Monitoring and Handling of Actuator Faults in a Distributed Model Predictive Control System

David Chilin, Jinfeng Liu, David Muñoz de la Peña, Panagiotis D. Christofides and James F. Davis

Abstract—In this work, we focus on monitoring and handling of actuator faults in a distributed model predictive control (DMPC) system to maintain operation at a desired steady state in the presence of control actuator faults. A model-based fault detection (FD) system is designed which detects the actuator faults in the closed-loop system and suitable strategies are devised for how to reconfigure the algorithms in the DMPC system to account for the faults and maintain stability of the closed-loop system. A chemical process example, consisting of two continuous stirred tank reactors (CSTR) and a flash tank separator with a recycle stream, is used to demonstrate the approach.

I. INTRODUCTION

Optimal operation and management of abnormal situations are major challenges in the process industries. For example, abnormal situations account for at least $10 billion in annual lost revenue in the US alone. This realization has motivated significant research in the area of process control. 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, when the number of the process state variables, manipulated inputs and measurements in a chemical plant becomes large - a common occurrence in modern plants. The computational time needed for the solution of the 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. Model predictive control is a natural control framework to deal with the design of distributed control systems because of its ability to handle input and state constraints, and also because it can account for the actions of other actuators in computing the control action of a given set of control actuators 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 [1], [2], [3] for reviews of results in this area. More specifically, in [4], the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints was considered. In [5], [6], the effect of the coupling was modeled as a bounded disturbance compensated using a robust MPC formulation. In [7], it was proven that through multiple communications between distributed controllers and using system-wide control objective functions, stability of the closed-loop system can be guaranteed. In [8], DMPC of decoupled systems (a class of systems of relevance in the context of multi-agents systems) was studied. In [9], a DMPC algorithm was proposed for the case where the system is nonlinear, discrete-time and no information is exchanged between local controllers, and in [10], DMPC for nonlinear systems was studied from an input-to-state stability point of view. In [11], [12], [13], a game theory based DMPC scheme for constrained linear systems was proposed. In a recent work [14], we proposed a DMPC architecture with one-directional communication for general nonlinear process systems. In this architecture, two separate MPC controllers 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. Generally, the computational burden of these DMPC methods is smaller compared to the one of the corresponding centralized MPC because of the formulation of optimization problems with a smaller number of decision variables. However, 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.

The focus of this paper is to demonstrate the application of distributed control with integrated fault detection and fault-tolerant control (FTC) in a plant-wide setting subject to control actuator faults. To deal with control actuator faults that may occur in the closed-loop system, an FD/FTC system is designed to detect actuator faults in a timely manner and then determine how to reconfigure the DMPC system to handle the actuator faults. The FD/FTC system uses continuous measurement of process variables like temperature and
concentrations. To design this FD system we take advantage of recent results on fault detection [15]. The method is demonstrated using a reactor-separator process consisting of two CSTRs and a flash tank separator with recycle stream.

II. PRELIMINARIES

A. Problem formulation

In this work, we consider nonlinear systems described by the following state-space model

\[ \dot{x}(t) = f(x(t), u_1(t), u_2(t), d(t)) \]  

where \( x(t) \in \mathbb{R}^{n_x} \) denotes the vector of state variables, \( u_1(t) \in \mathbb{R}^{n_u_1} \) and \( u_2(t) \in \mathbb{R}^{n_u_2} \) are two different sets of possible control (manipulated) inputs and \( d \in \mathbb{R}^p \) is a model of the set of \( p \) possible faults. The faults are unknown and \( d_j, j = 1 \ldots p \), can take any value.

We assume that \( f \) is a locally Lipschitz vector function and \( f(0, 0, 0, 0) = 0 \). This means that the origin is an equilibrium point for fault-free system \((d_j(t) \equiv 0 \text{ for all } t)\) with \( u_1 = 0 \) and \( u_2 = 0 \). System (1) is controlled with the two sets of control inputs \( u_1 \) and \( u_2 \), which could be multiple inputs of a system or a single input divided artificially into two terms (i.e., \( \dot{x}(t) = f(x(t), u_1(t), d(t)) \) with \( u(t) = u_1(t) + u_2(t) \)).

