ARTICLE IN PRESS

ISA Transactions ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

ISA Transactions

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

Research Article

A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes

Aydogan Savran*, Gokalp Kahraman[1]

Department of Electrical and Electronics Engineering, Ege University, 35100 Bornova, Izmir, Turkey

A R T I C L E I N F O

Article history:
Received 19 September 2012
Received in revised form
16 September 2013
Accepted 26 September 2013
This paper was recommended for publication by Prof. A.B. Rad

Keywords:
Adaptive
PID
Fuzzy
Prediction
Soft limiter
LM
BPTT
Control

A B S T R A C T

We develop a novel adaptive tuning method for classical proportional–integral–derivative (PID) controller to control nonlinear processes to adjust PID gains, a problem which is very difficult to overcome in the classical PID controllers. By incorporating classical PID control, which is well-known in industry, to the control of nonlinear processes, we introduce a method which can readily be used by the industry. In this method, controller design does not require a first principal model of the process which is usually very difficult to obtain. Instead, it depends on a fuzzy process model which is constructed from the measured input–output data of the process. A soft limiter is used to impose industrial limits on the control input. The performance of the system is successfully tested on the bioreactor, a highly nonlinear process involving instabilities. Several tests showed the method's success in tracking, robustness to noise, and adaptation properties. We as well compared our system's performance to those of a plant with altered parameters with measurement noise, and obtained less ringing and better tracking. To conclude, we present a novel adaptive control method that is built upon the well-known PID architecture that successfully controls highly nonlinear industrial processes, even under conditions such as strong parameter variations, noise, and instabilities.

1. Introduction

PID controllers are well-known and popular in industry due to their simplicity and robustness, but most of them lack well-tuning and accuracy, that motivated much work to develop tuning and adaptation techniques to cope with nonlinear industrial processes [1–8]. Adaptive tuning becomes necessary for parameter variations, which fixed parameter controllers cannot sufficiently compensate. Recently, predictive control technique has been widely appeared in the process control [9–11]. In the predictive control, a process model is required to predict the future effects of the control action at the present. First principles-based nonlinear models are difficult to develop for many industrial cases. Nonlinear process can be alternatively modeled with fuzzy systems. Fuzzy logic was introduced by Zadeh [12,13] and then used for many control and modeling applications [14,15]. Takagi–Sugeno [16] and Mamdani [17] fuzzy models are the two main types of fuzzy systems. The Takagi–Sugeno fuzzy model (Takagi–Sugeno modeling methodology) includes fuzzy sets only in the premise part and a regression model as the consequent [16] while the Mamdani fuzzy model includes fuzzy sets in both parts. Many successful applications of the predictive control using fuzzy models have been reported in the literature [18–21].

Although, the PID tuning and adaptation techniques have extensively developed for linear time-invariant systems [2,3,22–26], still much effort should be spent for nonlinear as well as time-variant systems. This has motivated this study. In this paper, a novel adaptive tuning procedure for classical PID controllers is developed and efficiently applied for a nonlinear process control problem. One of the main features of the proposed approach is that the PID controller architecture is solely classic making it readily usable in nonlinear industrial processes, avoiding lengthy and less familiar fuzzy systems. Our classic PID design allows computing the parameters from the measured input–output data of the nonlinear process more precisely than other known designs by virtue of its multi-step forward prediction, on-line predictor training, adaptive controller tuning, and fast training features. By this way, a predictive adjustment procedure is performed based on the predictions of a fuzzy model of the process. Adaptation part involves a classical PID controller driving a fuzzy predictor. The fuzzy predictor output is used to adjust the gains by minimizing the error between the predictions and the reference. Additionally, the fuzzy predictor is trained on-line to adapt to the variations in plant parameters, and thus improve the prediction accuracy. The actual plant is controlled by a PID controller identical to that of the

* Corresponding author. Tel.: +90 232 3111664; fax: +90 232 3886024.
E-mail addresses: aydogan.savran@ege.edu.tr (A. Savran), gokalp.kahraman@ege.edu.tr (G. Kahraman).
[1] Tel.: +90 232 3111669; fax: +90 232 3886024.

0019-0578/$ - see front matter © 2013 Published by Elsevier Ltd. on behalf of ISA.
http://dx.doi.org/10.1016/j.isatra.2013.09.020

adaptation part, the gains of which are transferred at each control cycle. A soft limiter is used to impose industrial limits on the control input.

In a concurrent work [27], we have used a fuzzy system to implement the controller that resulted in procedurally lengthy and computationally heavy algorithms that are unfavorable in industrial users. In contrast, in this work simple PID gain parameters replace lengthy fuzzy-system computations. The predictor is constructed only from the measurement of the input–output data of the actual process as a fuzzy model, without requiring a first-principal model of the process which is usually very difficult to obtain.

Section 2 provides background information on the structure of the fuzzy systems and the PID controller, used in the paper, whereas Section 3 explains the control system architecture and the training procedure. Section 4 discusses the results of the performance tests done on the bioreactor, a highly nonlinear process involving instabilities. Several tests and comparative study showed the method's success in tracking, robustness to noise, and adaptation properties. Section 5 is the conclusion, summarizing the performance of our method.

