_k) = \lambda r_i(t_k) + (1 - \lambda)r_{E,i}(t_{k-1}) \quad (5.11)$$

with $r_{E,i}(t_0) = r_i(t_0)$ and the weighting factor $\lambda \in (0, 1]$. The parameter λ determines the rate at which past data enters into the calculations of the weighted residual. When $\lambda = 1$, $r_{E,i}$ is equivalent

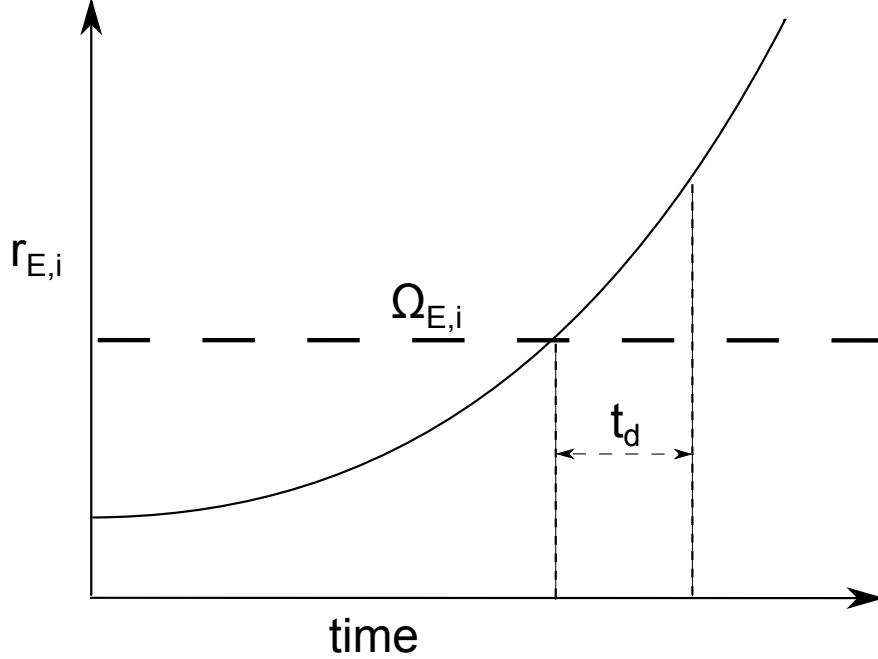


Figure 5.2: Monitoring scheme of PID response behavior based on the EWMA residuals of the process state. Poor tuning is declared after $r_{E,i}$ exceeds its threshold $\Omega_{E,i}$ continuously for $t = t_d$.

to r_i . The typical range of λ is between 0.2 and 0.5 depending on the desired level of sensitivity [5, 6]. Lower values of λ make the $r_E(t)$ curve smoother as potential noise spikes will have a smaller effect on the overall shape of the curve; i.e., instances of false alarm will be reduced. However, in the event where an actual poor tuning occurs, it may be detected and isolated more slowly.

The threshold, denoted by $\Omega_{E,i}$, for fault detection is defined as follows:

$$\Omega_{E,i} = \mu_i + \alpha \sigma_i \sqrt{\frac{\lambda}{2 - \lambda}} \quad (5.12)$$

where α is a threshold parameter determining how sensitive the FDI is; typical value of α is an integer value between 1 and 5. The parameters μ_i and σ_i are the mean and the standard deviation of the i^{th} residual during normal operation. Once $r_{E,i}$ exceeds the threshold ($\Omega_{E,i}$) for a fixed amount of time t_d (determined by the user), then poor tuning is declared in the actuator(s) directly affecting the i^{th} state and the retuning algorithm is activated. Figure 5.2 shows the schematic of how the EWMA residuals are used to activate the PID retuning algorithm at the end of waiting time t_d .

Once a poorly-tuned actuator is isolated, a PID tuning method can be applied to the PID controller based on the estimated transfer function of the actuator G'_p . To help ensure the stability

of the retuning algorithm, we employ a stability constraint. Specifically, whenever retuning is performed, the retuning algorithm makes sure that $\frac{G'_p G_c}{1+G'_p G_c}$ contains only strictly negative poles. In this work, we use Cohen-Coon and internal model control method to retune the PID parameters to demonstrate the approach. If desired, other model-based tuning rules may be used as well. Please see [46, 48] for other PID tuning methods.