We also assume that the state \( x \) of the system is sampled continuously and continuously and the time instants that we have measurement samplings are indicated by the time sequence \( \{t_k \geq 0\} \) with \( t_k = t_0 + k\Delta, \ k = 0, 1, \ldots \) where \( t_0 \) is the initial time and \( \Delta \) is the sampling time.

B. 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 (see [16]), this assumption implies that there exist functions \( \alpha_i(\cdot), \ i = 1, 2, 3, 4 \) of class \( K^+ \) and a continuous differentiable Lyapunov function \( V(x) \) for the nominal closed-loop system that satisfy the following inequalities:

\[ \alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \]

\[ \frac{\partial V(x)}{\partial x} f(x, h(x), 0, 0) \leq -\alpha_3(|x|) \]

\[ |\frac{\partial^2 V(x)}{\partial x^2}| \leq \alpha_4(|x|) \]  

for all \( x \in D \subseteq \mathbb{R}^{n_x} \) where \( D \) is an open neighborhood of the origin and \( | \cdot | \) is used to denote the Euclidean norm of a vector. We denote the region \( \Omega_r \subseteq D^+ \) 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., \( D = \mathbb{R}^{n_x} \)), b) the construction of \( V \) can be readily done using a variety of methods (see [16], [17] 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.

C. DMPC design

Following [14], we design a DMPC architecture to maintain the closed-loop stability and performance and to reduce the computational burden in the evaluation of the optimal manipulated inputs 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 architecture 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 of \( u_2 \), it evaluates the future input trajectory of \( u_1 \) based on \( x(t_k) \) 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.

First, we present the optimization problem of LMPC 2. This optimization problem depends on \( x(t_k) \), 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, \( u_2 \) is required to satisfy a constraint that guarantees a given minimum decrease rate of the Lyapunov function \( V \). The LMPC 2 is based on the following optimization problem:

\[ \min_{u_2 \in \mathbb{S}^\Delta} \int_0^{N\Delta} [\dot{x}^T(\tau)Q_c\dot{x}(\tau) + u_{d_1}^T(\tau)R_{c_1}u_{d_1}(\tau) + u_{d_2}^T(\tau)R_{c_2}u_{d_2}(\tau)]d\tau \]  

\[ \dot{x}(\tau) = f(\dot{x}(\tau), u_{d_1}(\tau), u_{d_2}(\tau), 0) \]  

\[ u_{d_1}(\tau) = h(x(\tau)), \ \forall \tau \in [j\Delta, (j+1)\Delta), \ j = 0 \ldots N-1 \]  

\[ \dot{x}(0) = x(t_k) \]  

\[ |\frac{\partial^2 V(x)}{\partial x^2}| f(x(t_k), h(x(t_k)), u_{d_2}(0), 0) \leq |\frac{\partial V(x)}{\partial x}| f(x(t_k), h(x(t_k)), 0, 0) \]  

\[ \Omega_r \subseteq \{x \in \mathbb{R}^{n_x} | V(x) \leq r\}. \]
where \( \tilde{x} \) is the predicted trajectory of the fault-free system with \( u_2 \) being the input trajectory computed by the LMPC of Eq. 3 (i.e., LMPC 2) and \( u_1 \) being the Lyapunov-based controller \( h(x) \) applied in a sample and hold fashion. 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. 3e guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time, if \( u_1 = h(x(t_k)) \) and \( u_2 = u^*_2(0|t_k) \) are applied, is lower than or equal to the value obtained when \( u_1 = h(x(t_k)) \) and \( u_2 = 0 \) are applied.

The optimization problem of LMPC 1 depends on \( x(t_k) \) and the decision taken 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:

\[
\min_{u_1 \in \mathcal{U}(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_e \tilde{x}(\tau) + u^T_1(\tau)R_{c1}u_1(\tau)
+ u^T_2(\tau|t_k)R_{e2}u^*_2(\tau|t_k)]d\tau
\]

\[
\dot{x}(\tau) = f(\tilde{x}(\tau), u_1(\tau), u^*_2(\tau|t_k), 0)
\]

\[
\tilde{x}(0) = x(t_k)
\]

\[
\frac{\partial V(x)}{\partial x} f(x(t_k), u_1(0), u^*_2(0|t_k), 0)
\leq \frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), u^*_2(0|t_k), 0)
\]

where \( \tilde{x} \) 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 the LMPC of Eq. 4 (i.e., LMPC 1). The optimal solution to this optimization problem is denoted by \( u^*_2(\tau|t_k) \). The constraint of Eq. 4d guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time, if \( u_1 = u^*_1(0|t_k) \) and \( u_2 = u^*_2(0|t_k) \) are applied, is lower than or equal to the value obtained when \( u_1 = h(x(t_k)) \) and \( u_2 = u^*_2(0|t_k) \) are applied.

