Journal of Process Control 20 (2010) 1059–1075

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Detection, isolation and handling of actuator faults in distributed model predictive control systems

David Chilin[a], Jinfeng Liu[a], David Muñoz de la Peña[b], Panagiotis D. Christofides[a,c,*], James F. Davis[a]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, United States
[b] Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos S/N, 41092 Sevilla, Spain
[c] Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1592, United States

ARTICLE INFO

Article history:
Received 30 October 2009
Received in revised form 5 February 2010
Accepted 3 May 2010

Keywords:
Fault detection and isolation
Fault-tolerant control
Distributed model predictive control
Nonlinear processes

ABSTRACT

In this work, we focus on monitoring and reconfiguration of distributed model predictive control systems applied to general nonlinear processes in the presence of control actuator faults. Specifically, we consider nonlinear process systems controlled with a distributed control scheme in which two Lyapunov-based model predictive controllers manipulate two 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, a model-based fault detection and isolation and fault-tolerant control system which detects and isolates actuator faults and determines how to reconfigure the distributed control system to handle the actuator faults while maintaining closed-loop stability is designed. A detailed mathematical analysis is carried out to determine precise conditions for the stabilizability of the fault detection and isolation and fault-tolerant control system. A chemical process example, consisting of two continuous stirred tank reactors and a flash tank separator with a recycle stream and involving stabilization of an unstable steady-state, is used to demonstrate the approach.

1. Introduction

Optimal operation and management of abnormal situations are major challenges in the process industries as they account for at least $10 billion in lost annual revenue within the US alone. This issue has motivated significant research efforts in the areas of process control and operations. Traditionally, control systems rely on centralized control architectures utilizing dedicated, wired links to measurement sensors and control actuators to regulate appropriate process variables at desired values. While this paradigm to process control has been successful, it is limited in the number of the process state variables, manipulated inputs and measurements in a chemical plant because the computational time needed for the solution of a centralized control problem may increase significantly and may impede the ability of centralized control systems (particularly when nonlinear constrained optimization-based control systems like model predictive control (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 [1,2] 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 action 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 [3–5] for reviews of results in this area. More specifically, in [6], the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints was considered. In [7,8], the effect of the coupling was modeled as a bounded disturbance compensated using a robust MPC formulation. In [9], it was proven that through multiple communications

[*] Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, 5531 Boelter Hall, Los Angeles, CA 90095-1592, United States. Tel.: +1 310 794 1015; fax: +1 310 206 4107.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

0959-1524/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2010.05.001

between distributed controllers and using system-wide control objective functions, stability of the closed-loop system can be guaranteed. In [10], DMPC of decoupled systems (a class of systems of relevance in the context of multi-agents systems) was studied. In [11], an MPC algorithm was proposed for the case where the system is nonlinear, discrete-time and no information is exchanged between local controllers, and in [12], MPC for nonlinear systems was studied from an input-to-state stability point of view. In [13], a game theory based DMPC scheme for constrained linear systems was proposed.

In our previous work [14], we proposed a DMPC architecture with one-directional communication for nonlinear process systems. 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 [15], we also considered the design of DMPC architectures for systems with asynchronous and delayed measurements. In a recent work [16], we extended the DMPC architecture developed in [14] to include multiple distributed controllers and relaxed the requirement that one of the distributed controllers should be able to stabilize the closed-loop system. In the new proposed DMPC architecture in [16], 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 (FTC). Specifically, the problem of using fundamental process models for the purpose of detecting faults has been studied extensively in the context of linear systems [17–20]; and also, some existential results in the context of nonlinear systems have been derived [21,22]. 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 (FDI). FTC has been an active area of research primarily within the context of aerospace control engineering (see, e.g., [23,24]). Over the last 3 years, we have initiated an effort on FTC of nonlinear processes trying to bring together the disconnected fields of process fault-diagnosis and nonlinear process control. We have looked at both actuator [25] and sensor [26] faults and their impact and handling in the context of chemical process control. Other important recent work on the subject of fault diagnosis and handling includes [27–30] where the emphasis has been on plants described by distributed parameter systems. Despite this progress, there are no results on monitoring and reconfiguration of cooperative, distributed control systems.

The focus of this work 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. Problem formulation and preliminaries

2.1. Class of nonlinear systems

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

$$\dot{x} = f(x) + g_1(x)(u_1 + \tilde{u}_1) + g_2(x)(u_2 + \tilde{u}_2) \tag{1}$$

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:

$$U_1 = \{u_1 + \tilde{u}_1 \in R^{m_1} : |u_1 + \tilde{u}_1| \le u_1^{\max}\}$$
$$U_2 = \{u_2 + \tilde{u}_2 \in R^{m_2} : |u_2 + \tilde{u}_2| \le u_2^{\max}\}.$$

We consider possible (independent) faults, $\tilde{u}_{f,j} \in R, j = 1, \ldots, m_1 + m_2$, in each of the manipulated inputs. 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 $j = 1, \ldots, m_1 + m_2$. 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 $\{t_{k\ge 0}\}$ with $t_k = t_0 + k\Delta,\ k = 0, 1, \ldots$ where t_0 is the initial time and Δ is the sampling time.

Remark 1. The variable $\tilde{u}_{f,j}$ associated with the jth 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 $u_{c,j}$; or it can be a function of the form $\tilde{u}_{f,j} = -u_{c,j} + c$ 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. 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 [31], this assumption implies that there exist functions $\alpha_i(\cdot),\ i = 1, 2, 3, 4$ of class $\mathcal{K}$ [1] and a continuous differentiable Lyapunov function $V(x)$ for the nominal closed-loop system

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

that satisfy the following inequalities:

$$\begin{aligned}
&\alpha_1(|x|) \le V(x) \le \alpha_2(|x|)\\
&\frac{\partial V(x)}{\partial x}(f(x)+g_1(x)h(x)) \le -\alpha_3(|x|)\\
&\left|\frac{\partial V(x)}{\partial x}\right| \le \alpha_4(|x|)\\
&h(x) \in U_1
\end{aligned} \tag{2}$$

for all $x \in D \subseteq R^n$ where D is an open neighborhood of the origin. We denote the region $\Omega_\rho \subseteq D$[2] 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 asymptotic stability of $x = 0$ under $h(x)$, (b) the construction of $V(x)$ can be readily done using a variety of methods (see [31,32] 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 that

$$|f(x)+g_1(x)(u_1+\tilde{u}_1)+g_2(x)(u_2+\tilde{u}_2)| \le M_1 \tag{3}$$

$$\begin{aligned}
&\left|\frac{\partial V}{\partial x}(f(x)+g_1(x)u_1+g_2(x)u_2) - \frac{\partial V}{\partial x}(f(x')+g_1(x')u_1+g_2(x')u_2)\right|\\
&\quad \le L_{x,1}|x-x'|
\end{aligned} \tag{4}$$

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

2.3. DMPC design for fault-free system

Following [14], 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. (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:

$$\min_{u_2 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + u_1(\tau)^T R_{c1} u_1(\tau) + u_2(\tau)^T R_{c2} u_2(\tau)]\, d\tau \tag{5a}$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + g_1(\tilde{x}(\tau))u_1(\tau) + g_2(\tilde{x}(\tau))u_2(\tau) \tag{5b}$$

$$u_1(\tau) = h(\tilde{x}(j\Delta)), \quad \forall \tau \in [j\Delta, (j+1)\Delta),\ j = 0, \ldots, N-1 \tag{5c}$$

$$\tilde{x}(0) = x(t_k) \tag{5d}$$

$$u_2(\tau) \in U_2 \tag{5e}$$

$$\frac{\partial V(x)}{\partial x} g_2(x(t_k))u_2(0) \le 0. \tag{5f}$$

In the optimization problem of Eq. (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. (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. (5e) defines the constraint on the manipulated input u_2 and the stability constraint of Eq. (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:

$$\min_{u_1 \in S(\Delta)} \int_0^{N\Delta} [\tilde{x}(\tau)^T Q_c \tilde{x}(\tau) + u_1(\tau)^T R_{c1} u_1(\tau) + u_2^*(\tau|t_k)^T R_{c2} u_2^*(\tau|t_k)]\, d\tau \tag{6a}$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + g_1(\tilde{x}(\tau))u_1(\tau) + g_2(\tilde{x}(\tau))u_2^*(\tau|t_k) \tag{6b}$$

$$\tilde{x}(0) = x(t_k) \tag{6c}$$

$$u_1(\tau) \in U_1 \tag{6d}$$

$$\frac{\partial V(x)}{\partial x} g_1(x(t_k))u_1(0) \le \frac{\partial V(x)}{\partial x} g_1(x(t_k))h(x(t_k)). \tag{6e}$$

In the optimization problem of Eq. (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. (6). The optimal solution to this optimization problem is denoted by $u_1^*(\tau|t_k)$. The constraint of Eq. (6d) defines the constraint on the manipulated input u_1 and the stability constraint of Eq. (6e) is also required to guarantee the closed-loop stability.