Remark 5.3. *One feature that should be noted is that the PID retuning will be initiated if the magnitude of the residuals is above a certain threshold. This means that even if the difference between $u_{aj}(t)$ and $u'_{aj}(t)$ is appreciable but the difference between $\check{x}_i(t)$ and $x_i(t)$ is smaller than the threshold, the retuner will do nothing. This is a direct result of the fact that the real value of $u_a(t)$ is unknown and has to be estimated from the trajectories of the process states. A scenario like this can also happen when $G_{ij}(\cdot)$ is small.*

Remark 5.4. *The isolability structure of the system is also critical to the use of the monitoring algorithm proposed here. If the pattern of the residuals is not unique to an actuator (i.e., two actuators have the same signature because they directly affect the same system states) then a poorly-performing actuator cannot be isolated with high confidence, then all control actuators that may be poorly tuned should be retuned.*

Remark 5.5. *In the design of the filter of Eq. 5.9, a well-tuned PID controller, G'_c , is assumed to be known and is used to calculate the benchmark performance of the overall control system. In the case that G'_c is not known, the control actuation commanded by the MPC, u_m , can be used directly in the filter design (i.e., replace u'_a by u_m in the filter). Once a poorly-tuned actuator is isolated, retuning of the PID parameters should be carried out. Even in the case, G'_c is known, the retuning is recommended to account for changes in operation conditions as well as control actuator wear and tear over time.*

5.4 Application to a Nonlinear Chemical Process Network

In this section, we apply the PID monitoring and retuning methodology presented in the previous section to a three-vessel reactor-separator chemical process network example. More simulation examples can be found in [22].

The chemical process network example considered operates at an unstable steady-state whose detailed description and modeling as well as a schematic can be found in [6]. Sensor and process noise were added to the simulations.

We focus on the problem of monitoring and retuning of the PID controllers used to regulate the three heat input control actuators to each of the vessels: Q_1 , Q_2 , Q_3 , at the values computed by the MPC in each sampling time. In order to calculate the benchmark performance for each actuator ($u'_a(s)$) and a new set of PID parameters when PID retuning is needed, a first-order approximation of the transfer function of the actuator (G'_p) must be computed. In this example, all actuator dynamics are modeled with first-order transfer functions with time delay. All actuators have the same time constant (τ_p) of 2.82 seconds and time delay (τ_d) of 3.60 seconds, resulting in the following transfer function:

$$G_{actuator} = \frac{e^{-3.60s}}{2.82s + 1} \quad (5.13)$$

The control action computed by the MPC is sent to the control actuators every $\Delta_m = 0.01hr$. Thus, at every sampling time $t = N\Delta_m$, $N = 0, 1, 2, \dots$, the low-level PID controllers take the MPC command ($u_m(t)$) as the set-point and drive the actual actuation level ($u_a(t)$) to the set-point under the following closed-loop dynamics:

$$u_a(s) = \frac{G_p G_c}{1 + G_p G_c} u_m(s)$$

We choose the following parameters for PID monitoring and retuning. The EWMA parameter λ is set to 0.2. The EWMA residual threshold parameter α is chosen to be 5. The waiting time for fault isolation based on the EWMA residual is set to be $t_d = 0.01hr$.