Once both optimization problems are solved, the manipulated inputs of the DMPC design based on the LMPC 1 and LMPC 2 of Eq.3-4 are defined as follows:

\[
u_1(t|x(t_k)) = u^*_1(t - t_k|t_k), \quad \forall t \in [t_k, t_{k+1})
\]

\[
u_2(t|x(t_k)) = u^*_2(t - t_k|t_k), \quad \forall t \in [t_k, t_{k+1})
\]

The above DMPC design computes the inputs \( u_1 \) and \( u_2 \) applied to the system in such a way that in the closed-loop system, the value of the Lyapunov function at time instant \( t_k \) (i.e., \( V(x(t_k)) \)) is a decreasing sequence of values with a lower bound [14]. Following Lyapunov arguments, this property guarantees practical stability of the closed-loop system. This is achieved due to the stability constraints of Eqs. 3e and 4d. Please see [14] for results on the feasibility and stability of the DMPC design. With respect to closed-loop performance, we note that, in general for nonlinear systems, there is no guarantee that the closed-loop performance of a centralized MPC should be superior than the one of a distributed MPC scheme because the MPC designs are implemented in a receding horizon scheme, the prediction horizon is finite and the optimization problems are non-convex.

D. Fault detection (FD) and Fault-tolerant control (FTC)

A filter can be constructed to estimate the fault-free evolution of states in order to detect and isolate a possible fault. The filter state, \( \hat{x} \), and measurement state, \( x \), are set equal at some initial time \( (\hat{x}(0) = (x(0)) \) and allowed to propagate forward in time. The filter takes the following form:

\[
\dot{\hat{x}} = f(\hat{x}, u_1(\hat{x}), u_2(\hat{x}), d)
\]

With a little abuse of notation, we have dropped the time index of the LMPC inputs and denote \( u_1(t|\hat{x}(t)) \) with \( u_1(\hat{x}) \) in order to simplify the FD filter definitions. The information generated by this filter provides a fault-free estimate of the real state at any time \( t \) and easy detection of the real system deviation due to a fault.

The FD residual can be defined as [18], [19]:

\[
r(t) = |\dot{\hat{x}}(t) - \dot{x}(t)|
\]

The residual \( r \) is computed continuously because \( \dot{\hat{x}}(t) \) is known for all \( t \) and the state measurement, \( x \), is also available for all \( t \). Under no-fault conditions (\( d \equiv 0 \)) with \( \dot{\hat{x}}(0) = \dot{x}(0) \), the filter states will track the true process states. In this case the FD filter, \( \dot{\hat{x}} \), and state measurement, \( x \), are identical and \( r(t) = 0 \).

When a fault \( d_j \) occurs, the residual, \( r \), will become nonzero. The objective of the FD scheme is to quickly detect when an actuator fault has occurred. Once a fault is detected, the monitoring system will declare a fault alarm or trigger the FTC system which will send the fault information and re-configuration policy to the two distributed controllers (see Fig. 1). 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, due to 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. Specifically, FTC in this case can be achieved as follows: When there is a fault in the loop of \( u_2 \), the FTC strategy is to shut down the control actions of \( u_2 \) and leave the rest of the control system to stabilize the closed-loop system. Because it is unnecessary to use \( u_2 \) in the stabilization of the closed-loop system, this
FTC strategy will maintain the closed-loop stability, however, the performance of the closed-loop system may be reduced to some extent. When there is a fault in the loop of $u_1$, whether FTC can be successfully carried out depends on the remaining control actions if there are no backup control loops.

From the analysis of Section II-C, 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 $u_1$, say $u_{11}$, together with $u_2$ that can stabilize the closed-loop system as well. Let us denote the other subset of $u_1$ as $u_{12}$. When there is a fault in the subset $u_{12}$, the FTC strategy would be to shut down the control action $u_{12}$ and re-configure the DMPC algorithm as in [20] to manipulate $u_{11}$ and $u_2$ to control the process. In [20], the DMPC system is designed based on a Lyapunov-based controller which manipulates all the available control inputs (i.e., $u_{11}$ and $u_2$) and can stabilize the closed-loop system asymptotically. Due to space limitations, we do not present the explicit formulation of the LMPC designs in this manuscript. When there is a fault in the subset $u_{11}$, it is impossible to successfully carry out FTC without activating backup actuators. These concepts will be demonstrated in the context of the example in the next section.