Once both optimization problems are solved (see [14] for results on the feasibility and stability of the LMPCs of Eqs. (5) and (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. (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 practical stability, as the Lyapunov-based control law h implemented in a sample-and-hold fashion [14].

[2] We use Ω_ρ to denote the set $\Omega_\rho := \{x \in R^n : V(x) \le \rho\}$.

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 constrains 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. (1) with $u_{11}=h_{21}(x)$, $u_2=h_{22}(x)$ and $u_{12}=0$ that satisfy the following inequalities:

$$\begin{aligned}
&\alpha_1'(|x|)\le V_2(x)\le\alpha_2'(|x|)\\
&\frac{\partial V_2(x)}{\partial x}(f(x)+g_1(x)[h_{21}(x)^T\ 0^T]^T+g_2(x)h_{22}(x))\le-\alpha_3'(|x|)\\
&\left|\frac{\partial V_2(x)}{\partial x}\right|\le\alpha_4'(|x|)\\
&h_{21}(x)\in U_1,\ h_{22}(x)\in U_2
\end{aligned}\tag{7}$$

for all $x\in D_2\subseteq R^n$ where D_2 is an open neighborhood of the origin. We denote $\Omega_{2,\gamma}\subseteq D_2$ [3] 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

$$|f(x)+g_1(x)(u_1+\tilde{u}_1)+g_2(x)(u_2+\tilde{u}_2)|\le M_2\tag{8}$$

$$\left|\frac{\partial V_2}{\partial x}(f(x)+g_1(x)u_1+g_2(x)u_2)-\frac{\partial V_2}{\partial x}(f(x')+g_1(x')u_1+g_2(x')u_2)\right|\le L_{x,2}|x-x'|\tag{9}$$

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 [16]. 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 2.3.

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

$$\min_{u_2\in S(\Delta)}\int_0^{N\Delta}[\tilde{x}(\tau)^TQ_c\tilde{x}(\tau)+u_1(\tau)^TR_{c1}u_1(\tau)+u_2(\tau)^TR_{c2}u_2(\tau)]\,d\tau\tag{10a}$$

$$\dot{\tilde{x}}(\tau)=f(\tilde{x}(\tau))+g_1(\tilde{x}(\tau))[u_{11}(\tau)^Tu_{12}(\tau)^T]^T+g_2(\tilde{x}(\tau))u_2(\tau)\tag{10b}$$

$$u_{11}(\tau)=h_{21}(\tilde{x}(j\Delta)),\quad\forall\,\tau\in[j\Delta,(j+1)\Delta),\quad j=0,\ldots,N-1\tag{10c}$$

$$u_{12}(\tau)=0\tag{10d}$$

$$\tilde{x}(0)=x(t_k)\tag{10e}$$

$$u_2(\tau)\in U_2\tag{10f}$$

$$\frac{\partial V_2(x)}{\partial x}g_2(x(t_k))u_2(0)\le\frac{\partial V_2(x)}{\partial x}g_2(x(t_k))h_{22}(x(t_k)).\tag{10g}$$

The solution to the optimization problem of Eq. (10) is denoted by $u_2^{b,*}(\tau|t_k)$. The backup LMPC 1 optimizes u_{11} and is designed as follows:

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

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

$$\tilde{x}(0)=x(t_k)\tag{11c}$$

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

$$\frac{\partial V_2(x)}{\partial x}g_1(x(t_k))[u_{11}(0)^T\ 0^T]^T\le\frac{\partial V_2(x)}{\partial x}g_1(x(t_k))[h_{21}(x(t_k))^T\ 0^T]^T.\tag{11e}$$

The solution to the optimization problem of Eq. (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:

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

The fault-free closed-loop system of Eq. (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 achieves practical stability of the origin [16].

Remark 2. Note that in the DMPC design of Eqs. (5)–(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. (10)–(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 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 3.2 for further discussion on this issue.

Remark 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 3 for the proposed fault-tolerant control methods and [33,25] for more discussion on the relationship between system structure and FTC schemes.

Remark 5. Because the manipulated inputs enter the dynamics of the system in an affine manner, the constraints of Eqs. (10g) and (11e) in the backup DMPC design are decoupled for the distributed

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

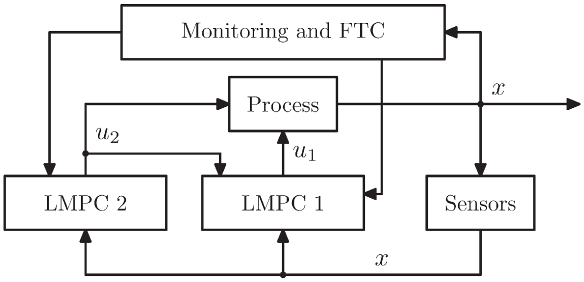

Fig. 1. Proposed FDI and FTC structure for DMPC.

controllers. Because of this, the stability of the closed-loop system is ensured even when the two LMPCs are evaluated in a completely decentralized fashion. However, the communication between the two LMPCs (i.e., the backup LMPC 2 sends $u_2^{b,*}(\tau|t_k)$ to the backup LMPC 1) as well as the use of h_{22} to estimate the control actions of LMPC 1 (i.e., the backup LMPC 2 uses h_{22} to approximate the control actions of the backup LMPC 1) allows us to improve the closed-loop performance. Please see [16] for more discussions on the stability and optimality properties of the backup DMPC design.

3. FDI and FTC strategies

In this section, we look at the closed-loop system under the DMPC control of Eqs. (5)–(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. 1.

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 [33,34].

The DMPC system of Eqs. (5)–(6) is the control configuration for the fault-free system of Eq. (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 pth, $p = 1, \ldots, n$, state in the system state vector x is as follows [33]:

$$\dot{\hat{x}}_p(t) = f_p(X_p) + g_{1p}(X_p)u_1^L(X_p) + g_{2p}(X_p)u_2^L(X_p) \quad (13)$$

where $\hat{x}_p$ is the filter output for the pth state, f_p, g_{1p} and g_{2p} are the pth 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:

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

Note that in the filter of Eq. (13), the control inputs $u_1^L(X_p)$ and $u_2^L(X_p)$ are determined by the same LMPC 1 of Eq. (6) and the LMPC 2 of Eq. (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 [33]:

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

with $p = 1, \ldots, n$. 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. This property of the filters is summarized in Theorem 1.

Theorem 1 (cf. [33]). *Consider the system of* Eq. (1) *in closed-loop under the DMPC design of* Eqs. (5)–(6). *Let the FDI filter for the pth state be designed as in Eq.* (13) *and* $\hat{x}_p(0) = x_p(0)$. *Assume that the state* x_p *is only directly affected by the jth element,* $u_{c,j}$, *of the input vector* $[u_1^T\ u_2^T]^T$. *Let* t_p^f *be the earliest time for which the fault associated with* $u_{c,j}$ *is not zero* (i.e., $\tilde{u}_{f,j} \neq 0$), *then the FDI filter of* Eq. (13) *ensures that* $r_p(t_p^{f+}) \neq 0$. *Also,* $r_p(t) \neq 0$ *only if* $\tilde{u}_{f,j}(s) \neq 0$ *for some* $0 \leq s < t$.

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. Note also that although the control inputs are involved in the formulation of filters, the design of the filters is independent from the design of the control system; therefore, the way we split the control inputs in the DMPC design does not affect the FDI system.

The objective of the FDI scheme it 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., $\tilde{u}_{f,j}$, $j = 1, \ldots, m_1 + m_2$) 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. 1.

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 2.3, 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. (5)–(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 2 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 2.4, 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. (10)–(11) to manipulate u_{11} and u_2 to control the process. Theorem 3 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. (1) within the DMPC system of Eqs. (5)–(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:

$$u_1(t) = u_1^L(x), \quad \forall t$$
$$u_2(t) = \begin{cases} u_2^L(x), & t \le t_f \\ 0, & t > t_f \end{cases} \tag{14}$$

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

$$u_1(t) = \begin{cases} u_1^L(x), & t \le t_f \\ \begin{bmatrix} u_{11}^b(x) \\ 0 \end{bmatrix}, & t > t_f \end{cases} \tag{15a}$$

$$u_2(t) = \begin{cases} u_2^L(x), & t \le t_f \\ u_2^b(x), & t > t_f \end{cases} \tag{15b}$$

In what follows, we summarize the properties of the switching rules of Eqs. (14) and (15) in Theorems 2 and 3. In order to state and prove the two theorems, we first introduce the following proposition.