For the actuators with the transfer function presented in Eq. 5.13, the PID parameters that give the best closed-loop response were found to be the following:

$$K_c^* = 0.648, \tau_I^* = 5.94 \text{ s}, \tau_D^* = 0.54s \quad (5.14)$$

These parameters were used to calculate G'_c . The poles of $\frac{G'_p G'_c}{1 + G'_p G'_c}$ calculated with the parameters above are found to be all negative. This, in conjunction with the approximate transfer function

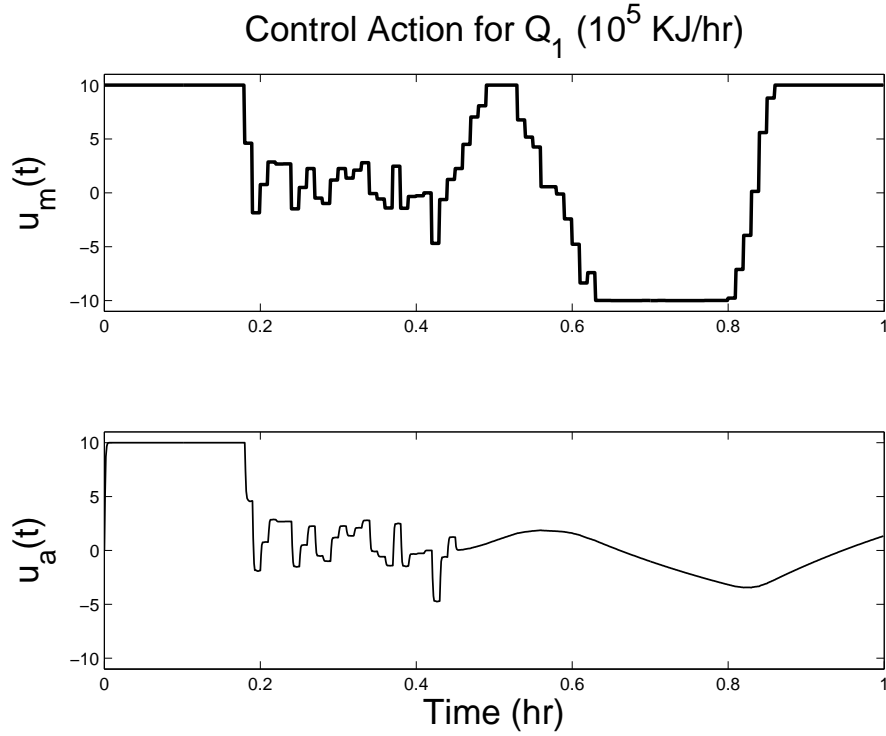


Figure 5.3: Requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is not implemented.

(G'_p) of the actuators of Eq. 5.13, was then used to approximate the ideal actuation performance ($u'_a(s)$) of each control actuator.

In the following example, we will illustrate how PID monitoring and retuning are applied to the system.

In this example, we start the process from the following initial condition: $x(0) = 0.8x_s$ where x_s is the operating steady-state. All the control actuators are properly tuned with the PID parameters shown in Eq. 5.14. At time $t = 0.45hr$, we apply poor tuning to the PID controller for the actuator Q_1 with the following parameters:

$$K_c = 0.00909, \tau_I = 11.9 \text{ s}, \tau_D = 0.655 \text{ s} \quad (5.15)$$

Figure 5.3 shows the comparison between the requested actuation level $u_m(t)$ and the actual actuation level $u_a(t)$ for Q_1 if the monitoring and retuning system is inactive. The EWMA residuals of the temperature in the 3 vessels are shown in Fig. 5.4.

With the monitoring system active, Fig. 5.5 shows the evolution of PID response $u_a(t)$ as it is