III. APPLICATION TO A REACTOR-SEPARATOR PROCESS

A. Process description and modeling

We demonstrate FD/FTC of the DMPC system of Fig. 1 using a three vessel reactor-separator chemical process as shown in Fig. 2. The first two vessels are assumed to be ideal continuous stirred tank reactors (CSTRs), followed by a flash tank separator. There is a fresh feed stream of pure reactant $A$ to both reactors ($F_{10}$ and $F_{20}$) and a recycle stream ($F_r$) from the flash tank to the first reactor. The overhead vapor from the flash tank is condensed and recycled to the first CSTR, and the bottom product stream is removed. The effluent of vessel 1 is fed to vessel 2, the effluent from vessel 2 is fed to the flash tank. Each vessel has an external heat input ($Q_1$, $Q_2$ and $Q_3$). There are two chemical reactions considered in this process. In the first reaction, reactant $A$ is converted to desired product $B$ (referred to as reaction 1) and in the second reaction, reactant $A$ is converted to undesired product $C$ (referred to as reaction 2). The solvent does not react. The dynamic energy and material balance equations used to describe this process are presented in Eq. 6. The description of the process variables can be found in [14] and the values of the parameters are given in Table I, they are operationally different from [14] to demonstrate FD/FTC around an unstable steady state.

\[
\frac{dT_1}{dt} = \frac{F_{10}}{V_1}(T_{10} - T_1) + \frac{F_r}{V_1}(T_3 - T_1) + u_1 + \frac{\Delta H_1}{\rho C_p}k_1 \frac{r_1}{\rho C_p} C_A + \frac{-\Delta H_2}{\rho C_p}k_2 \frac{r_2}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{A1}}{dt} = \frac{F_{10}}{V_1}(C_{A10} - C_{A1}) + \frac{F_r}{V_1}(C_{Ar} - C_{A1}) - k_1 \frac{r_1}{\rho C_p} C_A + \frac{-\Delta H_2}{\rho C_p}k_2 \frac{r_2}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{B1}}{dt} = \frac{-F_{10}}{V_1}C_{B1} + \frac{F_r}{V_1}(C_{Br} - C_{B1}) + k_1 \frac{r_1}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{C1}}{dt} = \frac{-F_{10}}{V_1}(C_{C1} + \frac{F_r}{V_1}(C_{Cr} - C_{C1})) + k_2 \frac{r_2}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dT_2}{dt} = \frac{F_1}{V_2}(T_1 - T_2) + \frac{F_{20}}{V_2}(T_{20} - T_2) + u_2 + \frac{-\Delta H_1}{\rho C_p}k_1 \frac{r_1}{\rho C_p} C_A + \frac{-\Delta H_2}{\rho C_p}k_2 \frac{r_2}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{A2}}{dt} = \frac{F_1}{V_2}(C_{A1} - C_{A2}) + \frac{F_{20}}{V_2}(C_{A20} - C_{A2}) - k_1 \frac{r_1}{\rho C_p} C_A + \frac{-\Delta H_2}{\rho C_p}k_2 \frac{r_2}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{B2}}{dt} = \frac{F_1}{V_2}(C_{B1} - C_{B2}) + \frac{F_{20}}{V_2}(C_{B20} - C_{B2}) + k_1 \frac{r_1}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{C2}}{dt} = \frac{F_1}{V_2}(C_{C1} - C_{C2}) + \frac{F_{20}}{V_2}(C_{C20} - C_{C2}) + k_2 \frac{r_2}{\rho C_p} C_A + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dT_3}{dt} = \frac{F_2}{V_3}(T_2 - T_3) + \frac{H_{exp}}{\rho Cp V_3} + u_3.
\]

\[
\frac{dC_{A3}}{dt} = \frac{F_2}{V_3}(C_{A2} - C_{A3}) + \frac{F_r}{V_3}(C_{Ar} - C_{A3}) + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{B3}}{dt} = \frac{F_2}{V_3}(C_{B2} - C_{B3}) + \frac{F_r}{V_3}(C_{Br} - C_{B3}) + \Delta m \frac{\Delta m}{hr}.
\]

\[
\frac{dC_{C3}}{dt} = \frac{F_2}{V_3}(C_{C2} - C_{C3}) + \frac{F_r}{V_3}(C_{Cr} - C_{C3}) + \Delta m \frac{\Delta m}{hr}.
\]