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

$$-\alpha_3(\alpha_2^{-1}(\rho_s)) + \alpha_4(\alpha_1^{-1}(\rho))L_{x,1}M_1\Delta \le \frac{-\epsilon_s}{\Delta}. \tag{16}$$

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

$$\rho_{min} = \max\{\rho_s, \max\{V(\hat{x}(t+\Delta)) : V(\hat{x}(t)) \le \rho_s\}\} \tag{17}$$

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

$$V(\hat{x}(k\Delta)) \le \max\{V(\hat{x}(0)) - k\epsilon_s, \rho_{\min}\}. \tag{18}$$

Proposition 1 ensures that if the fault-free system of Eq. (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. (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 2. *Consider the system of* Eq. (1) *in closed-loop under the DMPC scheme of* Eqs. (5)–(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. (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. (13), this fault can be detected and isolated immediately after t_f. According to the switching rule of Eq. (14), from t_0 to t_f, the closed-loop system of Eq. (1) is controlled under the DMPC scheme of Eqs. (5)–(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. (5)–(6), if $x(t_0) \in \Omega_\rho$, the state of the closed-loop system of Eq. (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. (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. (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 2. □

Theorem 3. *Consider the system of* Eq. (1) *in closed-loop under the DMPC scheme of* Eqs. (5)–(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. (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. (13), this fault can be detected and isolated immediately after t_f. According to the switching rule of Eq. (15), from t_0 to t_f, the closed-loop system of Eq. (1) is controlled under the DMPC scheme of Eqs. (5)–(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. (5)–(6), if $x(t_0) \in \Omega_\rho$, the state of the closed-loop system of Eq. (1) will be maintained in Ω_ρ.

According to the switching rule of Eq. (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. (10)–(11). This proves Theorem 3. □

Remark 6. 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 7. In Theorems 2 and 3, 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 introduces 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 that the state of the closed-loop system is still within the stability regions of the backup control systems at the time of fault isolation. This point is demonstrated in the application of the proposed methods to a chemical process in Section 4.

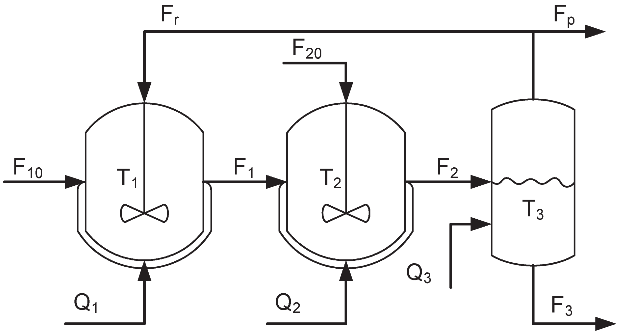

Fig. 2. Two CSTRs and a flash tank with recycle stream.

4. Application to a reactor–separator process

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. 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 $A \to B$ and $A \to C$ (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 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:

$$\begin{aligned}\frac{dT_1}{dt} &= \frac{F_{10}}{V_1}(T_{10}-T_1)+\frac{F_r}{V_1}(T_3-T_1)+\frac{-\Delta H_1}{\rho C_p}k_1 e^{-E_1/RT_1}C_{A1}\\ &\quad+\frac{-\Delta H_2}{\rho C_p}k_2 e^{-E_2/RT_1}C_{A1}+\frac{Q_1}{\rho C_p V_1}\end{aligned} \tag{19a}$$

$$\begin{aligned}\frac{dC_{A1}}{dt} &= \frac{F_{10}}{V_1}(C_{A10}-C_{A1})\\ &\quad+\frac{F_r}{V_1}(C_{Ar}-C_{A1})-k_1 e^{-E_1/RT_1}C_{A1}-k_2 e^{-E_2/RT_1}C_{A1}\end{aligned} \tag{19b}$$

$$\frac{dC_{B1}}{dt} = \frac{-F_{10}}{V_1}C_{B1}+\frac{F_r}{V_1}(C_{Br}-C_{B1})+k_1 e^{-E_1/RT_1}C_{A1} \tag{19c}$$

$$\frac{dC_{C1}}{dt} = \frac{-F_{10}}{V_1}C_{C1}+\frac{F_r}{V_1}(C_{Cr}-C_{C1})+k_2 e^{-E_2/RT_1}C_{A1} \tag{19d}$$

$$\begin{aligned}\frac{dT_2}{dt} &= \frac{F_1}{V_2}(T_1-T_2)+\frac{F_{20}+\Delta F_{20}}{V_2}(T_{20}-T_2)\\ &\quad+\frac{-\Delta H_1}{\rho C_p}k_1 e^{-E_1/RT_2}C_{A2}+\frac{-\Delta H_2}{\rho C_p}k_2 e^{-E_2/RT_2}C_{A2}\\ &\quad+\frac{Q_2}{\rho C_p V_2}\end{aligned} \tag{19e}$$

$$\begin{aligned}\frac{dC_{A2}}{dt} &= \frac{F_1}{V_2}(C_{A1}-C_{A2})+\frac{F_{20}+\Delta F_{20}}{V_2}(C_{A20}-C_{A2})\\ &\quad-k_1 e^{-E_1/RT_2}C_{A2}-k_2 e^{-E_2/RT_2}C_{A2}\end{aligned} \tag{19f}$$

$$\frac{dC_{B2}}{dt} = \frac{F_1}{V_2}(C_{B1}-C_{B2})-\frac{F_{20}+\Delta F_{20}}{V_2}C_{B2}+k_1 e^{-E_1/RT_2}C_{A2} \tag{19g}$$

Table 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 volatilities 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

$$\frac{dC_{C2}}{dt} = \frac{F_1}{V_2}(C_{C1}-C_{C2})-\frac{F_{20}+\Delta F_{20}}{V_2}C_{C2}+k_2 e^{-E_2/RT_2}C_{A2} \tag{19h}$$

$$\frac{dT_3}{dt} = \frac{F_2}{V_3}(T_2-T_3)-\frac{H_{vap}F_r}{\rho C_p V_3}+\frac{Q_3}{\rho C_p V_3} \tag{19i}$$

$$\frac{dC_{A3}}{dt} = \frac{F_2}{V_3}(C_{A2}-C_{A3})-\frac{F_r}{V_3}(C_{Ar}-C_{A3}) \tag{19j}$$

$$\frac{dC_{B3}}{dt} = \frac{F_2}{V_3}(C_{B2}-C_{B3})-\frac{F_r}{V_3}(C_{Br}-C_{B3}) \tag{19k}$$

$$\frac{dC_{C3}}{dt} = \frac{F_2}{V_3}(C_{C2}-C_{C3})-\frac{F_r}{V_3}(C_{Cr}-C_{C3}) \tag{19l}$$

The definitions for the variables used in Eq. (19) can be found in Table 1, with the parameter values given in Table 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 the overhead stream relative to the composition of the liquid holdup in the flash tank:

$$\begin{aligned}&C_{Ar}=\frac{\alpha_A C_{A3}}{K},\qquad C_{Br}=\frac{\alpha_B C_{B3}}{K},\qquad C_{Cr}=\frac{\alpha_C C_{C3}}{K}\\ &K=\alpha_A C_{A3}\frac{MW_A}{\rho}+\alpha_B C_{B3}\frac{MW_B}{\rho}+\alpha_C C_{C3}\frac{MW_C}{\rho}+\alpha_D x_D \rho\end{aligned} \tag{20}$$

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

Table 2
Parameter values.

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

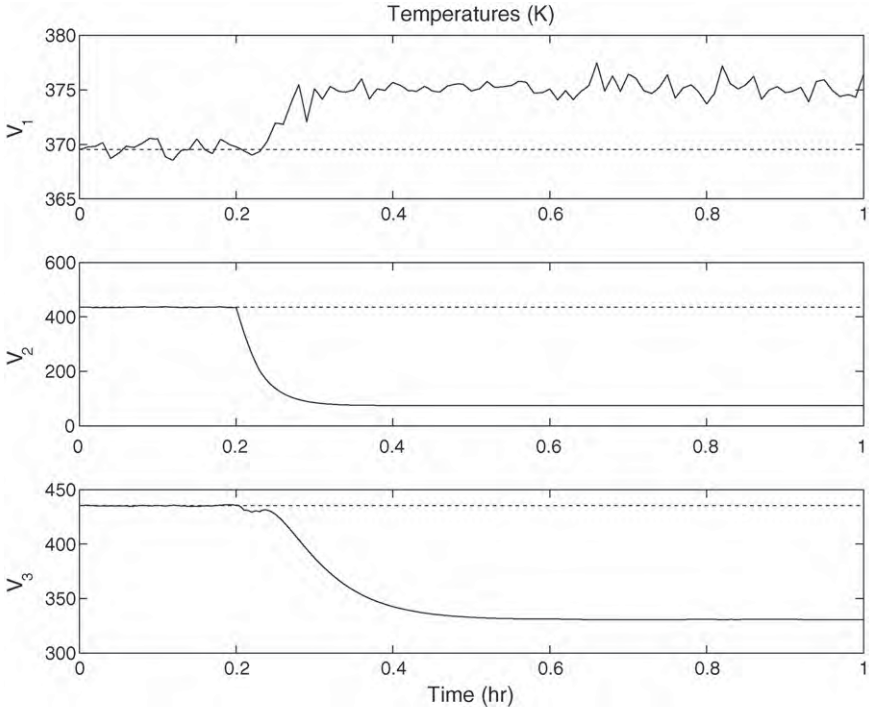

Fig. 3. Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. No FTC is implemented.