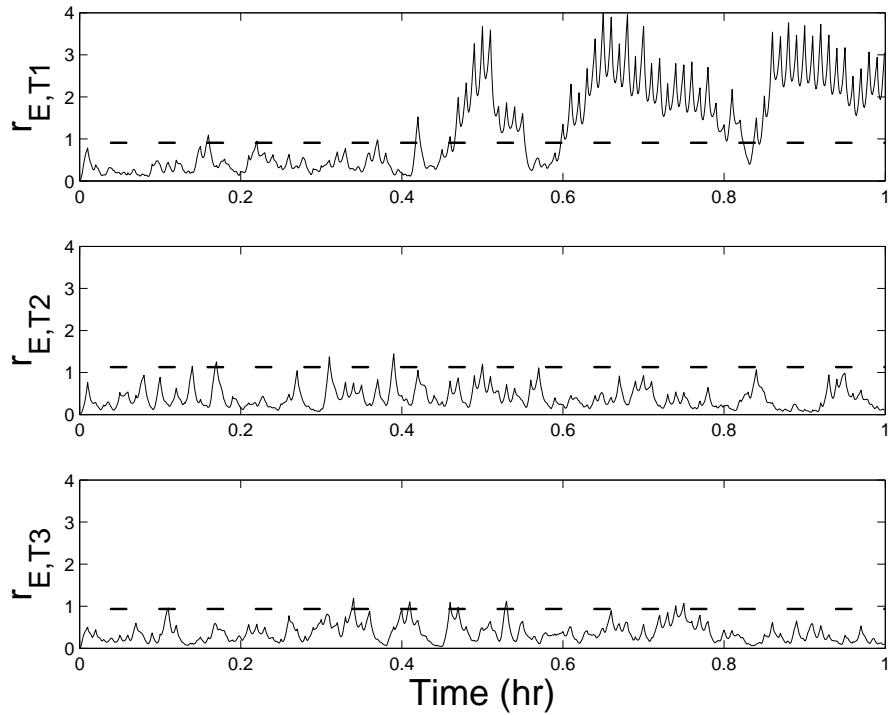


Figure 5.4: Temperature residuals for the 3 vessels computed via EWMA when PID retuning is not implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$.

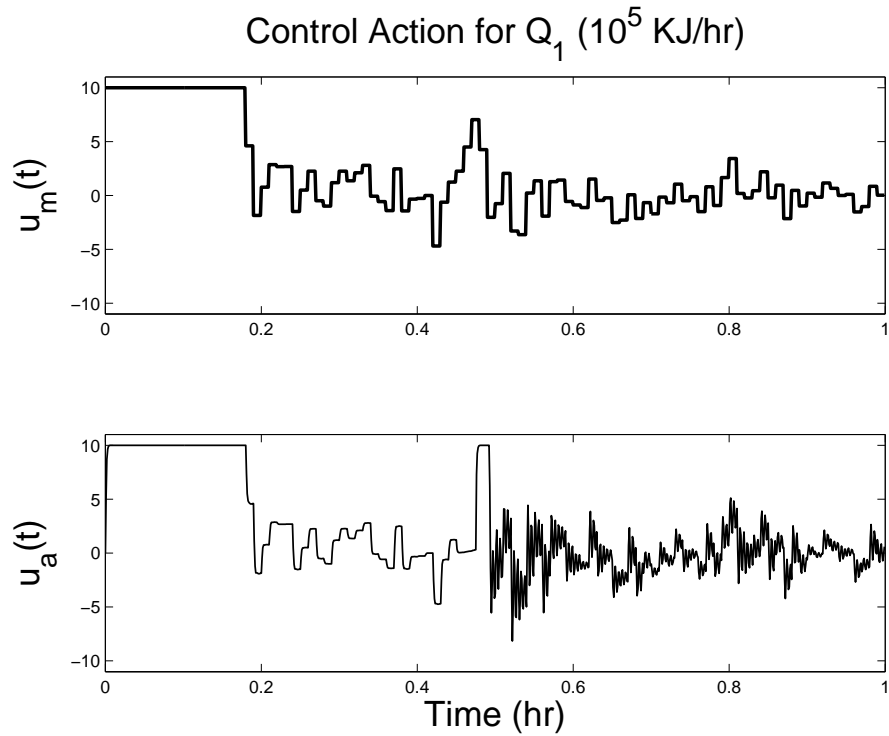


Figure 5.5: Requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is implemented.

