Chemical Engineering Science 69 (2012) 287–295

Contents lists available at SciVerse ScienceDirect

Chemical Engineering Science

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

Monitoring and retuning of low-level PID control loops

Akradej Leosirikul [a], David Chilin [a], Jinfeng Liu [a], James F. Davis [a], Panagiotis D. Christofides [a,b,*]

[a] *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA*
[b] *Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1592, USA*

ARTICLE INFO

Article history:
Received 5 September 2011
Received in revised form
16 October 2011
Accepted 20 October 2011
Available online 29 October 2011

Keywords:
Process control
PID control
Stability
Simulation
Process monitoring
Chemical processes

ABSTRACT

In this work, we focus on the problem of monitoring and retuning of low-level proportional-integral-derivative (PID) control loops used to regulate control actuators to the values computed by advanced model-based control systems like model predictive control (MPC). We consider the case where the real-time measurement of the actuation level is unavailable, and thus PID controller monitoring has to be achieved on the basis of process state measurements. A fault detection and isolation (FDI) method involving process models and real-time process measurements is used to monitor the PID control loops and compute appropriate residuals. Once poor tuning is detected and isolated, a PID tuning method based on the estimated transfer function of the control actuator is applied to the isolated, poorly functioning PID controller. An example of a non-linear reactor–separator process operating under MPC with low-level PID controllers regulating the control actuators is used to demonstrate the approach.

1. Introduction

Model predictive control (MPC) is widely used in industry because of its ability to handle input/state constraints and to incorporate optimization considerations in a single formulation (Mayne et al., 2000; Qin and Badgwell, 2003). In general, in the calculation of the optimal input trajectories for the manipulated inputs via MPC, the dynamics of the corresponding control actuators that will implement the control actions computed by the MPC are neglected and the MPC-computed control actions are assumed to be directly implemented by the control actuators. However, in practice, these control actuators have their own specific dynamics. As a result of this, there are always discrepancies (i.e., time lags, magnitude differences, etc.) between the actual control actions applied to the process by the control actuators and the control actions requested by the MPC. To mitigate the influence of these discrepancies in closed-loop performance, PID controllers (typically called “low-level” PID controllers) are usually implemented on the control actuators to regulate the outputs of the actuators at the values requested by the MPC (Astrom et al., 1993). The representation of this added extra layer of the PID controllers around the control actuators is shown in Fig. 1. In this case, the tuning of the PID controllers is critical for the overall control actuator and closed-loop system performance. An actuator with a well-tuned PID controller can effectively implement the actions requested by the MPC; whereas, an actuator with a poorly tuned PID controller may reduce the performance of the closed-loop system dramatically or may even cause instability of the closed-loop system.

Monitoring the performance of low-level PID loops provides the motivation for this work. With respect to previous works on the subject, there is indeed a plethora of techniques discussed in the literature on monitoring of the performance and tuning of PID controller parameters. With respect to tuning, methods such as Ziegler–Nichols (Ziegler and Nichols, 1942; Hang et al., 1991), Cohen–Coon (Cohen and Coon, 1953), internal model control (Skogestad, 2003; Veronesi and Visioli, 2009), pole placement (Wang et al., 2009; Zayed et al., 2009), and others have been widely used to tune PID controller parameters based on either the estimated plant's transfer function or experimentally obtained step response and/or frequency response curves. Gain scheduling (Rugh, 1991; Zhao et al., 1993) has also been developed to allow PID controllers to be able to self-tune to accommodate changing operating conditions. Multiple works have also been published on automatic retuning of PID parameters based on the current performance of the PID controller and online system identification (Sung et al., 1998; Veronesi and Visioli, 2009; Zhuang and Atherton, 1993; Chand, 1992; Saito, 1990; Teng et al., 2008; Anderson et al., 1988; Nishikawa et al., 1984). On the monitoring front, Eriksson and Isaksson (1994) and Qin (1998) provide a survey of available monitoring techniques. Specifically, minimum variance control (Harris, 1989) has been developed as a tool to assess PID performance, while Tsung and Shi (1999), Tsung et al. (1999), and Tsung (2000) utilize statistical process control (SPC)

[*] Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

0009-2509/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ces.2011.10.046

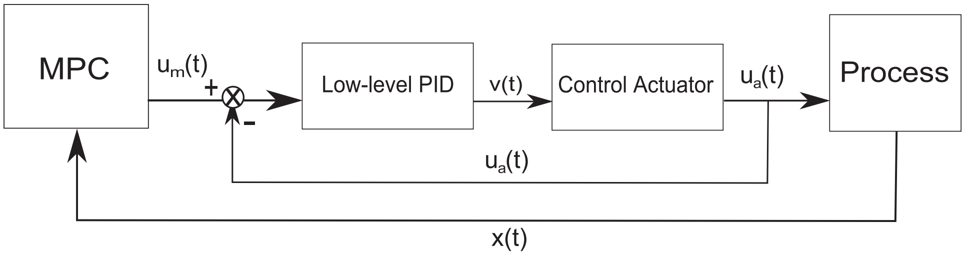

Fig. 1. Closed-loop system with MPC as advanced model-based controller and low-level PID controller implemented to regulate the control actuators.

to monitor and provide performance criteria to assess the performance of PID controllers. In another work (Shi and Tsung, 2003), a monitoring scheme was proposed to determine poor tuning/faults using principal component analysis (PCA) and neural networks. One common feature in all of the works in the PID monitoring field mentioned above is the assumption that measurements of the output of the PID-controlled loop are available.

Motivated by the above considerations, we address the problem of real-time monitoring and retuning of low-level PID controllers in the case where the measurement of the actual control action implemented on the process is unavailable. Specifically, we present a method for monitoring the PID performance via a model-based FDI method (Mhaskar et al., 2006, 2007) coupled with real-time process measurements. Using an estimated transfer function model of the control actuators, model-based FDI can be used to detect the discrepancies between the expected actuation level and the actual actuation level performed by the control actuators. Based on the patterns of the residuals, a poorly tuned actuator can be isolated and retuned accordingly. An example of a non-linear reactor–separator process under MPC with low-level PID controllers around the control actuators is used to demonstrate the approach.

2. Preliminaries

2.1. Class of non-linear systems

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

$$\dot{x}(t) = f(x(t)) + G(x(t))u_a(t) + w(t) \tag{1}$$

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

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

2.2. Lyapunov-based MPC

In this work, the model-based controller that is used to determine the set-points for each actuator is a Lyapunov-based model predictive controller (LMPC) (Mhaskar et al., 2006). One assumption about the design of the model-based control system used in this work is that it does not explicitly account for the dynamics of the control actuators and the presence of the process noise. Therefore, the model used for the design of the model-based control system assumes the following dynamics for the process:

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t)) + G(\tilde{x}(t))u_m(t) \tag{2}$$

where u_m is the computed actuation by the high-level MPC.