2. The background

2.1. The PID controller

Proportional–integral–derivative (PID) controllers, which have relatively simple structures and robust performances, are the most common controllers in industry. By taking the time-derivative of the both sides of the continuous-time PID equation, and discretizing the resulting equation, one easily gets the PID equation in the incremental form as below:

$$u(k)=u(k-1)+K_P(e(k)-e(k-1))+\frac{K_I T}{2}(e(k)+e(k-1)) + \frac{K_D}{T}(e(k)-2e(k-1)+e(k-2)) \tag{1}$$

where K_P is the proportional gain, K_I is the integral gain, and K_D is the derivative gain. T is the sampling period, u is the output of the PID controller, and k is the discrete time index. The difference between the reference input (r) and the actual plant output (y) is the error term, $e=r-y$. Equation can be simplified to

$$\Delta u(k)=K_P e_P(k)+K_I e_I(k)+K_D e_D(k)$$
$$u(k)=u(k-1)+\Delta u(k) \tag{2}$$

where

$$e_P(k)=e(k)-e(k-1)$$
$$e_I(k)=\frac{T}{2}(e(k)+e(k-1))$$
$$e_D(k)=\frac{1}{T}(e(k)-2e(k-1)+e(k-2)) \tag{3}$$

with $e(k)=0$ for $k<0$.

In order to provide the control signal in physical limits, a soft limiter is placed after the controller. The control signal at the output of the soft limiter becomes

$$u_l(k)=h(u(k)) \tag{4}$$

where $h(u)=1/1+e^{-u}$ is a sigmoid type function.

2.2. The fuzzy system

The fuzzy system (FS), $f(\mathbf{x};\boldsymbol{\theta})$, used in this study consists of the product inference engine, the singleton fuzzifier, the center average defuzzifier, and the Gaussian membership functions [28,29]:

$$f(\mathbf{x}(k);\boldsymbol{\theta})=\frac{\sum_{j=1}^{M} b_j \prod_{i=1}^{N}\exp(-\frac{1}{2}(x_i(k)-c_{ij}/\sigma_{ij})^2)}{\sum_{j=1}^{M}\prod_{i=1}^{N}\exp(-\frac{1}{2}(x_i(k)-c_{ij}/\sigma_{ij})^2)} \tag{5}$$

where x_i is the ith input of the FS, N is the number of inputs, M is the number of membership functions assigned to each input, c_{ij} and σ_{ij} are, respectively, the center and the spread of the jth membership function corresponding to the ith input. Any output membership function which has a fixed spread of 1 is characterized only by the center parameter b_j. The input vector $\mathbf{x}(k)$ represents all of the inputs of the FS at time k. The parameter vector $\boldsymbol{\theta}=[\mathbf{b},\mathbf{c},\boldsymbol{\sigma}]$ contains all of the fuzzy-set parameters. The derivatives $(\partial f(\mathbf{x}(k);\boldsymbol{\theta})/\partial\boldsymbol{\theta})$ for the FS output with respect to its parameters are computed as

$$\frac{\partial f}{\partial b_j}=\frac{\partial f}{\partial \alpha}\frac{\partial \alpha}{\partial b_j}=\frac{1}{\phi}z_j$$
$$\frac{\partial f}{\partial c_{ij}}=\frac{\partial f}{\partial z_j}\frac{\partial z_j}{\partial c_{ij}}=\frac{(b_j-f)}{\phi}z_j\frac{(x_i-c_{ij})}{(\sigma_{ij})^2}$$
$$\frac{\partial f}{\partial \sigma_{ij}}=\frac{\partial f}{\partial z_j}\frac{\partial z_j}{\partial \sigma_{ij}}=\frac{(b_j-f)}{\phi}z_j\frac{(x_i-c_{ij})^2}{(\sigma_{ij})^3} \tag{6}$$

where

$$z_j=\prod_{i=1}^{N}\exp(-\tfrac{1}{2}(x_i-c_{ij}/\sigma_{ij})^2)$$
$$\alpha=\sum_{j=1}^{M} b_j z_j$$
$$\varphi=\sum_{j=1}^{M} z_j$$
$$f=\frac{\alpha}{\varphi} \tag{7}$$

The input–output sensitivities of the FS are computed as

$$\frac{\partial f(\mathbf{x})}{\partial x_i}=\sum_{j=1}^{M}\frac{f(\mathbf{x})-b_j}{\varphi}z_j\left(\frac{x_i-c_{ij}}{\sigma_{ij}^2}\right) \tag{8}$$

3. The adaptive PID control system

3.1. Control system architecture

Fig. 1 depicts the adaptive PID control system architecture. The adaptation part includes the PID controller and the fuzzy predictor. The PID controller computes the control actions. The predictor is constructed only from the measurement of the input–output data of the actual process as a fuzzy model, without requiring a first-principal model of the process which is usually very difficult to obtain. The multi-step ahead predictions of the process output are provided by the fuzzy predictor. The fuzzy predictor output is used to adjust the gains by minimizing the sum of the squared errors between the predictions and the reference over the prediction horizon. The fuzzy predictor is trained on-line to adapt to the variations in plant parameters, and thus improve the prediction accuracy. A certain history of the actual plant inputs and outputs is stored in a first-in first-out stack, providing the online training data at each training step [30]. The PID controller gains and the predictor fuzzy system parameters are both adjusted by the Levenberg–Marquardt (LM) method [31]. The actual plant is controlled by a PID controller identical to that of the adaptation phase, the gains of which are transferred at each control cycle. A sigmoid-type soft limiter is used to impose industrial limits on the control input.