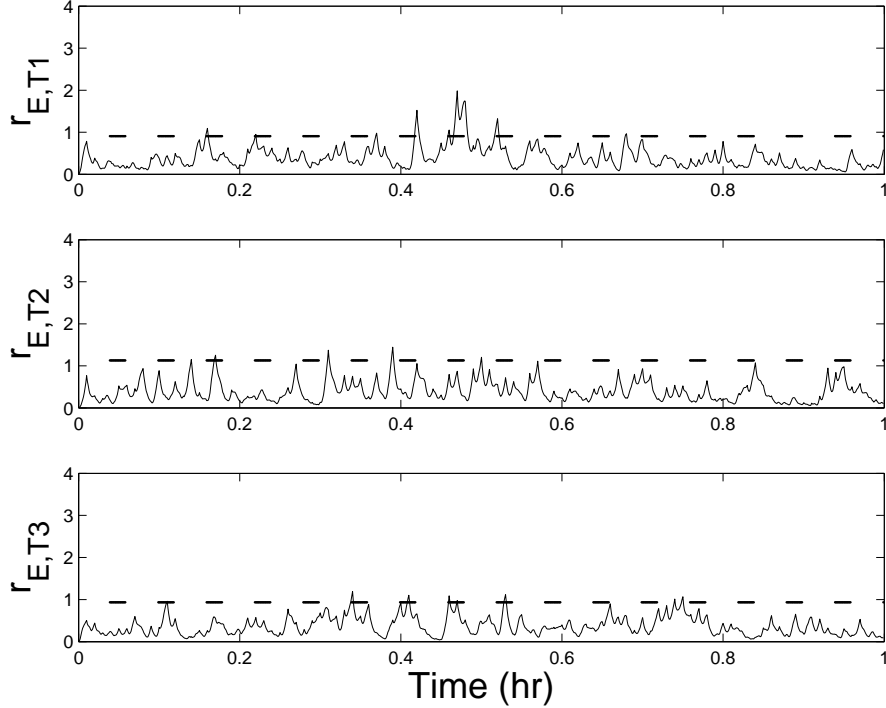


Figure 5.6: Temperature residuals for the 3 vessels computed via EWMA when PID retuning is implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$.

retuned at $t = 0.475hr$. As shown in Fig. 5.6, at $t = 0.465hr$, r_{E,T_1} exceeds its threshold Ω_{E,T_1} . At this point, the value of r_{E,T_1} starts being monitored closely for $t_d = 0.01hr$. By the time the system reaches $t = 0.475hr$, the value of r_{E,T_1} is found to have been above its threshold Ω_{E,T_1} for the entire duration from $t = 0.465hr$ to $t = 0.475hr$. Because the process state T_1 is the only state directly affected by the control actuator Q_1 , given the model-based FDI filter design, any anomaly detected in r_{E,T_1} is the result of a problem with the Q_1 control actuator. Therefore, the actuator Q_1 can be isolated with high confidence as the actuator with poor PID tuning. While other residuals (r_{E,T_2} and r_{E,T_3}) occasionally exceed their thresholds at various time instances during the operation, they do not exceed the thresholds for longer than $t_d = 0.01hr$. Thus, the monitoring system concludes that their values exceed their thresholds solely due to process and measurement noise.

Once the Q_1 control actuator is isolated as the poorly-tuned actuator, Cohen-Coon tuning method is applied to the controller around Q_1 based on the estimated transfer function of the control actuator G'_p . The Cohen-Coon tuning rule is based on the first-order-plus-dead-time estimation of the transfer function of the controlled process. Specifically, the Cohen-Coon tuning rule is as

follows [9]:

$$K_c = \frac{\tau_p}{K_p \tau_d} \left(\frac{4}{3} + \frac{\tau_d}{4\tau_p} \right), \quad \tau_I = \tau_d \frac{32 + 6 \frac{\tau_d}{\tau_p}}{13 + 8 \frac{\tau_d}{\tau_p}},$$

$$\tau_D = \tau_d \frac{4}{11 + 2 \frac{\tau_d}{\tau_p}}$$
(5.16)

where K_p is the actuator's gain, τ_d is the actuator dead time, and τ_p is the actuator's time constant. With this tuning rule and the estimated transfer function of the actuator G'_p presented in Eq.5.13, the resulting parameters for the PID of Q_1 are as follows:

$$K_c = 1.29, \quad \tau_I = 6.15 \text{ s}, \quad \tau_D = 1.06 \text{ s}$$
(5.17)

After Q_1 is retuned, no more problem can be detected from the EWMA residuals of T_1 . In terms of the actual control actuator performance, after being retuned with Cohen-Coon method, $u_a(t)$ tracks $u_m(t)$ quite well; please see Fig. 5.5.