We make the following assumptions regarding the stability of the closed-loop system. We assume that there exists a Lyapunov-based controller $h(\tilde{x})$ such that the origin of the nominal closed-loop system under this controller, i.e., system of Eq. (2) with $u_m(t) = h(\tilde{x})\ \forall t$, is asymptotically stable; this controller $h(\tilde{x})$ can be designed using geometric control or Lyapunov-based control techniques (e.g., Christofides and El-Farra, 2005). Using converse Lyapunov theorems, this implies that there exist class $\mathcal{K}$ functions[1]$\alpha_i(\cdot)$, $i = 1,2,3,4$ and a continuously differentiable Lyapunov function $V(\tilde{x})$ for the nominal closed-loop system that satisfy the following inequalities:

$$\alpha_1(\|\tilde{x}\|) \leq V(\tilde{x}) \leq \alpha_2(\|\tilde{x}\|) \tag{3a}$$

$$\frac{\partial V(\tilde{x})}{\partial x}(f(\tilde{x}) + G(\tilde{x})h(\tilde{x})) \leq -\alpha_3(\|\tilde{x}\|) \tag{3b}$$

$$\left\| \frac{\partial V(\tilde{x})}{\partial x} \right\| \leq \alpha_4(\|\tilde{x}\|) \tag*{(3b)}$$

for all $\tilde{x} \in D \subseteq R^{n_x}$ where D is an open neighborhood of the origin. We denote the region $\Omega_\rho \subseteq D$[2] as the stability region of the nominal closed-loop system, i.e., Eq. (2), under the control $u_m(t) = h(\tilde{x})$.

The existence of the controller $h(\tilde{x})$ allows us to formulate an MPC that inherits the stability properties of $h(\tilde{x})$ (Mhaskar et al., 2006), and it is described by the following optimization problem:

$$\min_{u_c \in S(\Delta)} \int_0^{N_c \Delta} [\hat{x}^T(\tau)Q\hat{x}(\tau) + u_c^T(\tau)Ru_c(\tau)]\, d\tau \tag{4a}$$

$$\dot{\hat{x}}(\tau) = f(\hat{x}(\tau)) + G(\hat{x}(\tau))u_c(\tau) \tag{4b}$$

$$\hat{x}(0) = x(t_k) \tag{4c}$$

$$u_c(\tau) \in U \tag{4d}$$

$$\frac{\partial V(x(t_k))}{\partial x} G(x(t_k))u_c(0) \leq \frac{\partial V(x(t_k))}{\partial x} G(x(t_k))h(x(t_k)) \tag{4e}$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ, Q and R are strictly positive definite symmetric weighting matrices, $x(t_k)$ is the process state measurement

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

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

obtained at t_k, $\hat{x}$ is the predicted trajectory of the system under the MPC, N_c is the number of steps in the prediction horizon, and V is the Lyapunov function corresponding to the controller $h(\tilde{x})$.

The optimal solution to this optimization problem is denoted by $u_c^*(\tau|t_k)$. The LMPC is implemented following a receding horizon strategy; at each sampling time t_k, a new state measurement $x(t_k)$ is received from the sensors and the optimization problem of Eq. (4) is solved, and $u_c^*(0|t_k)$ is sent to the actuators and it is implemented for $t \in [t_k, t_{k+1}]$. The constraint of Eq. (4e) guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time of the LMPC is less than or equal to the value obtained if only the Lyapunov-based control $u_m = h(\tilde{x})$ is implemented. This constraint allows the LMPC to inherit the stability properties of the Lyapunov-based control $h(\tilde{x})$ for sufficiently small sampling period Δ; in particular, practical stability of the closed-loop system can be proven for sufficiently small Δ. For detailed results on Lyapunov-based MPC, see Mhaskar et al. (2006).

Remark 1. Note that in the design of the LMPC of Eq. (4) and its closed-loop stability analysis, one assumption is that the requested actuation $u_m(t)$ is applied directly to the process by the control actuators. In a practical setting, however, $u_m(t)$ has to go through the dynamics of the PID-controlled actuators before the system is actuated with $u_a(t)$. The central focus of this work is on how to bring $u_a(t)$ to be as close as possible to $u_m(t)$. The relationship between $u_a(t)$ and $u_m(t)$ will be discussed in detail in the next section.

Remark 2. Though a Lyapunov-based MPC is used in this paper as the model-based control system to demonstrate how the problem of low-level PID monitoring and retuning based on process state measurements can be approached, the monitoring and retuning methods presented here can be applied to any type of model-based control system (i.e., geometric control or Lyapunov-based control, Christofides and El-Farra, 2005; distributed MPC, Liu et al., 2009, 2010, etc.). Specifically, as long as the requested actuation level $u_m(t)$ and the process state measurements are available to the monitoring and retuning system at all times, the same method presented in this work can be applied to detect the deviation of the actual actuation level $u_a(t)$ from the requested actuation level $u_m(t)$.

2.3. Low-level PID control loops

As depicted in Fig. 1, $u_m(t)$ is sent from the model-based controller as the set-point to the control actuators. PID controllers are installed around these control actuators to help accelerate the actuator's response so that $u_a(t)$ can approach the value of $u_m(t)$ faster. Eq. (5) shows the relationship between u_m and u_a in the Laplace domain:

$$u_a(s) = \frac{G_p G_c}{1 + G_p G_c} u_m(s) \tag{5}$$

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

$$G_c = K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s\right) \tag{6}$$

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

$$G'_p = K_p \frac{e^{-\tau_d s}}{\tau_p s + 1} \tag{7}$$

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

The estimation of the actuator's transfer function (G'_p) will be needed by the FDI algorithm below when the actuator's expected behavior is calculated and also at the retuning step when a new set of PID parameters is calculated. The expected actuation level (denoted by $u'_a(t)$) will be used as the benchmark upper limit of how well the control actuators can perform. We note that the parameters of the PID controller should be tuned in such a way that the low-level closed-loop response (i.e., actuator under the PID controller) is fast relative to the sampling time of the MPC such that the actual actuator output (control action implemented on the process) is as close as possible to the control action requested by the MPC at each sampling time. A rigorous analysis of this problem can be done using singular perturbation techniques for two-time-scale processes.