3.2. Adaptation of the PID controller gains

The fuzzy predictor parameters and the controller gains are adjusted at each control cycle. The cost function used to adjust the controller gains is defined as the sum of the squared error

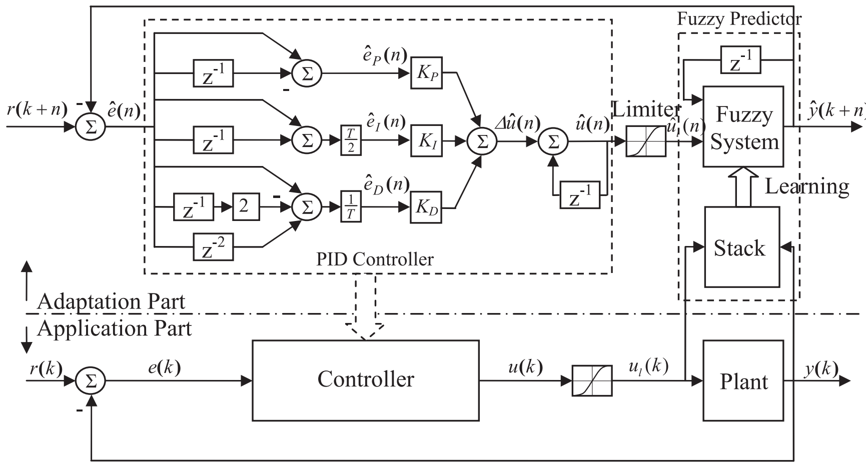

Fig. 1. Proposed PID control system architecture.

between the set point and the predictor output over the prediction horizon H as below:

$$E_c^H = \frac{1}{2H}\sum_{n=0}^{H-1}(r(k+n)-\hat{y}(k+n))^2 \tag{9}$$

where k is the initial edge of the prediction horizon. The Jacobian with respect to the controller gains $\boldsymbol{\theta}^c = [K_p, K_I, K_D]$ is computed as

$$J_c = \frac{\partial \hat{e}}{\partial \boldsymbol{\theta}^c} = \frac{\partial \hat{e}}{\partial f_p}\frac{\partial f_p}{\partial \hat{u}_l}\frac{\partial \hat{u}_l}{\partial \hat{u}}\frac{\partial \hat{u}}{\partial \boldsymbol{\theta}^c} \tag{10}$$

One should notice that the delay elements at the outputs of PID controller and the fuzzy predictor form recurrent systems, and as such their Jacobians (or equivalently gradients) are computed by the back-propagation through time (BPTT) method which is based on the adjoint model.

The Jacobians ($J_c = \partial\hat{e}/\partial\boldsymbol{\theta}^c$) are computed in two phases referred as the forward and the backward. In the forward phase, at every time step k, the controller and predictor sequentially run and their outputs are stored from $n=0$ to $n=H-1$, where n counts the ahead of time steps and H is the prediction horizon. The ahead-of-time value of control signal, denoted as $\hat{u}(n)$, drives the predictor, and it is given by

$$\hat{u}(n) = \hat{u}(n-1) + \Delta\hat{u}(n), \quad \text{for } n = 0,\ 1,\ \ldots,\ H-1 \tag{11}$$

where

$$\Delta\hat{u}(n) = K_P\hat{e}_P(n) + K_I\hat{e}_I(n) + K_D\hat{e}_D(n) \tag{12}$$

$$\hat{e}_p(n) = \hat{e}(n) - \hat{e}(n-1)$$
$$\hat{e}_I(n) = \frac{T}{2}(\hat{e}(n) + \hat{e}(n-1))$$
$$\hat{e}_D(n) = \frac{1}{T}(\hat{e}(n) - 2\hat{e}(n-1) + \hat{e}(n-2)) \tag{13}$$

The difference between the reference and the predictor output, $\hat{e}(n) = r(k+n) - \hat{y}(k+n)$ is the ahead-of-time error. The initial conditions of the equations from (11) to (12) are $\hat{u}(n=-1) = u(k-1)$ $\hat{e}(n=0) = e(k-1)$, $\hat{e}(n=-1) = e(k-2)$, and $\hat{e}(n=-2) = e(k-3)$.

The control action at the output of the limiter is obtained as

$$\hat{u}_l(n) = h(\hat{u}(n)), \tag{14}$$

where limiter h is a sigmoid type function. The predictor input vector contains the control action and the previous predictor output:

$$\hat{\mathbf{x}}^p(n) = [\hat{y}(k+n-1), \hat{u}_l(n)] \tag{15}$$

The predicted plant output is the predictor output:

$$\hat{y}(k+n) = f_p(\hat{\mathbf{x}}^p(n); \boldsymbol{\theta}^p) \tag{16}$$

Table 1
The forward phase computations.

$\boldsymbol{\theta}^c = [K_p, K_I, K_D]$
$\boldsymbol{\theta}^p = [\mathbf{b}^p, \mathbf{c}^p, \boldsymbol{\sigma}^p]$
$\hat{e}(0) = r(k-1) - y(k-1)$
$n = 0,\ \ldots,\ H-1$
$\hat{e}_p(n) = \hat{e}(n) - \hat{e}(n-1)$
$\hat{e}_I(n) = \frac{T}{2}(\hat{e}(n) + \hat{e}(n-1))$
$\hat{e}_D(n) = \frac{1}{T}(\hat{e}(n) - 2\hat{e}(n-1) + \hat{e}(n-2))$
$\Delta\hat{u}(n) = K_P\hat{e}_P(n) + K_I\hat{e}_I(n) + K_D\hat{e}_D(n)$
$\hat{u}(n) = \hat{u}(n-1) + \Delta\hat{u}(n)$
$\hat{u}_l(n) = h(\hat{u}(n))$
$\hat{\mathbf{x}}^p(n) = [\hat{u}_l(n), \hat{y}(k+n-1)]$
$\hat{y}(k+n) = f_p(\hat{\mathbf{x}}^p(n); \boldsymbol{\theta}^p)$
$\hat{e}(n+1) = r(k+n) - \hat{y}(k+n)$
$E_c^H = \frac{1}{2H}\sum_{n=0}^{H-1}\hat{e}^2(n+1)$

where the parameter vector $\boldsymbol{\theta}^p = [\mathbf{b}^p, \mathbf{c}^p, \boldsymbol{\sigma}^p]$ contains all of the fuzzy-set parameters of the predictor.