The system of Eq. (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 $\Delta_m = 0.001$ h. 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. (19) and changes with a frequency of $\Delta_p = 0.001$ h. In all three vessels, the heat input/removal is a manipulated variable for controlling the reactors at the appropriate operating 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

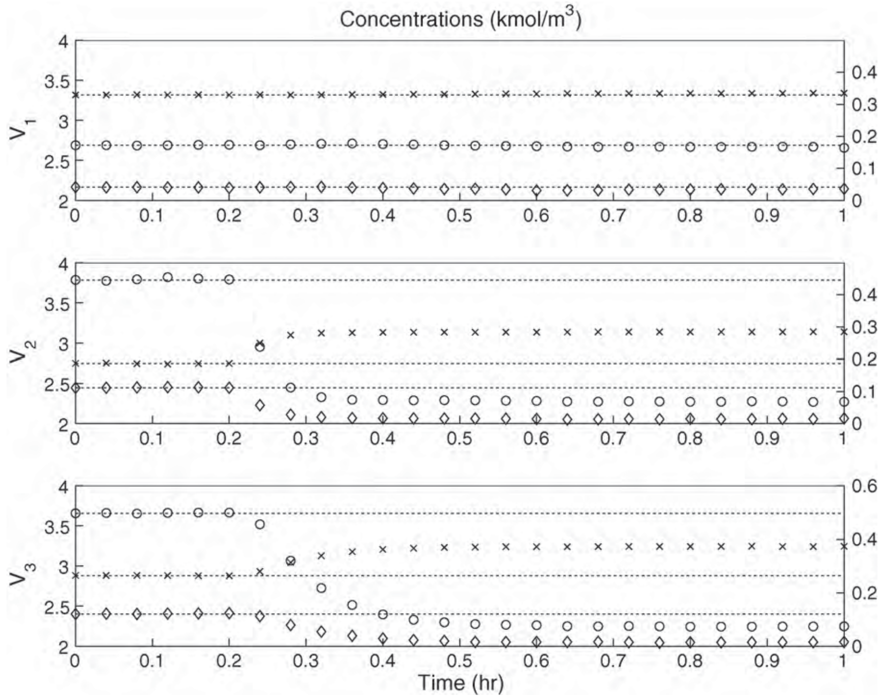

Fig. 4. Concentration profiles ($C_A = \times$ [left axis], $C_B = \bigcirc$, $C_C = \Diamond$ [right axis]) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. No FTC is implemented.

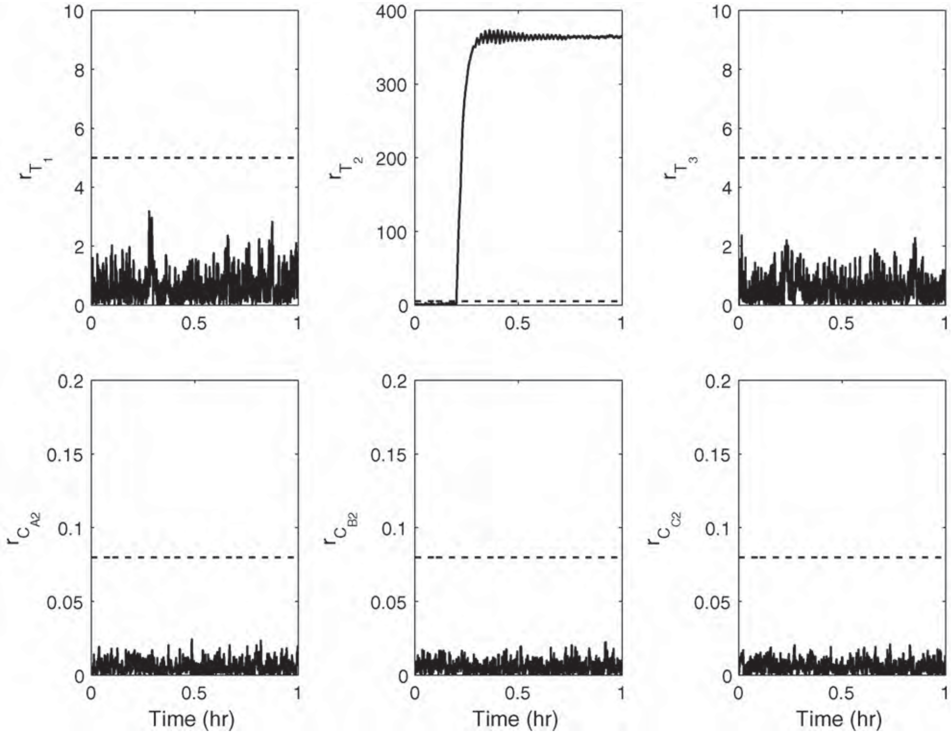

Fig. 5. FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. No FTC is implemented.

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 $u_1 = [\,Q_1 \quad Q_2 \quad Q_3\,]^T$; the second set of manipulated input is 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 $|Q_i| \leq 10^6$ kJ/h $(i = 1, 2, 3)$ and $|\Delta F_{20}| \leq 5\,\text{m}^3/\text{h}$.

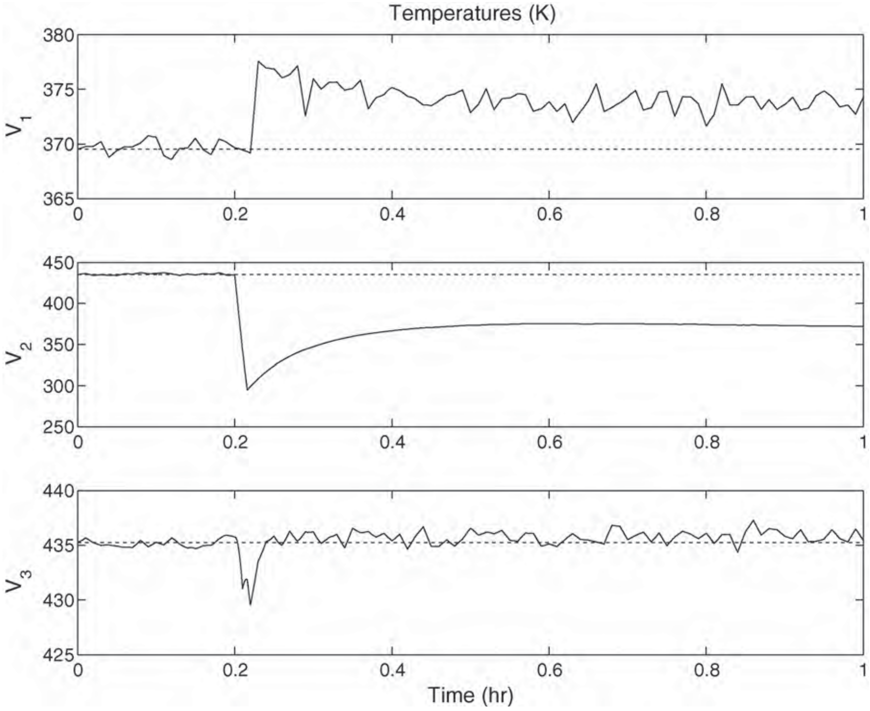

Fig. 6. Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15a) is implemented.

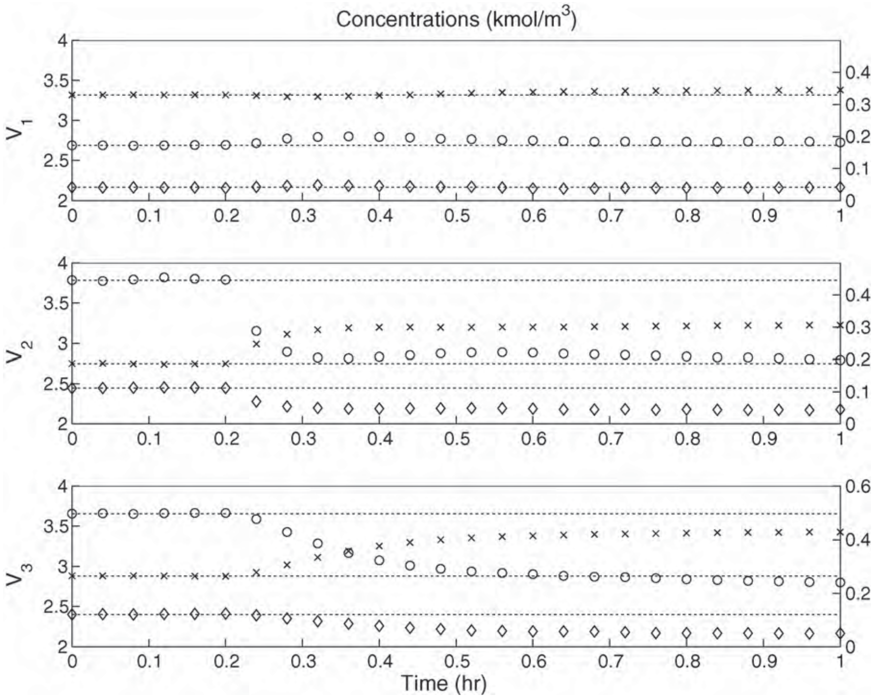

Fig. 7. Concentration profiles ($C_A = \times$ [left axis], $C_B = \bigcirc$, $C_C = \Diamond$ [right axis]) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15a) is implemented, but cannot stabilize T_2 and T_3 to the desired steady-state.