3. Monitoring and retuning of low-level PID loops

We consider the case where there is no access by the monitoring system to the measurements of the actual actuation levels $u_a(t)$ implemented by the control actuators on the process. Therefore, the detection of poor PID tunings must be performed based on the measurements of the states of the process. To this end, an FDI method is used as the main tool to extract actuator behavior from the process state measurements (Chilin et al., 2010). We use exponentially weighted moving average (EWMA) residuals to detect and isolate poorly tuned PID loops. Once a poorly tuned actuator is isolated, a model-based tuning rule such as Cohen–Coon or internal model control is applied to the PID controller that regulates the poorly tuned actuator.

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

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

$$u_a(s) = \frac{G_p G_c}{1 + G_p G_c} u_m(s) \tag{8}$$

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

$$\dot{\breve{x}}_i(t) = f_i(\hat{x}_i(t)) + G_i(\hat{x}_i(t))u'_a(t)$$

$$\hat{x}_i = [x_1 \cdots x_{i-1}, \breve{x}_i, x_{i+1} \cdots x_{n_x}]^T$$

$$u'_a(s) = \frac{G'_p G'_c}{1 + G'_p G'_c} u_m(s)$$

$$\breve{x}(N\Delta_m) = x(N\Delta_m), \quad \forall N = 0, 1, 2, \ldots \tag{9}$$

where Δ_m is the MPC sampling time, G'_p is the estimated transfer function matrix of the control actuators, and G'_c is a well-tuned PID controller transfer function matrix based on the estimated model of the actuator G'_p. This makes $u'_a(t)$ the expected actuation level of $u_a(t)$.

Using Eqs. (8) and (9), the real-time measurements of $x(t)$ can be compared against the evolution of $\breve{x}(t)$. The residual, or the difference between $x_i(t)$ and $\breve{x}_i(t)$ denoted by $r_i(t)$, is expressed in the following manner:

$$r_i(t) = |\breve{x}_i(t) - x_i(t)| \tag{10}$$

In the absence of noise and if $G_p' = G_p$, whenever the jth element of u_a deviates from its expected behavior u_{aj}' and the ith-row–jth-column element of the $G(x)$ matrix is non-zero, the ith residual (r_i) would instantaneously become non-zero. In other words, r_i is non-zero only when there is a problem with the actuators that directly affect the ith state of the system (relative degree of 1) (Mhaskar et al., 2006, 2007).

In practice, however, model mismatch, process noise, and measurement noise are always present to some degree. Therefore, in a practical setting, the residuals will be non-zero regardless of the accuracy of the process model used in Eq. (9). Thus, before the model-based FDI method can be used in practice, the effects of process and measurement noise levels must first be recorded from fault-free closed-loop process operation data (with both the PID controllers and the MPC being well-tuned). On the basis of these noisy closed-loop system states, the mean and the standard deviation of the residuals are calculated and the thresholds are determined.

Occasional noise spikes can make the residuals exceed the thresholds for a brief period of time even when the actuators are functioning well; this can lead to the common problem of false alarms. To reduce the incidence of false alarms, we define a modified residual $r_{E,i}, i=1,\ldots,n_x$, for each residual r_i, calculated at discrete time instants t_k with $t_k = t_0 + k\Delta_r$, $k=0,1,2,\ldots$ and Δ_r being the interval between two consecutive state measurements. The weighted residual is calculated using an exponentially weighted moving average (EWMA) method as follows (Chilin et al., 2010, in press):

$$r_{E,i}(t_k) = \lambda r_i(t_k) + (1-\lambda) r_{E,i}(t_{k-1}) \tag{11}$$

with $r_{E,i}(t_0) = r_i(t_0)$ and the weighting factor $\lambda \in (0,1]$. The parameter λ determines the rate at which past data enters into the calculations of the weighted residual. When $\lambda = 1$, $r_{E,i}$ is equivalent to r_i. The typical range of λ is between 0.2 and 0.5 depending on the desired level of sensitivity (Chilin et al., in press; Lucas and Saccucci, 1990). Lower values of λ make the $r_E(t)$ curve smoother as potential noise spikes will have a smaller effect on the overall shape of the curve; i.e., instances of false alarm will be reduced. However, in the event where an actual poor tuning occurs, it may be detected and isolated more slowly.

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

$$\Omega_{E,i} = \mu_i + \alpha\sigma_i \sqrt{\frac{\lambda}{2-\lambda}} \tag{12}$$

where α is a threshold parameter determining how sensitive the FDI is; typical value of α is an integer value between 1 and 5. The parameters μ_i and σ_i are the mean and the standard deviation of the ith residual during normal operation. Once $r_{E,i}$ exceeds the threshold ($\Omega_{E,i}$) for a fixed amount of time t_d (determined by the user), then poor tuning is declared in the actuator(s) directly affecting the ith state and the retuning algorithm is activated. Fig. 2 shows the schematic of how the EWMA residuals are used to activate the PID retuning algorithm at the end of waiting time t_d.

Once a poorly tuned actuator is isolated, a PID tuning method can be applied to the PID controller based on the estimated transfer function of the actuator G_p'. To help ensuring the stability of the retuning algorithm, we employ a stability constraint. Specifically, whenever retuning is performed, the retuning algorithm makes sure that $G_p'G_c/(1+G_p'G_c)$ contains only strictly negative poles. In this work, we use Cohen–Coon and internal model control method to retune the PID parameters to demonstrate the approach. If desired, other model-based tuning rules may be used as well. See Skogestad (2003), Sung et al. (1998), Veronesi and Visioli (2009), and Zhuang and Atherton (1993) for other PID tuning methods.

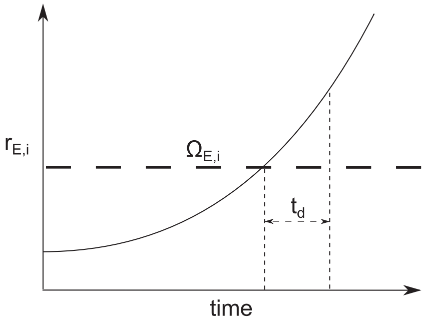

Fig. 2. Monitoring scheme of PID response behavior based on the EWMA residuals of the process state. Poor tuning is declared after $r_{E,i}$ exceeds its threshold $\Omega_{E,i}$ continuously for $t = t_d$.

Remark 3. One feature that should be noted is that the PID retuning will be initiated if the magnitude of the residuals is above a certain threshold. This means that even if the difference between $u_{aj}(t)$ and $u_{aj}'(t)$ is appreciable but the difference between $\tilde{x}_i(t)$ and $x_i(t)$ is smaller than the threshold, the retuner will do nothing. This is a direct result of the fact that the real value of $u_a(t)$ is unknown and has to be estimated from the trajectories of the process states. A scenario like this can also happen when $G_{ij}(\cdot)$ is small.