The flash tank’s recycle stream is described by Eq. 7, which assumes constant relative volatility for each species within the operating temperature range. This assumption allows calculating the composition in the recycle stream relative to the composition of the liquid holdup in the flash tank. Each tank is assumed to have static holdup and the reactions in the flash tank are considered negligible.

\[
C_{A'} = \frac{\alpha_A C_{A3}}{K} = \frac{\alpha_B C_{B3}}{K} = \frac{\alpha_C C_{C3}}{K}.
\]

\[
K = \alpha_A C_{A3} \frac{MW_A}{\rho} + \alpha_B C_{B3} \frac{MW_B}{\rho} + \alpha_C C_{C3} \frac{MW_C}{\rho} + \alpha_D \frac{MW_D}{\rho}.
\]
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. 6 and changes with a frequency of \( \Delta_{\rho} = 0.001 \text{hr} \). In all three vessels, the heat inputs are a set of manipulated variables for controlling the process at the desired operating point. In addition the second tank’s inlet flow rate is another manipulated variable. The system has one unstable and two stable steady states. The operating set- point is the unstable steady state:

\[
x_{\text{ass}} = [T_1, C_{A1}, C_{B1}, C_{C1}, T_2, C_{A2}, C_{B2}, C_{C2}, T_3, C_{A3}, C_{B3}, C_{C3}]
\]

\[
= [370.332, 0.17, 0.04, 435.275, 0.45, 0.11, 435.288, 0.50, 0.12]
\]

The control system being demonstrated herein consists of two distributed model predictive controllers as described in Section II-C. The two distributed controllers are designed following the designs given in Eq. 3 and Eq. 4. LMPC 1 operates the three heat input actuators \( (u_1) = [Q_1 - Q_{1a}, Q_2 - Q_{2a}, Q_3 - Q_{3a}]^T \) and LMPC 2 operates the flow control actuator \( (u_2) = \Delta F_{20} = F_{20} - F_{20,0} \). They are designed based on three PI controllers (i.e., \( h = [Q_1, Q_2, Q_3]^T \)) and a quadratic Lyapunov function \( V(x) = x^TPx \) with \( P = \text{diag}(20, 10^3, 10^3, 10, 10^3, 10^3, 10^3, 10^3, 10^3, 10^3, 10^3, 10^3) \). The PI controllers are used as \( h(x) \) in the LMPC designs (but are not implemented on the plant) for the three heat input actuators with proportional gains \( K_{p1} = K_{p2} = K_{p3} = 8000 \) and integral time constants \( \tau_{I1} = \tau_{I2} = \tau_{I3} = 10 \). In the design of the LMPC controllers, the weighting matrices are chosen to be \( Q_1 = P, R_1 = \text{diag}(5, 5, 5) \cdot 10^{-12} \) and \( R_2 = 100 \). The horizon for the optimization problem is \( N = 5 \) with a time step of \( \Delta_{\tau} = 0.01 \text{hr} \). Note that the control variables are deviation variables, whose values at the desired unstable steady state are zero and subject to the input constraints \( |Q_i| \leq 10^6 KJ/hr, (i = 1, 2, 3) \) and \( |\Delta F_{20}| \leq 5 m^3/hr \).

In the case of fault-tolerant control of a fault in \( u_1 \), we look into designing a new set of LMPCs through decomposition of the input space. The inputs of the process \( u_1 \) can be decomposed into two sets \( [Q_1, Q_3]^T \) and \( u_{12} = Q_2 \). In the case of a fault in \( u_2 \), we know \( u_{2} \) provides extra control action and is not necessary for stabilization of the system as described in Section II-D. But in the case of an actuator fault in \( u_1 \), a redundant control system is necessary to maintain stability or in the specific case of an actuator fault in \( Q_2 \), we can design a new set of LMPCs that take advantage of the extra control action of \( u_2 \) to stabilize the plant [20]. For this process under consideration there exists a controller design \( h_2 = [Q_1, Q_3, \Delta F_{20}]^T \) with \( Q_1 \) and \( Q_3 \) controlled by a similar LMPC design as the \( h \) controller introduced before in this section. The fourth PI controller used to control the flow control actuator is based on the measurement of \( T_2 \) with the proportional gain \( K_{p4} = -0.3 \) and integral time \( \tau_{I4} = 10 \). The control design \( h_2 \) can stabilize the closed-loop system asymptotically with continuous measurements and \( Q_2 = 0 \).