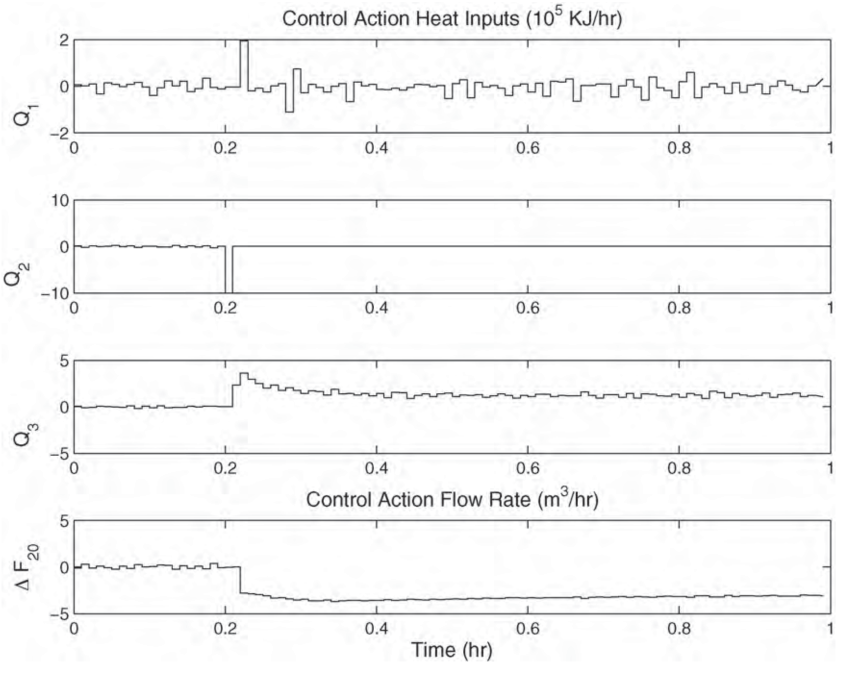

Fig. 8. Control actions with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15a) is implemented.

We consider a quadratic Lyapunov function $V(x) = x^T Px$ with $P = diag([20\quad 10^3\quad 10^3\quad 10^3\quad 10\quad 10^3\quad 10^3\quad 10^3\quad 10\quad 10^3\quad 10^3\quad 10^3])$ [4] 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 $u_1 = [Q_1\quad Q_2\quad Q_3]$ 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. (6) and (5), respectively. This control design is the fault-free control configuration for the closed-loop system. In the design of

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

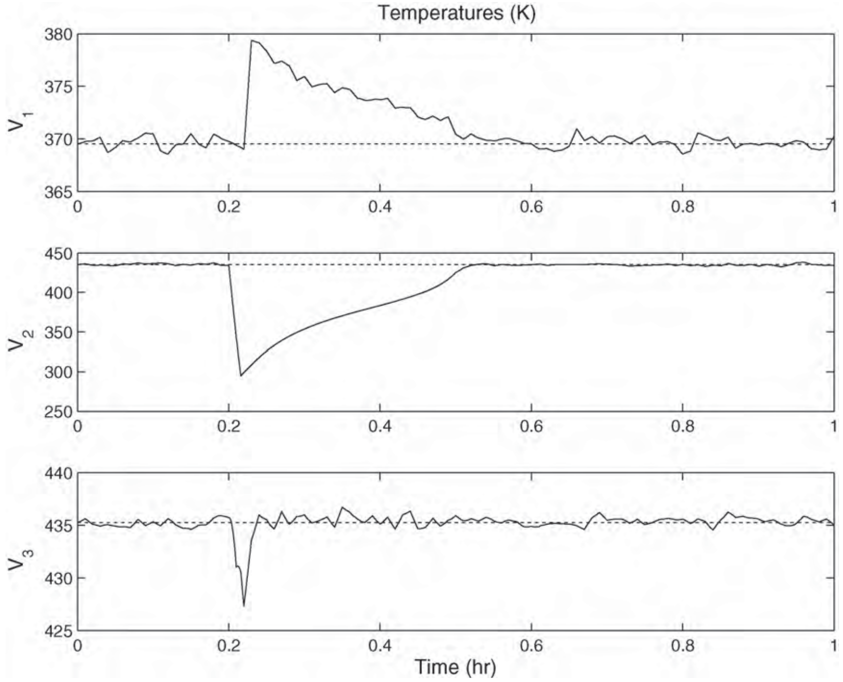

Fig. 9. Temperature profiles for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15) is implemented.

the LMPC controllers, the weighting matrices are chosen to be $Q_c = diag([20\quad 10^3\quad 10^3\quad 10^3\quad 10\quad 10^3\quad 10^3\quad 10^3\quad 10\quad 10^3\quad 10^3\quad 10^3])$, $R_1 = diag([(5\quad 5\quad 5)\cdot 10^{-12}])$ and $R_2 = 100$. The horizon for the optimization problem is $N = 5$ with a time step of $\Delta = 0.01$ h.

In addition, the control input u_1 can be divided into two sets, $u_{11} = [Q_1\quad Q_3]^T$ 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. (10)–(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 $K^b_{p1} = K^b_{p2} = 8000$, $K^b_{p3} = -0.3$ and integral time constants $\tau^b_{I1} = \tau^b_{I2} = \tau^b_{I3} = 10$ 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}.

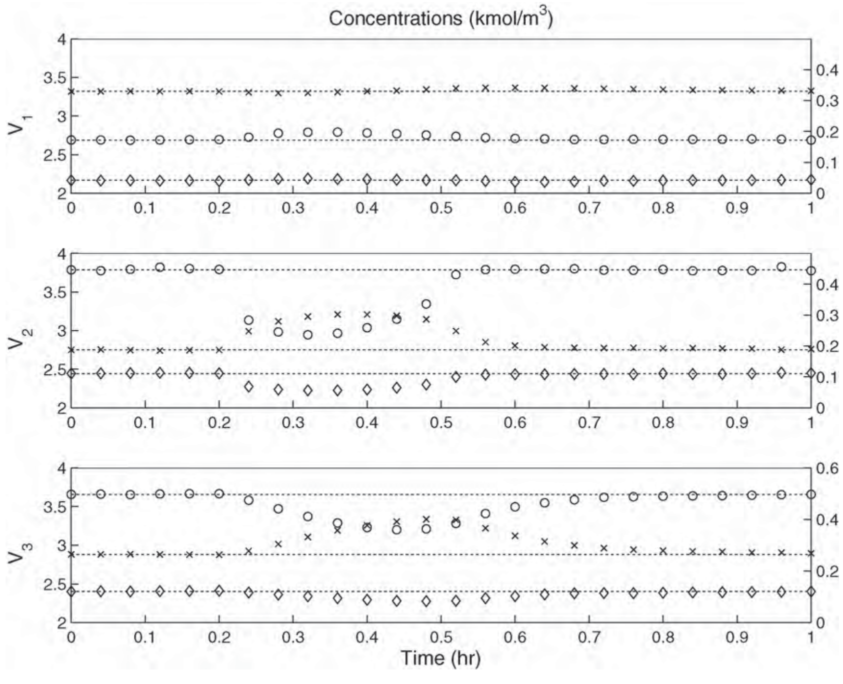

Fig. 10. Concentration profiles ($C_A = \times$ [left axis], $C_B = \bigcirc$, $C_C = \Diamond$ [right axis]) for each vessel with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15) is implemented.