Remark 4. The isolability structure of the system is also critical to the use of the monitoring algorithm proposed here. If, from the patterns of the residuals, a poorly performing actuator cannot be isolated with high confidence (i.e., two actuators have the same signature because they directly affect the same system state), then all control actuators that may be poorly tuned should be retuned. In principle, it is also possible to use empirical models from input–output data in the MPC design as well as in the monitoring of the PID control loops. One potential problem of using this approach is the difficulty of isolating which specific PID control loop is poorly performing because input/output empirical models cannot account for the coupling between different process variables the way state-space first principles models do.

Remark 5. In the design of the filter of Eq. (9), a well-tuned PID controller, G_c', is assumed to be known and is used to calculate the benchmark performance of the overall control system. In the case that G_c' is not known, the control action computed by the MPC, u_m, can be used directly in the filter design (i.e., replace u_a' by u_m in Eq. (9)) to obtain an estimate of the expected process state evolution. Furthermore, once a poorly tuned actuator is isolated, retuning of the parameters of PID controller used in this actuator should be carried out to account for changes in operation conditions as well as control actuator wear and tear over time.

4. Application to a non-linear chemical process network

4.1. Process description and modeling

We demonstrate the PID monitoring and retuning methodology presented in the previous section using a three-vessel reactor–separator chemical process network. A schematic of the process is shown in Fig. 3. The first two vessels are assumed to be ideal CSTRs, followed by a flash tank separator. There are two fresh feed streams of pure reactant A of concentration C_{A10} to both reactors (with flow rates F_{10} and F_{20} respectively) and a recycle stream (F_r) from the flash tank to the first reactor. Specifically, the overhead vapor from the flash tank is condensed and recycled to

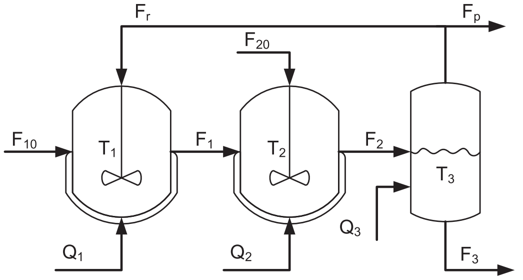

Fig. 3. Schematic of the process. Two CSTRs and a flash tank with recycle stream.

Table 1
Process parameter values.

$T_{10}=300$, $T_{20}=300$	K
$F_{10s}=5$, $F_{20s}=5$, $F_r=1.9$	m^3/h
$Q_{1s}=0$, $Q_{2s}=0$, $Q_{3s}=0$	kJ/h
$V_1=1.0$, $V_2=0.5$, $V_3=1.0$	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

the first CSTR, and the bottom product stream is removed. The effluent of vessel 1 is fed to vessel 2 and the effluent from vessel 2 is fed to the flash tank. Each vessel has an external heat input or heat removal system (Q_1, Q_2 and Q_3). The steady-state flow rate and heat input are denoted by F_{10s}, F_{20s}, Q_{1s},Q_{2s} and Q_{3s} and their values are given in Table 1. There are two parallel chemical reactions considered in this process; first, reactant A is converted to desired product B and second, A is converted to undesired product C (referred to as reactions 1 and 2, respectively). Under standard modeling assumptions, the dynamic energy and material balance equations that can describe this process take the following form:

$$\frac{dT_1}{dt}=\frac{F_{10}}{V_1}(T_{10}-T_1)+\frac{F_r}{V_1}(T_3-T_1)+\frac{Q_1}{\rho C_p V_1}+\frac{-(\Delta H_1)}{\rho C_p}k_1 e^{-E_1/RT_1}C_{A1}+\frac{(-\Delta H_2)}{\rho C_p}k_2 e^{-E_2/RT_1}C_{A1} \tag{13a}$$

$$\frac{dC_{A1}}{dt}=\frac{F_{10}}{V_1}(C_{A10}-C_{A1})+\frac{F_r}{V_1}(C_{Ar}-C_{A1})-k_1 e^{-E_1/RT_1}C_{A1}-k_2 e^{-E_2/RT_1}C_{A1} \tag{13b}$$

$$\frac{dC_{B1}}{dt}=\frac{-F_{10}}{V_1}C_{B1}+\frac{F_r}{V_1}(C_{Br}-C_{B1})+k_1 e^{-E_1/RT_1}C_{A1} \tag{13c}$$

$$\frac{dC_{C1}}{dt}=\frac{-F_{10}}{V_1}C_{C1}+\frac{F_r}{V_1}(C_{Cr}-C_{C1})+k_2 e^{-E_2/RT_1}C_{A1} \tag{13d}$$

$$\frac{dT_2}{dt}=\frac{F_1}{V_2}(T_1-T_2)+\frac{F_{20}}{V_2}(T_{20}-T_2)+\frac{Q_2}{\rho C_p V_2}+\frac{(-\Delta H_1)}{\rho C_p}k_1 e^{-E_1/RT_2}C_{A2}+\frac{(-\Delta H_2)}{\rho C_p}k_2 e^{-E_2/RT_2}C_{A2} \tag{13e}$$

$$\frac{dC_{A2}}{dt}=\frac{F_1}{V_2}(C_{A1}-C_{A2})+\frac{F_{20}}{V_2}(C_{A20}-C_{A2})-k_1 e^{-E_1 RT_2}C_{A2}-k_2 e^{-E_2/RT_2}C_{A2} \tag{13f}$$

$$\frac{dC_{B2}}{dt}=\frac{F_1}{V_2}(C_{B1}-C_{B2})-\frac{F_{20}}{V_2}C_{B2}+k_1 e^{-E_1/RT_2}C_{A2} \tag{13g}$$

$$\frac{dC_{C2}}{dt}=\frac{F_1}{V_2}(C_{C1}-C_{C2})-\frac{F_{20}}{V_2}C_{C2}+k_2 e^{-E_2/RT_2}C_{A2} \tag{13h}$$

$$\frac{dT_3}{dt}=\frac{F_2}{V_3}(T_2-T_3)-\frac{H_{vap}F_r}{\rho C_p V_3}+\frac{Q_3}{\rho C_p V_3} \tag{13i}$$

$$\frac{dC_{A3}}{dt}=\frac{F_2}{V_3}(C_{A2}-C_{A3})-\frac{F_r}{V_3}(C_{Ar}-C_{A3}) \tag{13j}$$

$$\frac{dC_{B3}}{dt}=\frac{F_2}{V_3}(C_{B2}-C_{B3})-\frac{F_r}{V_3}(C_{Br}-C_{B3}) \tag{13k}$$

$$\frac{dC_{C3}}{dt}=\frac{F_2}{V_3}(C_{C2}-C_{C3})-\frac{F_r}{V_3}(C_{Cr}-C_{C3}) \tag{13l}$$