The mean square error through the prediction horizon is computed as

$$E_c^H = \frac{1}{2H}\sum_{n=0}^{H-1}\hat{e}^2(n+1) \tag{17}$$

Table 1 summarizes the forward phase computations.

At the completion of the forward phase at each k, the backward phase computations from $n=H-1$ to $n=0$ are performed by means of the adjoint model shown in Fig. 2. In the adjoint model, the arrow directions of the signal propagation are reversed, the summing junctions and the branching points are interchanged, and the advance operators replace the delay operators.

The backward action at the predictor output is obtained as

$$\delta_p(n) = [-1 + v(n+1)], \quad \text{for } n = H-1,\ \ldots,\ 0. \tag{18}$$

with $v(n=H) = 0$. The $v(n)$ is computed via $\delta_p(n)$:

$$v(n) = \left.\frac{\partial f_p(\mathbf{x}^p; \boldsymbol{\theta}^p)}{\partial x_1^p}\right|_{\hat{\mathbf{x}}^p(n)} \delta_p(n) \tag{19}$$

The backward action at the limiter input is obtained as

$$\delta_l(n) = [h(\hat{u}(n))]\left.\frac{\partial f_p(\mathbf{x}^p; \boldsymbol{\theta}^p)}{\partial x_2^p}\right|_{\hat{\mathbf{x}}^p(n)} \delta_p(n) \tag{20}$$

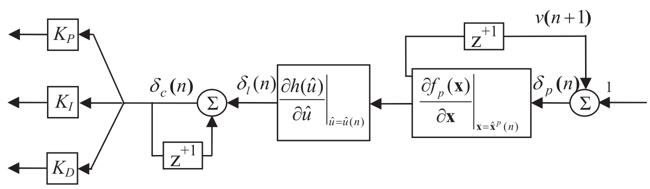

Fig. 2. The adjoint model of the adaptation part.

Table 2
The backward phase computations.

$$n = H-1, \ldots, 0$$
$$\delta_p(n) = [-1 + v(n+1)] \text{ with } v(n=H) = 0$$
$$v(n) = \frac{\partial f_p(\mathbf{x}^p; \boldsymbol{\theta}^p)}{\partial x_1^p}\Big|_{\hat{\mathbf{x}}^p(n)} \delta_p(n)$$
$$h'(\hat{u}(n)) = \frac{\partial h}{\partial u}\Big|_{u(n)\,=\,\hat{u}(n)}$$
$$\delta_l(n) = [h'(\hat{u}(n))] \frac{\partial f_p(\mathbf{x}^p; \boldsymbol{\theta}^p)}{\partial x_2^p}\Big|_{\hat{\mathbf{x}}^p(n)} \delta_p(n)$$
$$\delta_c(n) = \delta_l(n) + \delta_c(n+1) \text{ with } \delta_c(n=H) = 0$$
$$\frac{\partial \Delta \hat{u}(n)}{\partial \boldsymbol{\theta}^c} = \left[\frac{\partial \Delta \hat{u}(n)}{\partial K_p}, \frac{\partial \Delta \hat{u}(n)}{\partial K_I}, \frac{\partial \Delta \hat{u}(n)}{\partial K_D}\right] = [\hat{e}_p(n), \hat{e}_I(n), \hat{e}_D(n)]$$
$$J_c(n) = \frac{\partial \hat{e}(n)}{\partial \boldsymbol{\theta}^c} = \delta_c(n) \frac{\partial \Delta \hat{u}(n)}{\partial \boldsymbol{\theta}^c}$$

where $h'(\hat{u}(n)) = \partial h / \partial u|_{u(n)\,=\,\hat{u}(n)}$is the derivative of the limiter function with respect to its input. The backward action at the controller output is obtained as

$$\delta_c(n) = \delta_l(n) + \delta_c(n+1) \tag{21}$$

with $\delta_c(n=H) = 0$. The Jacobian with respect to the controller gains is calculated by

$$J_c(n) = \frac{\partial \hat{e}(n)}{\partial \boldsymbol{\theta}^c} = \delta_c(n) \frac{\partial \Delta \hat{u}(n)}{\partial \boldsymbol{\theta}^c} \tag{22}$$

Where the derivative $(\partial \Delta \hat{u} / \partial \boldsymbol{\theta}^c)$ is obtained as

$$\frac{\partial \Delta \hat{u}(n)}{\partial \boldsymbol{\theta}^c} = \left[\frac{\partial \Delta \hat{u}(n)}{\partial K_p}, \frac{\partial \Delta \hat{u}(n)}{\partial K_I}, \frac{\partial \Delta \hat{u}(n)}{\partial K_D}\right] = [\hat{e}_p(n), \hat{e}_I(n), \hat{e}_D(n)] \tag{23}$$

Table 2 summarizes the backward phase computations.