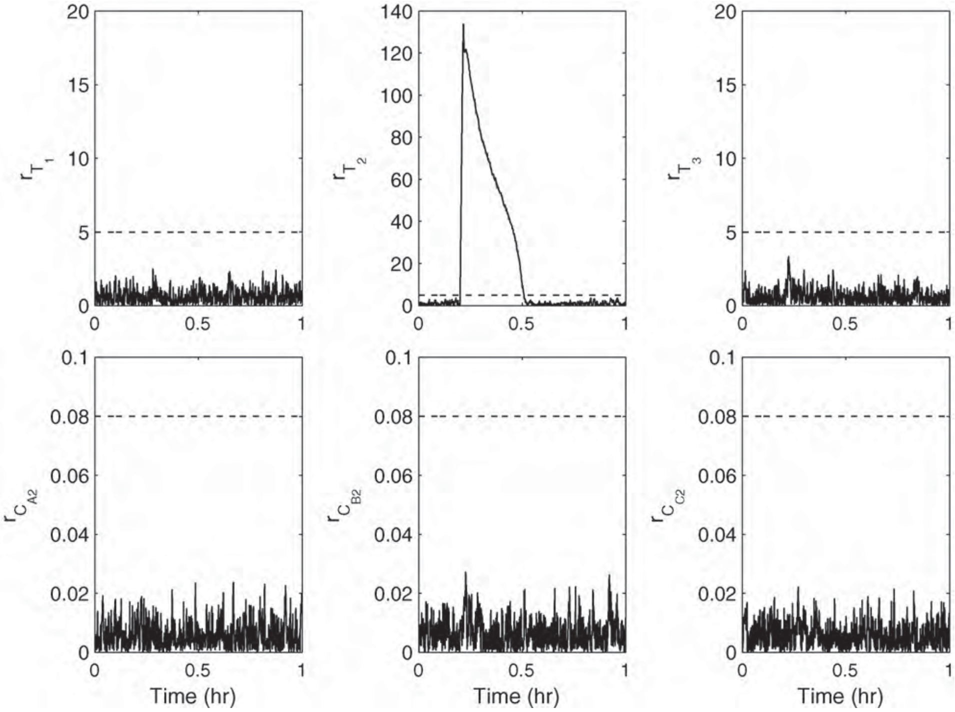

Fig. 11. FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15) is implemented.

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. (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:

$$\begin{aligned} r_{T_i}(t) &= |\hat{T}_i(t) - T_i(t)|, \quad i = 1, 2, 3 \\ r_{C_{i2}}(t) &= |\hat{C}_{i2}(t) - C_{i2}(t)|, \quad i = A, B, C. \end{aligned} \tag{21}$$

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

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

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

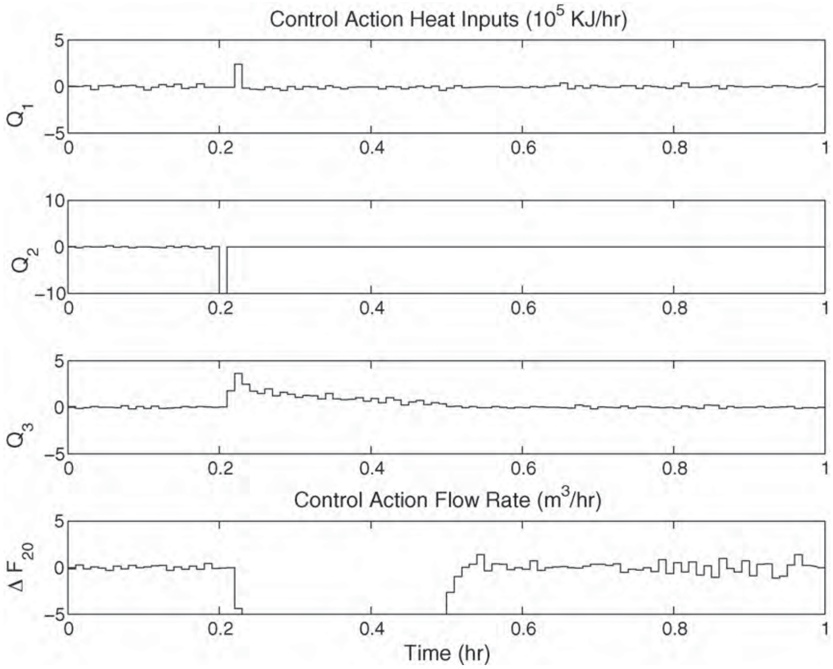

Fig. 12. Control actions with a fault in the heat input/removal actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.201$ h and isolated at $t = 0.216$ h. The FTC switching rule of Eq. (15) is implemented.

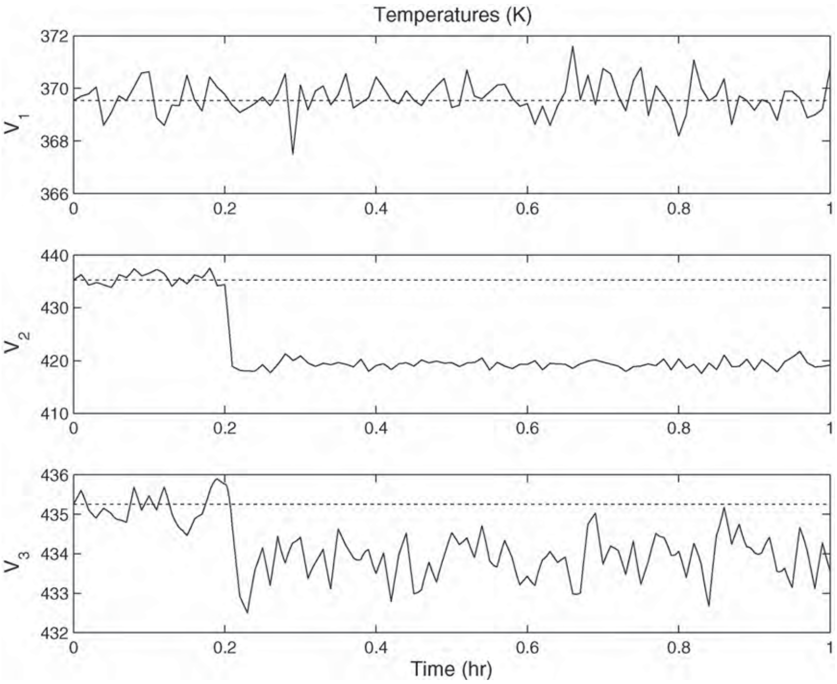

Fig. 13. Temperature profiles for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.204$ h and isolated at $t = 0.219$ h. No FTC is implemented.

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 $t_w = 0.015$ h.

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.

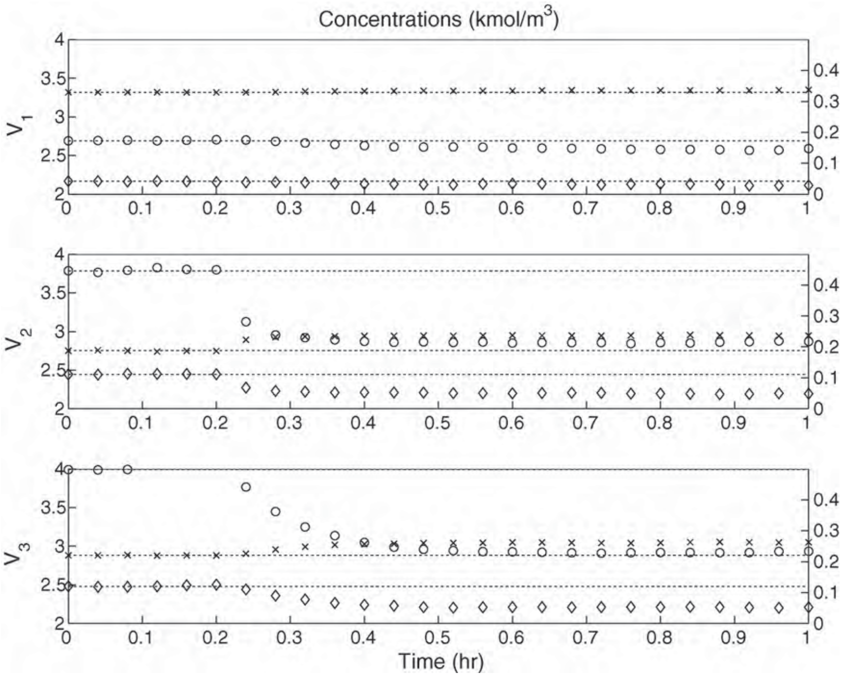

Fig. 14. Concentration profiles ($C_A = \times$ [left axis], $C_B = \bigcirc$, $C_C = \Diamond$ [right axis]) for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.204$ h and isolated at $t = 0.219$ h. No FTC is implemented.