where T_1, T_2, and T_3 are the temperatures of vessels 1, 2, and 3, respectively, T_{10} and T_{20} are the temperatures of the feed streams to vessels 1 and 2, respectively, F_{10} and F_{20} are the volumetric feed flow rates into vessels 1 and 2, respectively, and F_1 and F_2 are the volumetric flow rates out of vessels 1 and 2, respectively. F_r is the recycle stream volumetric flow rate from vessel 3 to vessel 1. V_1, V_2, and V_3 are the volumes of the three vessels, Q_1, Q_2, and Q_3 are the heat inputs into the vessels, C_{A1}, C_{B1}, C_{C1}, C_{A2}, C_{B2}, C_{C2}, C_{A3}, C_{B3}, and C_{C3} are the concentrations of A, B, and C in the vessels 1, 2, and 3, respectively, C_{Ar}, C_{Br}, and C_{Cr} are the concentrations of A, B, and C in the recycle stream. ρ is the mass density of the reacting fluid, C_p is the heat capacity of the reacting fluid, k_1 and k_2 are the pre-exponential reaction rate constants of reactions 1 and 2, respectively, E_1 and E_2 are the activation energies of reactions 1 and 2, respectively, ΔH_1 and ΔH_2 are the enthalpies of reactions 1 and 2, respectively, and H_{vap} is the heat of vaporization for the fluid in vessel 3. Finally, R is the universal gas constant.

The composition of the flash tank recycle stream is described by Eq. (14), which assumes constant relative volatility for each species within the temperature operating range. This assumption allows calculation of 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. Specifically, we have:

$$C_{Ar}=\frac{\alpha_A C_{A3}}{K} \tag{14a}$$

$$C_{Br}=\frac{\alpha_B C_{B3}}{K} \tag{14b}$$

$$C_{Cr}=\frac{\alpha_C C_{C3}}{K} \tag{14c}$$

$$K=\alpha_A C_{A3}\frac{MW_A}{\rho}+\alpha_B C_{B3}\frac{MW_B}{\rho}+\alpha_C C_{C3}\frac{MW_C}{\rho}+\alpha_D x_D \tag{14d}$$

where α_A, α_B, α_C, and α_D are the relative volatility constants of the three reacting species along with the inert species D. MW_A, MW_B, and MW_C, are the molecular weights of the three reacting species. Finally, x_D is the mass fraction of the inert species D in the liquid phase of vessel 3. The values of the process parameters are given in Table 1.

The system of Eq. (13) is solved numerically using explicit Euler method with a time step of $\Delta_p=0.001$ h. Process and sensor measurement noise are also used in the process simulation. The sensor measurement noise is generated using a zero-mean normal distribution with a standard deviation of 2.5 K for the three temperature state measurements and 1 kmol/m^3 for the nine concentration state measurements. The process noise is generated similarly and it is included as an additive term in the right-hand-side of the ordinary differential equations of Eq. (13) with a zero-mean normal distribution and the same standard deviation values used for the measurement noise. In all three vessels, the heat inputs are used as the manipulated variables for controlling the process network at the operating steady-state. Therefore, the corresponding relative degrees of these variables with respect to the temperatures of the three vessels (reactor 1, reactor 2 and separator) are all one, thereby allowing isolation of poor-tuning in each one of these actuators from process measurements. In addition the second tank's inlet flow rate is chosen as another manipulated variable. The system has one unstable and two stable steady states. The operating steady-state is the *unstable steady-state* shown in Table 2.

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

$$G_{actuator} = \frac{e^{-3.60s}}{2.82s+1} \tag{15}$$

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

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

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

For the actuators with the transfer function presented in Eq. (15), the PID parameters that give the best closed-loop response were found to be the following:

$$K_c^* = 0.648$$

$$\tau_I^* = 5.94 \text{ s}$$

$$\tau_D^* = 0.54 \text{ s} \tag{16}$$

These parameters were used to calculate G'_c. The poles of $G'_p G'_c/(1+G'_p G'_c)$ calculated with the parameters above are found to be all negative. This, in conjunction with the approximate transfer function (G'_p) of the actuators of Eq. (15), was then used to approximate the ideal actuation performance ($u'_a(s)$) of each control actuator.

Table 2
Operating steady-state (x_s).

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

4.2. Simulation results

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

4.2.1. Example 1

In this example, we start the process from the following initial condition: $x(0) = 0.8x_s$ where x_s is the operating steady-state. All the control actuators are properly tuned with the PID parameters shown in Eq. (16). At time $t = 0.45$ h, we apply poor tuning to the PID controller for the actuator Q_1 with the following parameters:

$$K_c = 0.00909$$

$$\tau_I = 11.9 \text{ s}$$

$$\tau_D = 0.655 \text{ s} \tag{17}$$

Fig. 4 shows the comparison between the requested actuation level $u_m(t)$ and the actual actuation level $u_a(t)$ for Q_1 if the monitoring and retuning system is inactive. The EWMA residuals of the temperature in three vessels are shown in Fig. 5.

With the monitoring system active, Fig. 6 shows the evolution of PID response $u_a(t)$ as it is retuned at $t = 0.475$ h. As shown in Fig. 7, at $t = 0.465$ h, r_{E,T_1} starts exceeding its threshold Ω_{E,T_1}. At this point, the value of r_{E,T_1} starts being monitored closely for $t_d = 0.01$ h. By the time the system reaches $t = 0.475$ h, the value of r_{E,T_1} is found to have been above its threshold Ω_{E,T_1} for the entire duration from $t = 0.465$ h to $t = 0.475$ h. Because the process state T_1 is the only state that is directly affected by the control actuator Q_1, given the model-based FDI filter design, any anomaly detected in r_{E,T_1} is the result of a problem with the Q_1 control actuator. Therefore, the actuator Q_1 can be isolated with high confidence as the actuator with poor PID tuning. While other residuals (r_{E,T_2} and r_{E,T_3}) occasionally exceed their thresholds at various time instances during the operation, they do not exceed the thresholds for longer than $t_d = 0.01$ h. Thus, the monitoring system concludes that their values exceed their thresholds simply because of process and measurement noise.