From the design of LMPC 1 we can add an additional constraint to the optimization problem of Eq. 4 to account for the actuator fault within \( u_{12} \) and create a back-up control scheme that we will refer to as LMPC 1r (or simply controller \( u_{11} \)). The additional constraint takes the form of:

\[
\begin{align*}
u_{12}(\tau) & = 0. \\
\end{align*}
\]

In addition, we will redesign LMPC 2 to account for the loss of control action within \( u_1 \). The additional constraint, Eq. 8, is added to the optimization problem of Eq. 3. We will refer to the new LMPC 2 as LMPC 2r (or simply \( u_{2r} \)). Based on this \( h_2 \) controller, the new DMPC system is designed and used for fault-tolerant control purposes. In the design of the new FTC DMPC, the Lyapunov function used is the same as before.

In order to perform FD for the process, we construct the FD filter of the form in Eq. 5, where \( \hat{T}_i, \hat{C}_{Ai}, \hat{C}_{Bi}, \text{ and } \hat{C}_{Ci}, \text{ } i = 1, 2, 3 \) are the state filters and \( T_i, C_{Ai}, C_{Bi}, \text{ and } C_{Ci}, \text{ } i = 1, 2, 3 \) are the measured states (x). The filter states provide fault-free estimates for the temperature and concentration states allowing the computation of an FD filter residual:

\[
r = \sqrt{\sum_{i=1}^{3} r^2_{T_i} + \sum_{i=1}^{3} r^2_{C_{Ai}} + \sum_{i=1}^{3} r^2_{C_{Bi}} + \sum_{i=1}^{3} r^2_{C_{Ci}}} \tag{9}
\]

where \( r_{T_i}, r_{C_{Ai}}, r_{C_{Bi}} \text{ and } r_{C_{Ci}} \) take the following form:

\[
\begin{align*}
r_{T_i} & = \|\hat{T}_i(t_k) - T_i(t_k)\|, \text{ } i = 1, 2, 3 \\
r_{C_{Ai}} & = \|\hat{C}_{Ai}(t_k) - C_{Ai}(t_k)\|, \text{ } i = 1, 2, 3 \\
r_{C_{Bi}} & = \|\hat{C}_{Bi}(t_k) - C_{Bi}(t_k)\|, \text{ } i = 1, 2, 3 \\
r_{C_{Ci}} & = \|\hat{C}_{Ci}(t_k) - C_{Ci}(t_k)\|, \text{ } i = 1, 2, 3
\end{align*}
\]

Due to process and sensor measurement noise, the residuals will be nonzero even without a fault. This necessitates the use of a fault detection threshold so that a fault is declared only when the residual exceeds a specific threshold value, \( r_{max} \). This threshold value is chosen large enough to avoid false alarms due to process and sensor measurement noise, but should be sensitive enough to detect faults in a timely manner so that effective fault-tolerant control can be performed. The residual threshold value is \( r_{max} = 8.7 \).
B. Simulation Results

In the simulations below, the plant state was initialized near the target steady state \( (x_{\text{start}} = 0.99 \cdot x_{\text{uss}}) \) and simulated up to \( t = 1.0\text{hr} \). A fault is triggered at time \( t = 0.5\text{hr} \) in the heat input actuator to vessel 2. We will refer to this fault as \( Q_2 \)-fault. The process and measurement noise bounds used were \( w_p = [2.5 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25 \ 0.25] \) and \( w_m = 0.1 \cdot w_p \), respectively.

In Fig. 3 and Fig. 4, we see the temperature and concentration profiles for each vessel when the \( Q_2 \)-fault is triggered at \( t = 0.5\text{hr} \) and no FTC is implemented. We see that the control system is unable to stabilize the process at the desired steady state.