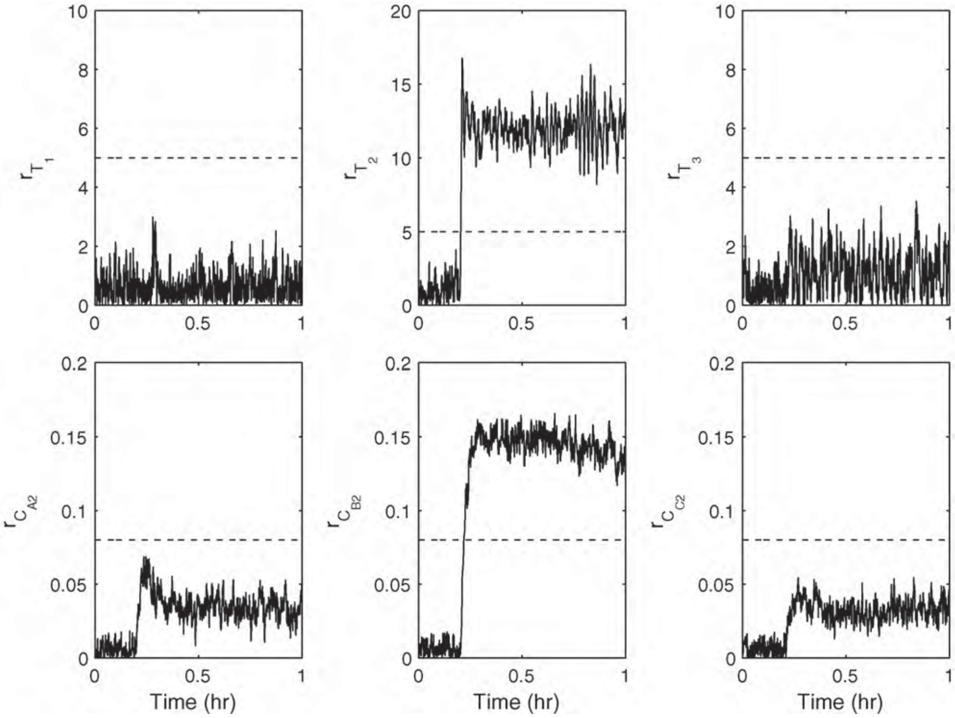

Fig. 15. FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.204$ h and isolated at $t = 0.219$ h. No FTC is implemented.

4.2. Simulation results

In the following simulations, the plant is initialized at the target steady state (x_{uss}) and simulated up to $t = 1.0$ h with a fault being triggered at time $t = 0.2$ h for all the simulations. The process and measurement noise bounds used were $w_p = [2.5 \quad 0.25 \quad 0.25 \quad 0.25 \quad 2.5 \quad 0.25 \quad 0.25 \quad 0.25 \quad 2.5 \quad 0.25 \quad 0.25 \quad 0.25]$ and $w_m = 0.1 w_p$, respectively.

First, we consider the fault in the heat input/removal actuator to vessel 2 which renders $Q_2 = -10^6$ kJ/h. Figs. 3 and 4 show the temperature and concentration profiles for each vessel when all controlled actuators are completely functional up to time $t = 0.2$ h, using $u_1 = [Q_1 \quad Q_2 \quad Q_3]^T$ 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 $t = 0.2$ h a fault is triggered and the fault is detected at time $t = 0.201$ h and correctly isolated to a fault

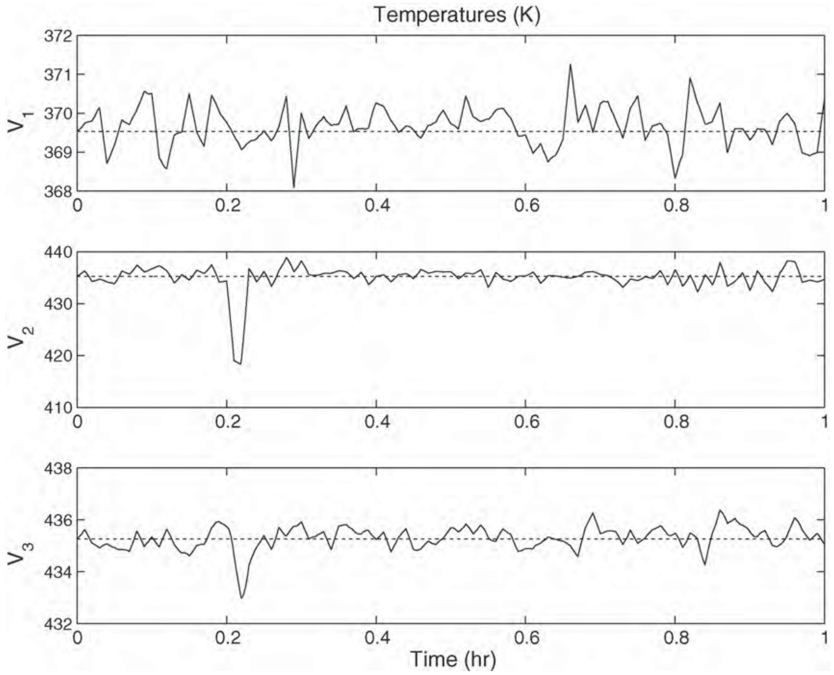

Fig. 16. Temperature profiles for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.204$ h and isolated at $t = 0.219$ h. The FTC switching rule of Eq. (14) is implemented.

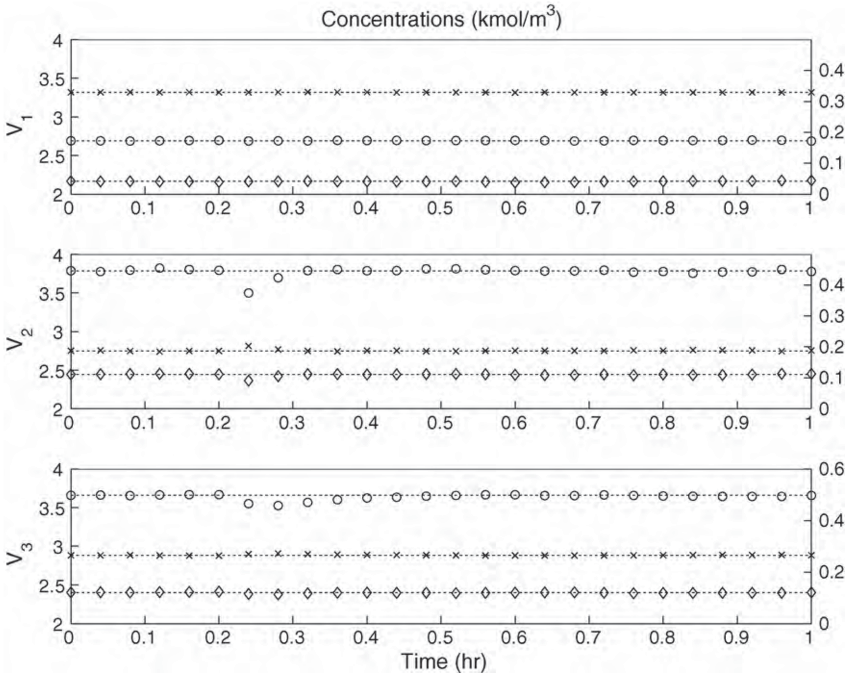

Fig. 17. Concentration profiles ($C_A = \times$ [left axis], $C_B = \bigcirc$, $C_C = \Diamond$ [right axis]) for each vessel with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.204$ h and isolated at $t = 0.219$ h. The FTC switching rule of Eq. (14) is implemented.

in Q_2 at time $t = 0.216$ h; we see that the control system cannot stabilize the process at the unstable steady state. Fig. 5 shows the corresponding residuals with no FTC.

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 $u_{11} = [Q_1 \quad Q_3]^T$ (i.e., from $u_1 = u_1^L$ to $u_1 = [u_{11}^b \quad 0]^T$) 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. (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. 6 and 7. From Figs. 6 and 7, we see that the system cannot be stabilized using only the switching rule of Eq. (15a). As shown in Fig. 8, using only the FTC switching rule of Eq. (15a), the con-

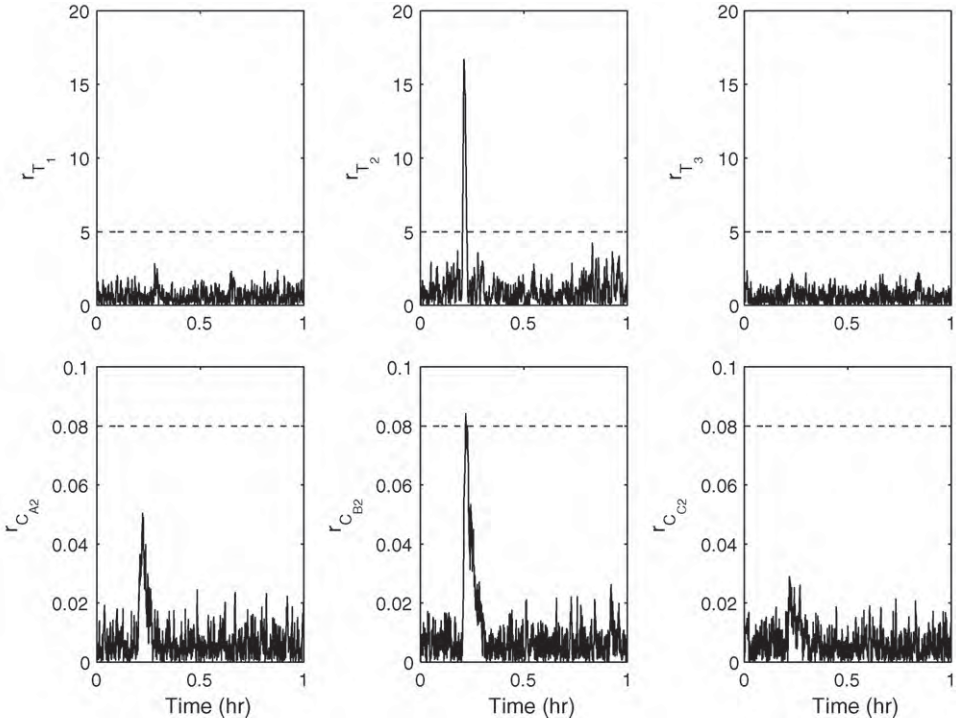

Fig. 18. FDI filter residuals for temperatures (T_1, T_2, T_3) and concentration (C_{A2}, C_{B2}, C_{C2}) with a fault in the inlet flow actuator to vessel 2 at $t = 0.2$ h. Fault is detected at $t = 0.204$ h and isolated at $t = 0.219$ h. The FTC switching rule of Eq. (14) is implemented.