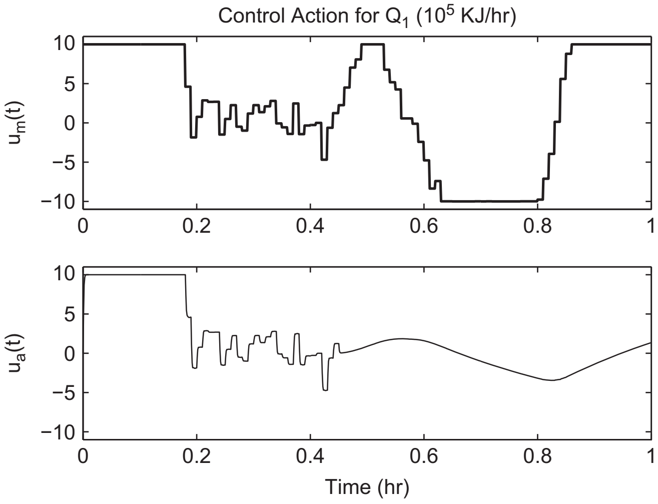

Fig. 4. Example 1: requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is not implemented.

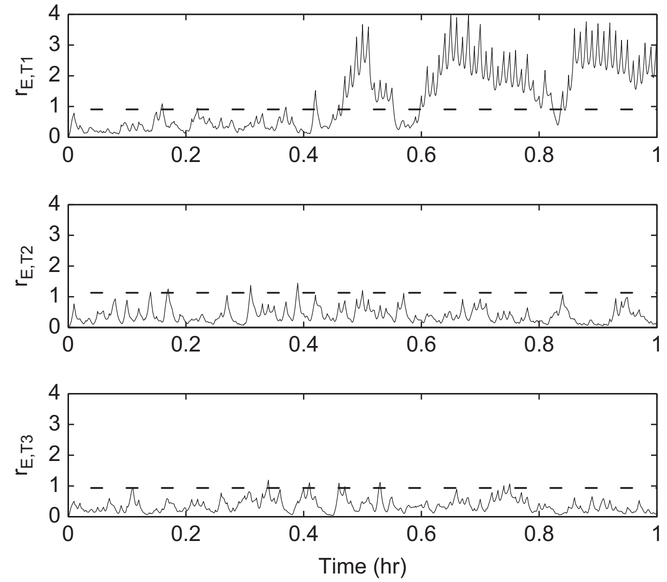

Fig. 5. Example 1: temperature residuals for the three vessels computed via EWMA when PID retuning is not implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$.

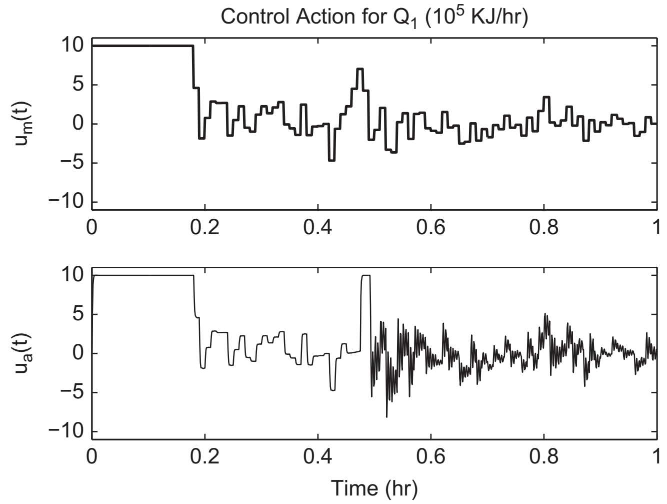

Fig. 6. Example 1: requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is implemented.

Once the Q_1 control actuator is isolated as the poorly tuned actuator, Cohen–Coon tuning method is applied to the controller around Q_1 based on the estimated transfer function of the control actuator G_p'. The Cohen–Coon tuning rule is based on the first-order-plus-dead-time estimation of the transfer function of the controlled process. Specifically, the Cohen–Coon tuning rule is as follows (Cohen and Coon, 1953):

$$K_c = \frac{\tau_p}{K_p \tau_d}\left(\frac{4}{3} + \frac{\tau_d}{4\tau_p}\right) \tag{18a}$$

$$\tau_I = \tau_d \frac{32 + 6\dfrac{\tau_d}{\tau_p}}{13 + 8\dfrac{\tau_d}{\tau_p}} \tag{18b}$$

$$\tau_D = \tau_d \frac{4}{11 + 2\dfrac{\tau_d}{\tau_p}} \tag{18c}$$

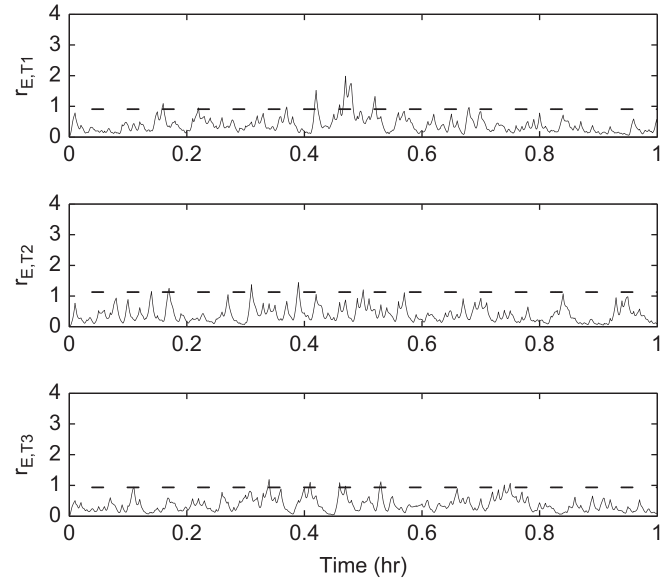

Fig. 7. Example 1: temperature residuals for the three vessels computed via EWMA when PID retuning is implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$.

where K_p is the actuator's gain, τ_d is the actuator dead time, and τ_p is the actuator's time constant. With this tuning rule and the estimated transfer function of the actuator G_p' presented in Eq. (15), the resulting parameters for the PID of Q_1 are as follows:

$$K_c = 1.29$$
$$\tau_I = 6.15 \text{ s}$$
$$\tau_D = 1.06 \text{ s} \tag{19}$$

After Q_1 is retuned, no more problem can be detected from the EWMA residuals of T_1. In terms of the actual control actuator performance, after being retuned with Cohen–Coon method, $u_a(t)$ tracks $u_m(t)$ quite well; see Fig. 6.

4.2.2. Example 2

In this example, we will use internal model control tuning rule (Skogestad, 2003) to tune the PID parameters. We initialize the process model from the following initial condition: $x(0) = 0.8x_s$ where x_s is the operating steady-state. All PID controllers start out being properly tuned with the parameters presented in Eq. (16). At time $t = 0.1$ h, a poor PID tuning with the following parameters:

$$K_c = 6.48$$
$$\tau_I = 0.594 \text{ s}$$
$$\tau_D = 5.40 \text{ s} \tag{20}$$

is applied to the PID controller for the control actuator Q_3. Fig. 8 shows that the tuning of the PID controller for Q_3 causes $u_a(t)$ to oscillate significantly. Fig. 9 shows the EWMA residuals of the temperature of the three vessels when PID retuning is not implemented.