5.5 Conclusion

In this chapter, we focused on the problem of monitoring and retuning of low-level PID control loops used to regulate control actuators to the values computed by advanced model-based control systems like MPC. Focusing on the case where the real-time measurement of the actuation level is unavailable, we use process state measurements and process models to carry out PID controller monitoring and compute appropriate residuals. Once a poorly-tuned PID controller is detected and isolated, a PID tuning method based on the estimated transfer function of the control actuator was applied to retune PID controller. The proposed method was applied to a nonlinear reactor-separator process operating under MPC control with low-level PID controllers regulating the control actuators and its performance was successfully evaluated via simulations.

Chapter 6

Conclusions

The dissertation develops an integrated approach to monitoring and isolating actuator faults in a nonlinear process system operating under a DMPC system. By taking advantage of both process models and process measurements an FDIFTC system based on the system structure and structure of potential faults can successfully and optimally recover from targeted actuator faults. In brief, successful fault tolerant control requires that the actuator fault may be observed (i.e., the appropriate states are monitored), secondly that the faults have an isolable structure, third that the control system is structured as to maintain proper control through either identical redundant actuators or a reconfigurable control scheme, and lastly that the FDIFTC system can operate sufficiently fast to maintain stable operation.

In Chapter 2, we developed the initial approach for an FDIFTC system that does not rely on an identical redundant actuator to maintain closed-loop stability. In that work, a more realistic approach to fault tolerant control is imposed by requiring that only the remaining control action be used to maintain closed-loop stability after isolation and resetting the actuator fault. Specifically for detection and isolation, we first design fault detection filters and corresponding filter residuals to effectively detect actuator faults. Considering that real measurement and process noise is present in the plant, this requires a data-based approach to compliment the model-based filters to properly detect a fault and again to confidently isolate a fault. Since successful FTC requires that a certain minimum amount of control actions remain that can stabilize the system, we specifically address these various situations.

In Chapter 3, we expand on the approach developed in Chapter 2 by addressing a more realistic problem of not being able to reset a faulty actuator. By having a persistent fault present, there is the possibility that the original operating point is no longer a viable option. In previous work by resetting a fault, its effects were sufficiently nullified upon initiation of fault tolerant control that a controllable structure insured successful reconfiguration. With the fault now being persistent, the plant continues to deviate away from the target operating point and possibly outside the control system's stability region, such that quick detection and isolation are critical for successful FTC. Then, we proposed a fault isolation approach which uses adaptive fault isolation time windows to quickly and accurately isolate actuator faults and reduce the probability of false alarms. Subsequently, we designed appropriate FTC strategies to handle the actuator faults by reconfiguring the DMPC system and maintaining the closed-loop system state within a desired operating region. The applicability and effectiveness of the proposed approach were illustrated via extensive simulations based on a nonlinear reactor-separator process example.

In Chapter 4, the methods developed in previous chapters were applied to the Catalytic Alkylation of Benzene Process to demonstrate the FDIFTC system's effectiveness in a real-world setting. Similarly Chapter 5, introduces a low-level PID layer to more realistically reflect industry practices of interfacing MPC with PIDs directly controlling actuators with the goal of monitoring and retuning a poorly tuned low-level PID control loops using methods from Chapters 2 and 3.