The PID controller gains are updated based on the elements of the Jacobian matrix ($J_c = \partial \hat{e} / \partial \boldsymbol{\theta}^c$). The change in parameter vector, $\Delta \boldsymbol{\theta}^c$, of the controller follows from

$$(J_c^T J_c + \mu I) \Delta \boldsymbol{\theta}^c = -J_c^T \hat{\mathbf{e}} \tag{24}$$

by the LM method. Here μ is a positive constant, I is the identity matrix, and the superscript T denotes transpose. For a sufficiently large value of μ, the matrix $(J_c^T J_c + \mu I)$ is positive definite, and $\Delta \boldsymbol{\theta}^c$ describes a descent direction. μ is determined with a trust region approach. A detailed description of the LM with the trust region approach is given in [32]. The computations are repeated for a certain iteration number at each stage to reduce the predicted tracking error to a suitable level. The adjusted controller gains are transferred to the controller of the application part which generates the control action to apply to the actual plant. Then the next stage of the adaptation process is started.

3.3. *The predictor training*

The predictor is a recurrent system using the FS defined by Eq. (5), and as such its Jacobian is computed by the back-propagation through time (BPTT) method. It can be trained off-line to accurately represent the process dynamics by using the input–output data of the process. An on-line training procedure is used to adapt to the changes in process dynamics (Fig. 3). In this procedure, a short history of the training data is stored in a first-in first-out stack of depth (L), where the oldest data is discarded, and the newest is entered. The stack length L is chosen to sufficiently represent as well as adapt the system dynamics, eliminating the too old data while keeping the recent changes.

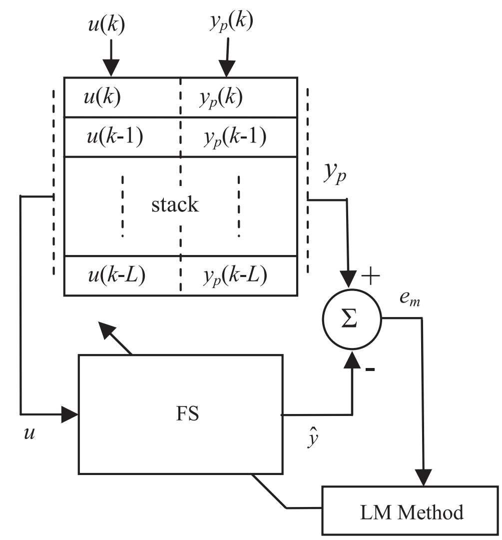

Fig. 3. The on-line predictor training.

The on-line training is performed using the stack data at each time step. The training of the predictor corresponds to adjusting the FS parameters, $\boldsymbol{\theta}^p = [\mathbf{b}^p, \mathbf{c}^p, \boldsymbol{\sigma}^p]$. The cost function at each time step k is the mean square error (MSE) between the actual plant outputs (y_p) and the predictor outputs ($\hat{y}$) in the stack, given by

$$E_m^L = \frac{1}{2L} \sum_{l=1}^{L} [y_p(l) - \hat{y}(l)]^2 \tag{25}$$

where k is omitted for brevity.

4. Illustrative benchmark example: the bioreactor control

The performance of the proposed control method is demonstrated on a nonlinear bioreactor benchmark problem [33], used in Refs. [34–36], since the strong nonlinearities and instabilities in bioreactor dynamics present a challenge for control. The bioreactor tank contains water in which nutrients and biological cells mix. The constituents of the tank are kept at a constant volume by keeping inflow and outflow rates equal to the same control input variable, u. The dimensionless state variables of the process, x_1 and x_2, are the amount of cells and nutrients in the tank, respectively. Coupled nonlinear differential equations describing the bioreactor process are

$$\begin{aligned} \frac{dx_1}{dt} &= -x_1 u + x_1(1-x_2)e^{x_2/\gamma} \\ \frac{dx_2}{dt} &= -x_2 u + x_1(1-x_2)e^{x_2/\gamma}(1+\beta)/(1+\beta-x_2) \end{aligned} \tag{26}$$

where $0 \le x_1$, $x_2 \le 1$, and $0 \le u \le 2$. The constant parameters γ and β determine the rates of cell formation and nutrient consumption, respectively. For the nominal benchmark specification, they are set as γ=0.48 and β=0.02 [33,34]. The simulation data sets are obtained by integrating Eq. (26) by a fourth-order Runge–Kutta algorithm using an integration step size of Δ=0.01 time units. We define 50Δ as the macro time step.

We first construct the predictor through the task of designing our adaptive PID control system. In the bioreactor problem, the

Please cite this article as: Savran A, Kahraman G. A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes. ISA Transactions (2013), http://dx.doi.org/10.1016/j.isatra.2013.09.020

predictor fuzzy system (FS) has 2 inputs ($N=2$) and 6 rules ($M=6$). The stack size L and the prediction horizon H are both set to 5. Initially, the Gaussian membership functions with equal spread are used, and their centers are equally placed to cover the entire domain. One of the fuzzy membership functions before and after training is shown in Fig. 4. The performance of the predictor is