With the monitoring system implemented, Fig. 11 shows that r_{E,T_3} is found to start exceeding its threshold Ω_{E,T_3} at $t = 0.206$ h. After waiting for $t_d = 0.01$ h, r_{E,T_3} is found to have been continuously above its threshold until $t = 0.216$ h. Because Q_3 is the only actuator that has relative degree 1 with the process state T_3, at $t = 0.216$ h the monitoring system isolates Q_3 and declares that Q_3 is poorly tuned. As a result, at $t = 0.216$ h, a set of PID parameters is calculated via internal model control tuning method based on

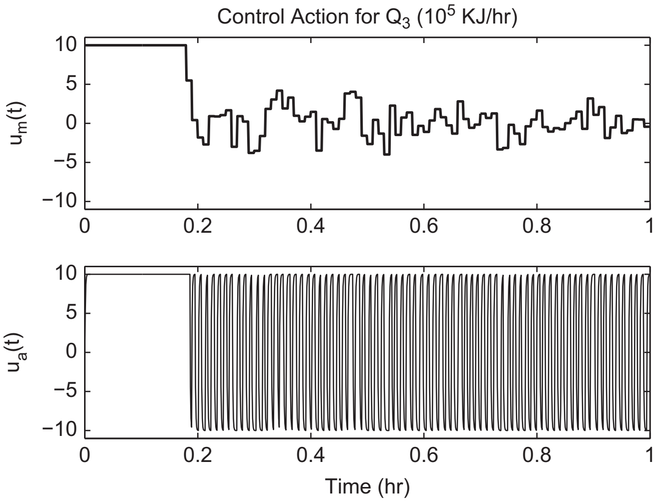

Fig. 8. Example 2: requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is not implemented.

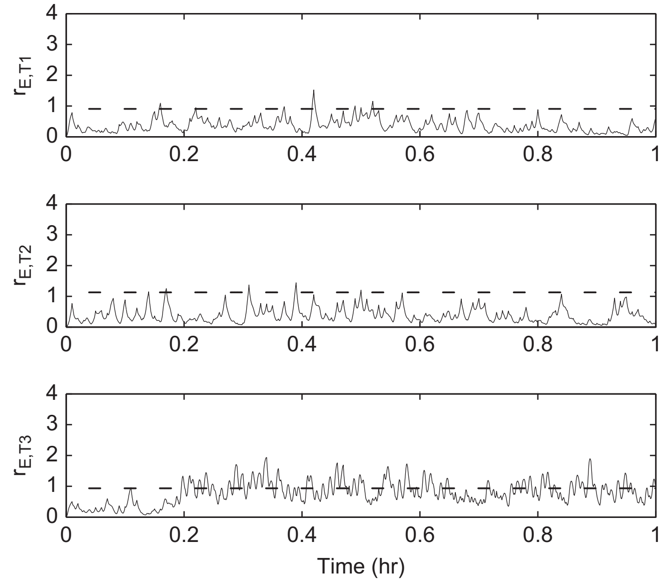

Fig. 9. Example 2: temperature residuals for the three vessels computed via EWMA when PID retuning is not implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$.

the estimated transfer function of the control actuator G_p'. For a tuning with fast PID step response, internal model control tuning rule suggests the following PID parameters for processes that can be approximated with first-order-plus-dead-time transfer function (Skogestad, 2003):

$$K_c = \frac{\tau_p}{2K_p\tau_d} \tag{21a}$$

$$\tau_I = \min(\tau_p, 8\tau_d) \tag{21b}$$

$$\tau_D = 0 \tag{21c}$$

where K_p is the actuator's gain, τ_d is the actuator dead time, and τ_p is the actuator's time constant. This results in the following PID parameters:

$$K_c = 0.392$$

$$\tau_I = 2.82 \text{ s}$$

$$\tau_D = 0 \text{ s} \tag{22}$$

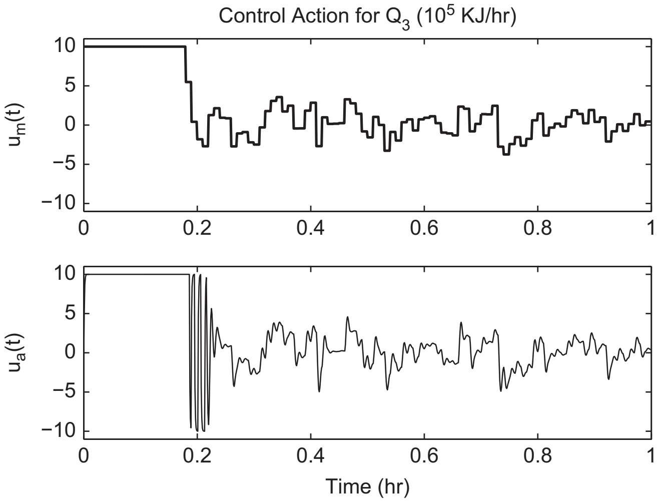

Fig. 10. Example 2: requested actuation level by the MPC ($u_m(t)$) and actual actuation level ($u_a(t)$) when PID retuning is implemented.

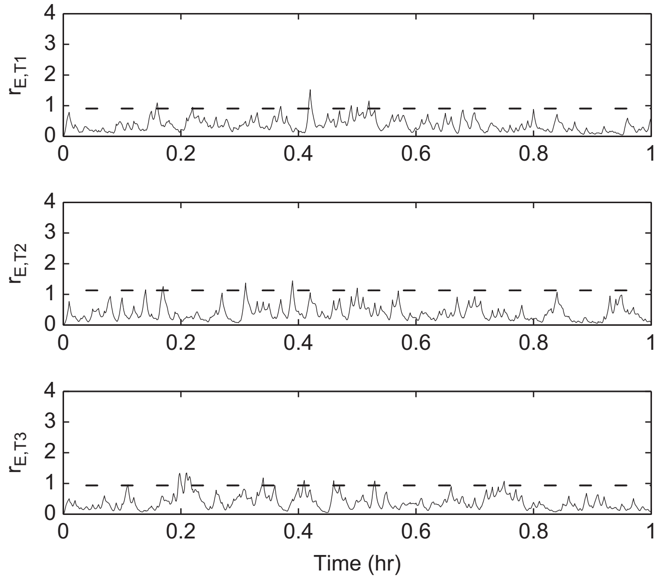

Fig. 11. Example 2: temperature residuals for the three vessels computed via EWMA when PID retuning is implemented. The dashed lines represent the EWMA residual thresholds $\Omega_{E,i}$.