Bibliography

- [1] *Ethylbenzene, Vol. I*, volume Ch. Petrochemicals and Petrochemical Processing. Taylor & Francis Group, 2006.
- [2] K. Astrom, T. Hagglund, C. Hang, and W. Ho. Automatic tuning and adaptation for PID controllers - a survey. *Control Engineering Practice*, 1:699–714, 1993.
- [3] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Cont. Sys. Mag.*, 22:44–52, 2002.
- [4] S. Chand. Self-monitoring tuner for feedback controller. *U.S. Patent*, 5,159,547, 1992.
- [5] D. Chilin, J. Liu, J. F. Davis, and P. D. Christofides. Data-based monitoring and reconfiguration of a distributed model predictive control system. *International Journal of Robust and Nonlinear Control*, 22:68–88, 2012.
- [6] D. Chilin, J. Liu, D. Muñoz de la Peña, P. D. Christofides, and J. F. Davis. Detection, isolation and handling of actuator faults in distributed model predictive control systems. *Journal of Process Control*, 20:1059–1075, 2010.
- [7] P. D. Christofides, J. F. Davis, N. H. El-Farra, D. Clark, K. R. D. Harris, and J. N. Gipson. Smart plant operations: Vision, progress and challenges. *AIChE Journal*, 53:2734–2741, 2007.
- [8] P. D. Christofides and N. H. El-Farra. *Control of nonlinear and hybrid process systems: Designs for uncertainty, constraints and time-delays*. Springer-Verlag, Berlin, German, 2005.
- [9] G. H. Cohen and G. A. Coon. Theoretical consideration of retarded control. *ASME*, 75:827–834, 1953.

- [10] C. DePersis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Trans. Automat. Contr.*, 46:853–865, 2001.
- [11] C. DePersis and A. Isidori. On the design of fault detection filters with game-theoretic-optimal sensitivity. *Int. J. Rob. & Non. Contr.*, 12:729–747, 2002.
- [12] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Automat. Contr.*, 52:1249–1263, 2007.
- [13] P.-G. Eriksson and A. J. Isaksson. Some aspects of control loop performance monitoring. *Control Applications*, 2:1029–1034, 1994.
- [14] 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.
- [15] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *J. Proc. Contr.*, 7:403–424, 1997.
- [16] H. Ganji, J.S. Ahari, A. Farshi, and M. Kakavand. Modelling and simulation of benzene alkylation process reactors for production of ethylbenzene. *Petroleum and Coal*, 46:55–63, 2004.
- [17] T. J. Harris. Assessment of control loop performance. *The Canadian Journal of Chemical Engineering*, 67:856–861, 1989.
- [18] D. Jia and B. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the American Control Conference*, pages 4507–4512, Anchorage, 2002.
- [19] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42:2105–2115, 2006.
- [20] H. K. Khalil. *Nonlinear Systems*. Macmillan Publishing Company, New York, 1992.
- [21] W. J. Lee. *Ethylbenzene dehydrogenation into styrene: kinetic modeling and reactor simulation*, *Ph.D. thesis*. PhD thesis, Texas A&M University, College Station, TX, USA, 2005.

- [22] A. Leosirikul, D. Chilin, J. Liu, J. F. Davis, and P. D. Christofides. Monitoring and retuning of low-level PID control loops. *Chemical Engineering Science*, 69:287–295, 2012.
- [23] Y. Lin, E. D. Sontag, and Y. Wang. A smooth converse lyapunov theorem for robust stability. *SIAM J. Control and Optimization*, 34:124–160, 1996.
- [24] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE Journal*, 56:2137–2149, 2010.
- [25] J. Liu, D. Muñoz de la Peña, and P. D. Christofides. Distributed model predictive control of nonlinear process systems. *AIChE J.*, 55:1171–1184, 2009.
- [26] J. Liu, D. Muñoz de la Peña, and P. D. Christofides. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica*, 46:52–61, 2010.
- [27] J. M. Lucas and M. S. Saccucci. Exponentially weighted moving average control schemes: Properties and enhancements. *Technometrics*, 32:1–12, 1990.
- [28] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. A distributed MPC scheme with low communication requirements. In *Proceedings of 2009 American Control Conference*, pages 2797–2802, Saint Louis, MO, USA, 2009.
- [29] L. Magni and R. Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42:1231–1236, 2006.
- [30] N. Mehranbod, M. Soroush, and C. Panjapornpon. A method of sensor fault detection and identification. *Journal of Process Control*, 15:321–339, 2005.
- [31] 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.
- [32] P. Mhaskar, A. Gani, N. H. El-Farra, P. D. Christofides, and J. F. Davis. Integrated fault-detection and fault-tolerant control of process systems. *AIChE Journal*, 52:2129–2148, 2006.