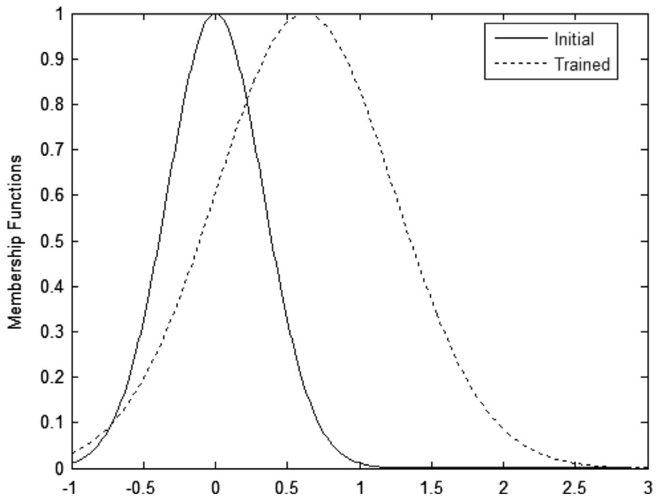

Fig. 4. The fuzzy membership function before and after training.

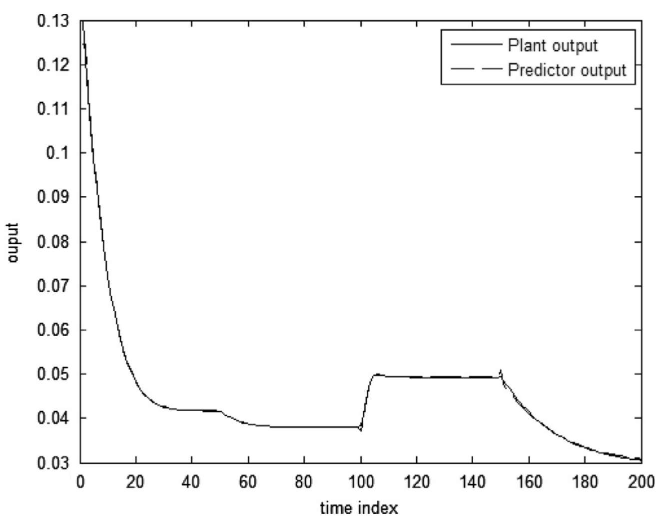

Fig. 5. The outputs of the plant and the fuzzy predictor.

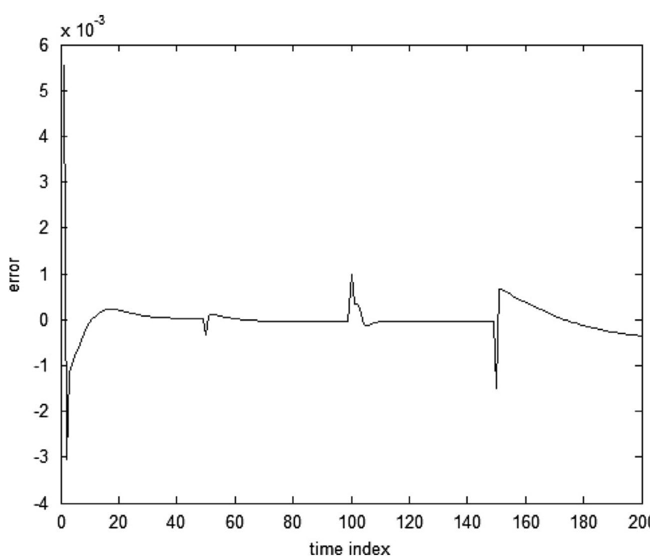

Fig. 6. Instantaneous error with respect to k.

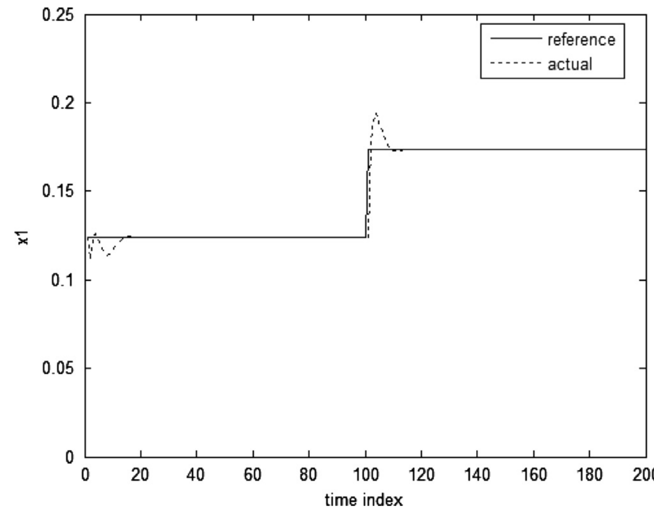

Fig. 7. The control system response in the stable ($k<100$) and unstable ($k>100$) regions.

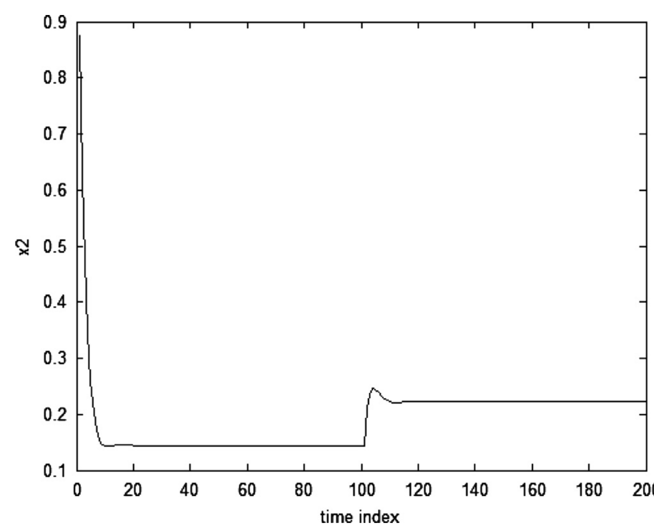

Fig. 8. The trajectory of the uncontrolled state x2.