In the following simulation, FD was augmented to the DMPC system, and after a fault is detected, it is possible to reconfigure the distributed control system to stabilize the system. In this particular example the FD system detects a fault when \( r > r_{\text{max}} \) is true. In this example the FD/FTC system is setup to assume there was a \( Q_2 \)-fault and only reconfigures one of the controllers from \( u_1 \) to \( u_{11} \) to reflect the failed actuator and resets \( u_{12} = Q_2 = 0 \), while maintaining the identical LMPC controller \( u_2 \). Note that the \( u_2 \) is operating on the assumption that LMPC 1 is using all three heat input actuators.

Fig. 5 and Fig. 6 show that the system cannot be stabilized using \( u_{11} \) and \( u_2 \). Fig. 7 shows the filter residual continues to exceed the threshold value even after reconfiguration. As shown in Fig. 8 by using \( u_{11} \) and the original \( u_2 \), the control action for \( u_2 = \Delta F_{20} \) is not sufficiently large as is required for stabilization since it expects the control action of \( Q_2 \) in \( u_1 \) to help stabilize the system (please see Fig. 11 as a comparison).

The next setup is identical to the conditions tested above, where we consider a \( Q_2 \)-fault, but the FD/FTC will in addition now update the LMPC control law of \( u_2 \) to account for the complimentary controller \( u_{11} = [Q_1 \ Q_2]^T \) controlling only two heat input actuators. The reconfigured \( u_2 \) will be referred to as \( u_{2r} \). The reconfiguration of both DMPC controllers to account for failure in the heat input actuator of the second vessel allows the DMPC system to work properly to stabilize the plant as shown in Fig. 9 and Fig. 10. The FD/FTC system detects a fault at \( t = 0.504 \) and the additional reconfiguration of \( u_2 \) to \( u_{2r} \) allows the system to be stabilized with an appropriately strong control action from \( u_{2r} \). The difference in control action can clearly be seen by comparing Fig. 11, where both \( u_1 \) and \( u_2 \) controllers
Fig. 7. FD filter residual with $Q_2$-fault triggered at $t = 0.5\text{hr}$. Reconfiguration of $u_1$ to $u_{11}$ only, no change to $u_2$. Note dashed line corresponds to $r_{\text{max}} = 8.7$.

Fig. 8. Control action profile with $Q_2$-fault and FD/FTC reconfiguring only $u_1$ to $u_{11}$. Note weaker F20 control action in comparison to Fig. 11, where both DMPCs are reconfigured.

Fig. 9. Temperature profile for each vessel with $Q_2$-fault and FTC reconfiguring both $u_1$ to $u_{11}$ and $u_2$ to $u_{2r}$.

are reconfigured, to Fig. 8, where only $u_1$ is reconfigured. The temperature and concentration trajectories return near the steady state at $t = 0.86\text{hr}$ and then minimal control action is required to further maintain system stability.

In the last simulation study, the system is initiated near the steady state and a fault is triggered in the flow control actuator of vessel 2 ($F_{20}$). Similar to previous examples, the FD system detects a fault at $t = 0.509$ and implements FTC (we setup the FTC to correctly assume $F_{20}$-fault). In this particular example the FD/FTC system only reconfigures one controller by shutting down the $u_2$ controller and resetting $u_2 = \Delta F_{20} = 0$, while maintaining the LMPC controller $u_1$ identically the same. We know from Section II-A that $u_1$ with $u_2 = 0$ can asymptotically stabilize the trajectories toward the set-point. Fig. 12 and 13 show the temperature and concentration trajectories approaching the target set-points soon after reconfiguration. The residual is shown in Fig. 14.

IV. CONCLUSIONS

In this work, we studied the monitoring and reconfiguration of a DMPC system applied to a chemical process in the presence of control actuator faults. To achieve the objective of stabilization of the closed-loop system at an open-loop unstable steady state in the presence of actuator faults, a
model-based fault detection (FD) system was designed to detect the actuator faults in the closed-loop system and suitable strategies were devised to reconfigure the algorithms in the DMPC system to account for the faults and maintain stability of the closed-loop system. We demonstrated that reconfiguration of only one of the distributed controllers is not sufficient to maintain the process at the desired steady state and that reconfiguration of both distributed controllers is required, so that all distributed controllers account for the faulty actuator and work together, to maintain closed-loop stability. Extensive simulation results in the presence of process and measurement noise demonstrated the proposed approach.

REFERENCES