- [33] P. Mhaskar, A. Gani, C. McFall, P. D. Christofides, and J. F. Davis. Fault-tolerant control of nonlinear process systems subject to sensor data losses. *AIChE Journal*, 53:654–668, 2007.
- [34] P. Mhaskar, C. McFall, A. Gani, P. D. Christofides, and J. F. Davis. Isolation and handling of actuator faults in nonlinear systems. *Automatica*, 44:53–62, 2008.
- [35] D. Muñoz de la Peña and P. D. Christofides. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Trans. Automat. Cont.*, 53:2076–2089, 2008.
- [36] B. Ohran, D. Muñoz de la Peña, P. D. Christofides, and J. F. Davis. Enhancing data-based fault isolation through nonlinear control. *AIChE Journal*, 53:2731–2741, 2008.
- [37] 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.
- [38] C. Perego and P. Ingallina. Combining alkylation and transalkylation for alkylaromatic production. *Green Chemistry*, 6:274–279, 2004.
- [39] S. J. Qin. Control performance monitoring - a review and assessment. *Computers & Chemical Engineering*, 23:173–186, 1998.
- [40] D. M. Raimondo, L. Magni, and R. Scattolini. Decentralized MPC of nonlinear system: An input-to-state stability approach. *Int. J. Robust Nonlinear Control*, 17:1651–1667, 2007.
- [41] J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *J. Proc. Contr.*, 18:839–845, 2008.
- [42] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80:1517–1531, 2007.
- [43] W. J. Rugh. Analytical framework for gain scheduling. *IEEE Control Systems Magazine*, 11:79–84, 1991.
- [44] R. Scattolini. Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, 19:723–731, 2009.
- [45] D. Shi and F. Tsung. Modeling and diagnosis of feedback-controlled process using dynamic PCA and neural networks. *International Journal of Production Research*, 41:365–379, 2003.

- [46] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13:291–309, 2003.
- [47] E. Sontag. A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization. *Systems and Control Letters*, 13:117–123, 1989.
- [48] S.-W. Sung, I.-B. Lee, and B.-K. Lee. On-line process identification and automatic tuning method for PID controllers. *Chemical Engineering Science*, 53:1847–1859, 1998.
- [49] F. Teng, A. Lotfi, and A. Tsoi. Novel fuzzy logic controllers with self-tuning capability. *Journal of Computers*, 3:9–16, 2008.
- [50] F. Tsung. Statistical monitoring and diagnosis of automatic control processes using dynamic PCA. *International Journal of Production Research*, 38:625–637, 2000.
- [51] F. Tsung and J. Shi. Integrated design of run-to-run PID controller and SPC monitoring for process disturbance rejection. *IIE Transactions*, 31:517–527, 1999.
- [52] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference ECC 2005*, pages 6680–6685, Seville, Spain, 2005.
- [53] A. Wächter and L. T. Biegler. On the implementation of primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.
- [54] Q. Wang, Z. Zhang, L. Chek, and K. Astrom. Guaranteed dominant pole placement with PID controllers. *Journal of Process Control*, 19:349–352, 2009.
- [55] E. B. Ydstie. New vistas for process control: Integrating physics and communication networks. *AIChE Journal*, 48:422–426, 2002.
- [56] H. You, W. Long, and Y. Pan. The mechanism and kinetics for the alkylation of benzene with ethylene. *Petroleum Science and Technology*, 24:1079–1088, 2006.
- [57] S. H. Zad and M. Massoumnia. Generic solvability of the failure detection and identification problem. *Automatica*, 35:887–893, 1999.

- [58] Z.-Y. Zhao, M. Tomizuka, and S. Isaka. Fuzzy gain scheduling of PID controllers. *IEEE Transactions on Systems, Man, and Cybernetics*, 23:1392–1398, 1993.
- [59] J. G. Ziegler and N. Nichols. Optimum settings for automatic controllers. *ASME*, 64:759–768, 1942.