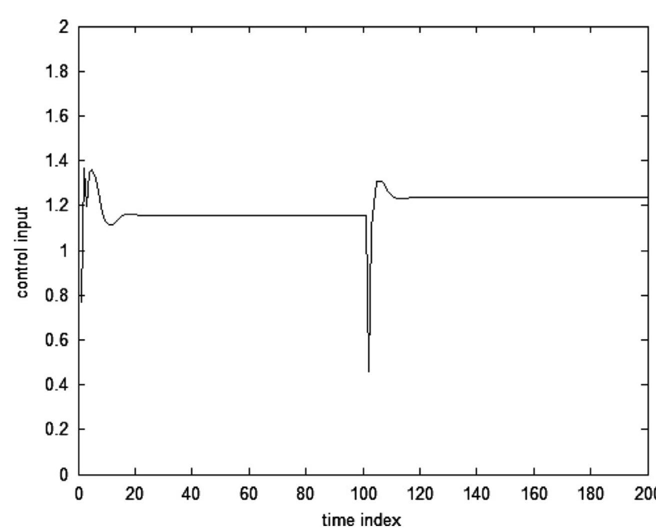

Fig. 9. The control input.

depicted in Figs. 5 and 6, where the plant output is estimated very closely with an off-line mean-square-error of 2.65×10^{-7}.

4.1. Control of the bioreactor in stable and unstable regions

We tested the performance of the control system for the nominal plant parameters (γ=0.48 and β=0.02). A sigmoid type limiter is considered to provide control input limits for the bioreactor. The limiter function is $2h(u)=2/(1+e^{-u})$, since the bioreactor control input must be between 0 and 2. We deal with the bioreactor control problem specified as the hardest by Ref. [33]. In this problem, the desired state is first set to a stable value $(x_1^*,x_2^*)=(0.1237,0.8760)$ with $u^*=1/1.3$. Then, after 100 micro time steps (50 s), 0.05 added to x_1^* giving $x_1^*=0.1737$. This shifts the set point from the stable region to the unstable region. The PID gains at the end of the simulation are obtained as $(K_p,K_I,K_D)=(-22.9408,-16.0045,-1.6895)$. The performance of our PID control system for this problem is shown in Figs. 7–9. The set points in both stable and unstable regions are satisfactorily tracked, where a mean square error of 2×10^{-5} is obtained.

4.2. Robustness to noise

The performance of the control system by injecting noise on the state variables is also demonstrated. This is performed by

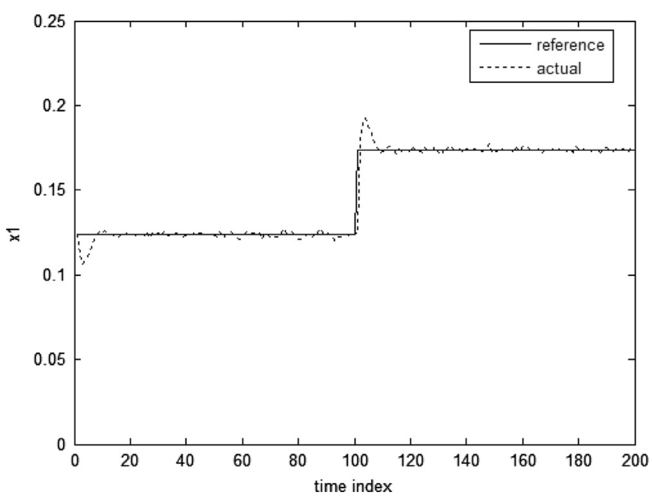

Fig. 10. The control system response under noise.

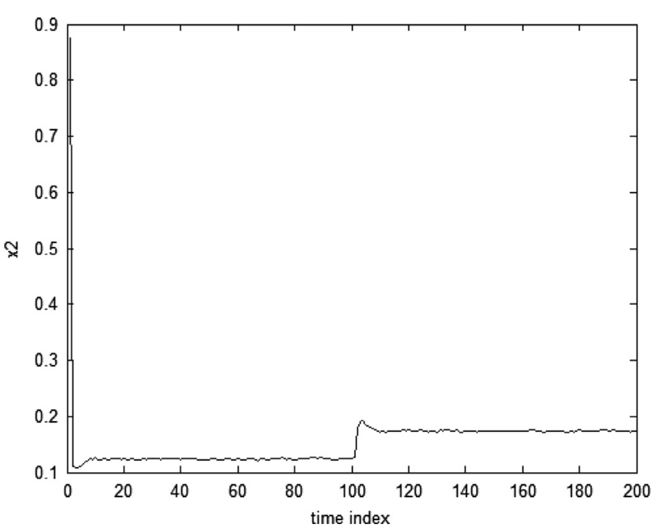

Fig. 11. The trajectory of the uncontrolled state x2 under noise.