trol 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. 12 for the profile of u_2 when the complete FTC switching rule of Eq. (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. (15) where the LMPC control law of u_2 is also updated to account for the complimentary controller $u_{11} = [Q_1 \quad Q_3]^T$ controlling only two heat input/removal actuators. Figs. 9 and 10 show the temperature and concentration profiles for each vessel when the fault in Q_2 is triggered at $t = 0.2$ h and FTC is carried out when the fault is isolated. Fig. 11 shows the corresponding residuals, where the fault is detected at time $t = 0.201$ h and isolated at $t = 0.216$ h. 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. (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 $t = 0.60$ h 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. 12, where both u_1 and u_2 controllers are reconfigured, to Fig. 8, where only u_1 is reconfigured.

Next, we consider a fault in the inlet flow control actuator of vessel 2, F_{20} which renders $\Delta F_{20} = 5\,\text{m}^3/\text{h}$. Figs. 13 and 14 show the temperature and concentration profiles for each vessel when the fault in F_{20} is triggered at $t = 0.2$ h and no FTC is implemented; we see that the control system cannot stabilize the process at the desired steady state. Fig. 15 shows the corresponding residuals, from which we see that the fault is detected at $t = 0.204$ h when the residual of T_2 exceeds its threshold and the fault can be isolated at $t = 0.219$ h 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 $t = 0.204$ h and isolation of the fault at $t = 0.219$ h, we carry out the switching rule of Eq. (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 3.2 that u_1 with $u_2 = 0$ can stabilize the trajectories toward the set-point. Figs. 16 and 17 show the temperature and concentration profiles for each vessel when the F_{20} fault is triggered at $t = 0.2$ h 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. (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. 18, where we see that an actuator fault in F_{20} significantly affects the residuals corresponding to T_2 and C_{B2}.

5. Conclusions

In this work, a model-based fault detection and isolation and fault-tolerant control system was designed for the monitoring and reconfiguration of distributed model predictive control systems applied to nonlinear processes in the presence of control actuator faults. Specific fault tolerant control 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.

Acknowledgments

Financial support from NSF and European Commission, INFSOICT-223866, is gratefully acknowledged.

References

[1] E.B. Ydstie, New vistas for process control: Integrating physics and communication networks, AIChE Journal 48 (2002) 422–426.

[2] P.D. Christofides, J.F. Davis, N.H. El-Farra, D. Clark, K.R.D. Harris, J.N. Gipson, Smart plant operations: vision, progress and challenges, AIChE Journal 53 (2007) 2734–2741.

[3] E. Camponogara, D. Jia, B.H. Krogh, S. Talukdar, Distributed model predictive control, IEEE Control Systems Magazine 22 (2002) 44–52.

[4] J.B. Rawlings, B.T. Stewart, Coordinating multiple optimization-based controllers: new opportunities and chanllenges, Journal of Process Control 18 (2008) 839–845.

[5] R. Scattolini, Architectures for distributed and hierarchical model predictive control—a review, Journal of Process Control 19 (2009) 723–731.

[6] W.B. Dunbar, Distributed receding horizon control of dynamically coupled nonlinear systems, IEEE Transactions on Automatic Control 52 (2007) 1249–1263.

[7] A. Richards, J.P. How, Robust distributed model predictive control, International Journal of Control 80 (2007) 1517–1531.

[8] D. Jia, B. Krogh, Min-max feedback model predictive control for distributed control with communication, in: Proceedings of the American Control Conference, Anchorage, 2002, pp. 4507–4512.

[9] A.N. Venkat, J.B. Rawlings, 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, Seville, Spain, 2005, pp. 6680–6685.

[10] T. Keviczky, F. Borrelli, G.J. Balas, Decentralized receding horizon control for large scale dynamically decoupled systems, Automatica 42 (2006) 2105–2115.

[11] L. Magni, R. Scattolini, Stabilizing decentralized model predictive control of nonlinear systems, Automatica 42 (2006) 1231–1236.

[12] D.M. Raimondo, L. Magni, R. Scattolini, Decentralized MPC of nonlinear system: an input-to-state stability approach, International Journal of Robust and Nonlinear Control 17 (2007) 1651–1667.

[13] J.M. Maestre, D. Muñoz de la Peña, E.F. Camacho, A distributed MPC scheme with low communication requirements, in: Proceedings of 2009 American Control Conference, Saint Louis, MO, USA, 2009, pp. 2797–2802.

[14] J. Liu, D. Muñoz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear process systems, AIChE Journal 55 (2009) 1171–1184.

[15] J. Liu, D. Muñoz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements, Automatica 46 (2010) 52–61.

[16] J. Liu, X. Chen, D. Muñoz de la Peña, P.D. Christofides, Sequential and iterative architectures for distributed model predictive control of nonlinear process systems, AIChE Journal, doi:10.1002/aic.12155, in press.

[17] P.M. Frank, Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results, Automatica 26 (1990) 459–474.

[18] P.M. Frank, X. Ding, Survey of robust residual generation and evaluation methods in observer-based fault detection systems, Journal of Process Control 7 (1997) 403–424.

[19] S.H. Zad, M. Massoumnia, Generic solvability of the failure detection and identification problem, Automatica 35 (1999) 887–893.

[20] N. Mehranbod, M. Soroush, C. Panjapornpon, A method of sensor fault detection and identification, Journal of Process Control 15 (2005) 321–339.

[21] C. DePersis, A. Isidori, A geometric approach to nonlinear fault detection and isolation, IEEE Transactions on Automatic Control 46 (2001) 853–865.

[22] C. DePersis, A. Isidori, On the design of fault detection filters with game-theoretic-optimal sensitivity, International Journal of Robust & Nonlinear Control 12 (2002) 729–747.

[23] F. Ahmed-Zaid, P.A. Ioannou, K. Gousman, R. Rooney, Accommodation of failures in the flight control system of the F-16 aircraft using adaptive control, IEEE Control Systems Magazine 11 (1991) 73–78.

[24] R.J. Patton, Fault-tolerant control systems: the 1997 situation, in: Proceedings of the IFAC Symposium SAFEPROCESS 1997, Hull, United Kingdom, 1997, pp. 1033–1054.

[25] P. Mhaskar, A. Gani, N.H. El-Farra, P.D. Christofides, J.F. Davis, Integrated fault-detection and fault-tolerant control of process systems, AIChE Journal 52 (2006) 2129–2148.

[26] P. Mhaskar, A. Gani, C. McFall, P.D. Christofides, J.F. Davis, Fault-tolerant control of nonlinear process systems subject to sensor data losses, AIChE Journal 53 (2007) 654–668.

[27] N.H. El-Farra, S. Ghantasala, Actuator fault isolation and reconfiguration in transport-reaction processes, AIChE Journal 53 (2007) 1518–1537.

[28] S. Ghantasala, N.H. El-Farra, Robust actuator fault isolation and management in constrained uncertain parabolic PDE systems, Automatica 45 (2009) 2368–2373.

[29] M.A. Demetriou, K. Ito, R.C. Smith, Adaptive monitoring and accommodation of nonlinear actuator faults in positive real infinite dimensional systems, IEEE Transactions on Automatic Control 52 (2007) 2332–2338.

[30] A. Armaou, M.A. Demetriou, Robust detection and accommodation of incipient component and actuator faults in nonlinear distributed processes, AIChE Journal 54 (2008) 2651–2662.

[31] H.K. Khalil, Nonlinear Systems, 2nd ed., Macmillan Publishing Company, New York, 1996.

[32] P.D. Christofides, N.H. El-Farra, Control of Nonlinear and Hybrid Process Systems: Designs for Uncertainty, Constraints and Time-Delays, Springer, New York, 2005.

[33] P. Mhaskar, C. McFall, A. Gani, P.D. Christofides, J.F. Davis, Isolation and handling of actuator faults in nonlinear systems, Automatica 44 (2008) 53–62.

[34] B. Ohran, D. Muñoz de la Peña, P.D. Christofides, J.F. Davis, Enhancing data-based fault isolation through nonlinear control, AIChE Journal 53 (2008) 2734–2741.

[35] D. Muñoz de la Peña, P.D. Christofides, Lyapunov-based model predictive control of nonlinear systems subject to data losses, IEEE Transactions on Automatic Control 53 (2008) 2076–2089.