Fig. 10 shows the resulting actual actuation level ($u_a(t)$) of Q_3. Though poor PID tuning is applied at $t = 0.1$ h, its effect in terms of PID response of the control actuator is observed at $t = 0.185$ h when the step change happens. In terms of detecting this oscillation pattern from the process sate measurements, this is detected and isolated at $t = 0.216$ h and the PID parameters of Q_3 are retuned.

Notice in Fig. 9 that the magnitude of the residuals of the directly affected process state (r_{E,T_3} in this case) is much lower than r_{E,T_1} in Example 1 (shown in Fig. 5). This is because the poor PID tuning problem in this example results in an actuator oscillation ($u_a(t)$) that oscillates with very high frequency around the set-point ($u_m(t)$). In terms of the process states, this leads to a smaller overall deviation of the actual process state ($x(t)$) from the expected process state ($\check{x}(t)$). This is why there is a slightly larger time lag between the initial time when $u_a(t)$ starts deviating from $u_m(t)$ and the time when the poor tuning is isolated, compared to Example 1.

Remark 6. While the mean and standard deviation of the residuals are calculated in the presence of process noise under normal operation at the desired steady-state, the applicability of the proposed dynamic filter for computing the residuals together with real-time state variable measurements is not limited to steady-state operation; the reason is the design of the proposed dynamic filter which can accurately predict normal evolution of the process state variables away from the steady-state in the closed-loop system, thereby leading to the computation of residual values that are valid for process operation away from the steady-state (note that the initial condition in the example is not chosen to be the steady-state).

5. Conclusion

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

References

Anderson, K., Blankenship, G., Lebow, L., 1988. A rule-based adaptive PID controller. In: Proceedings of IEEE Conference on Decision and Control, Austin, Texas, pp. 564–569.

Astrom, K., Hagglund, T., Hang, C., Ho, W., 1993. Automatic tuning and adaptation for PID controllers—a survey. Control Eng. Prac. 1, 699–714.

Chand, S., 1992. Self-Monitoring Tuner for Feedback Controller.

Chilin, D., Liu, J., Davis, J.F., Christofides, P.D. Data-based monitoring and reconfiguration of a distributed model predictive control system. Int. J. Robust Nonlinear Control, doi:10.1002/rnc.1759. In press.

Chilin, D., Liu, J., Muñoz de la Peña, D., Christofides, P.D., Davis, J.F., 2010. Detection, isolation and handling of actuator faults in distributed model predictive control systems. J. Process Control 20, 1059–1075.

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

Cohen, G.H., Coon, G.A., 1953. Theoretical consideration of retarded control. ASME 75, 827–834.

Eriksson, P.-G., Isaksson, A.J., 1994. Some aspects of control loop performance monitoring. Control Appl. 2, 1029–1034.

Hang, C., Astrom, K., Ho, W., 1991. Refinements of Zeigler–Nichols tuning formula. IEE Proc.—D 138, 111–118.

Harris, T.J., 1989. Assessment of control loop performance. Can. J. Chem. Eng. 67, 856–861.

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

Liu, J., Chen, X., Muñoz de la Peña, D., Christofides, P.D., 2010. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. AIChE J. 56, 2137–2149.

Lucas, J., Saccucci, M., 1990. Exponentially weighted moving average control schemes. Technometrics 32, 1–12.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained model predictive control: stability and optimality. Automatica 36, 789–814.

Mhaskar, P., Gani, A., El-Farra, N.H., McFall, C., Christofides, P.D., Davis, J.F., 2006. Integrated fault-detection and fault-tolerant control of process systems. AIChE J. 52, 2129–2148.

Mhaskar, P., El-Farra, N.H., Christofides, P.D., 2006. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. Syst. Control Lett. 55, 650–659.

Mhaskar, P., McFall, C., Gani, A., Christofides, P.D., Davis, J.F., 2007. Isolation and handling of actuator faults in nonlinear systems. Automatica 44, 53–62.

Nishikawa, Y., Sannomiya, N., Ohta, T., Tanaka, H., 1984. A method for auto-tuning of PID control parameters. Automatica 20, 321–332.

Qin, S.J., 1998. Control performance monitoring—a review and assessment. Comput. Chem. Eng. 23, 173–186.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11, 733–764.

Rugh, W.J., 1991. Analytical framework for gain scheduling. IEEE Control Syst. 11, 79–84.

Saito, T., 1990. PID Controller System.

Shi, D., Tsung, F., 2003. Modeling and diagnosis of feedback-controlled process using dynamic PCA and neural networks. Int. J. Prod. Res. 41, 365–379.

Skogestad, S., 2003. Simple analytic rules for model reduction and PID controller tuning. J. Process Control 13, 291–309.

Sung, S.-W., Lee, I.-B., Lee, B.-K., 1998. On-line process identification and automatic tuning method for PID controllers. Chem. Eng. Sci. 53, 1847–1859.

Teng, F., Lotfi, A., Tsoi, A., 2008. Novel fuzzy logic controllers with self-tuning capability. J. Comput. 3, 9–16.

Tsung, F., 2000. Statistical monitoring and diagnosis of automatic control processes using dynamic PCA. Int. J. Prod. Res. 38, 625–637.

Tsung, F., Shi, J., 1999. Integrated design of run-to-run PID controller and SPC monitoring for process disturbance rejection. IIE Trans. 31, 517–527.

Tsung, F., Shi, J., Wu, C., 1999. Joint monitoring of PID-controlled process. J. Qual. Technol. 31, 275–285.

Veronesi, M., Visioli, A., 2009. Performance assessment and retuning of PID controllers. Ind. Eng. Chem. Res. 48, 2616–2623.

Wang, Q., Zhang, Z., Chek, L., Astrom, K., 2009. Guaranteed dominant pole placement with PID controllers. J. Process Control 19, 349–352.

Zayed, A., El-Fallah, A., El-Fandi, M., Hussain, A., 2009. A novel implicit adaptive pole-placement PID controller. In: Proceedings of the IASTED International Conference on Modeling and Simulation, vol. 12, Banff, Canada, pp. 296–300.

Zhao, Z.-Y., Tomizuka, M., Isaka, S., 1993. Fuzzy gain scheduling of PID controllers. IEEE Trans. Syst. Man Cybern. 23, 1392–1398.

Zhuang, M., Atherton, D., 1993. Automatic tuning of optimum PID controllers. IEE Proc.—D 140, 216–224.

Ziegler, J.G., Nichols, N., 1942. Optimum settings for automatic controllers. ASME 64, 759–768.