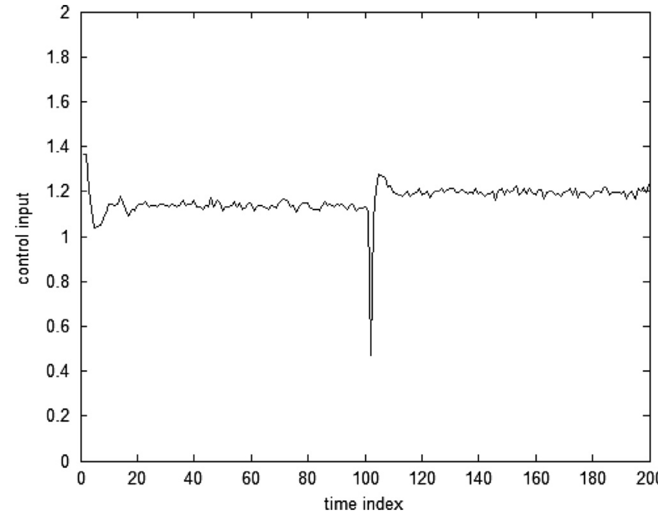

Fig. 12. The control input.

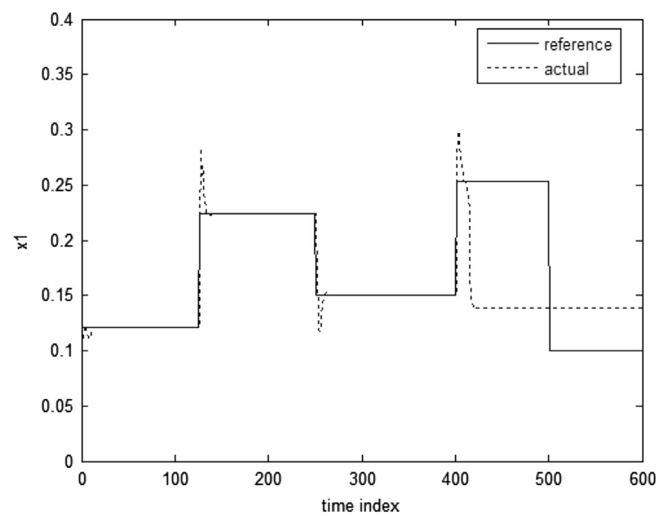

Fig. 13. The lack of adaptation of the control system with the fixed gains. The system parameters change at k=400.

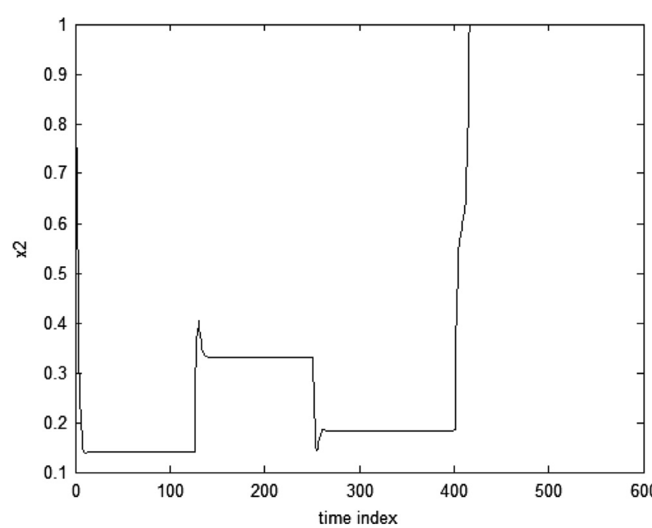

Fig. 14. The trajectory of the uncontrolled state x2 under fixed gains.

Please cite this article as: Savran A, Kahraman G. A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes. ISA Transactions (2013), http://dx.doi.org/10.1016/j.isatra.2013.09.020

adding the white Gaussian noise with a standard deviation of 0.001 to the state variables. The performance of the control system is depicted in Figs. 10–12. The control system sufficiently tracks the reference, showing the robustness to noise.

4.3. Adaptation performance

The adaptation is tested by altering the bioreactor parameters $\gamma=0.48$ to 0.456 (5%) and $\beta=0.02$ to 0.016 (20%) at the time index of 400. We performed simulations using the controller the gains of which are firstly fixed to their off-line trained values $(K_p, K_I, K_D)=(-22.9408, -16.0045, -1.6895)$ with ($\gamma=0.48$ and $\beta=0.02$), and secondly are adaptively on-line trained. Figs. 13 to 15 show the lack of adaptation to the system parameter change at $k=400$ in the first case.

The same simulation is carried out using the adaptive PID controller. The same initial values for the gains $(K_p, K_I, K_D)=(-22.9408, -16.0045, -1.6895)$ are used as in the fixed case. In this case, the gains continue to update. The variation of the PID controller gains with adaptation is shown in Fig. 16. Note that in both nominal ($k<400$) and altered ($k>400$) parameter regions, system is nonlinear but with a different set of parameters. Thus, the parameter change in the process is successfully compensated as shown in Figs. 17–19. The overall mean-squared-error (MSE) obtained is 1.5×10^{-4}.

4.4. Comparative study

The performance of the proposed control system is tested by comparing the two of the problems, namely “the nominal plant parameters ($\gamma=0.48$ and $\beta=0.02$)” and “a plant with altered parameters ($\gamma=0.456$ and $\beta=0.016$) with Gaussian measurement noise with a standard deviation of 0.01” both “with limited state information” studied on pages 291 and 292 in Ref. [34]. The reference trajectory is taken from figures 12 and 14 of Ref. [34]. Since only the state x_1 (the amount of cells) is used to design the controller and the predictor, our design relates to the limited-state problems stated in [34].

Firstly, the performance of the controller is tested with the nominal parameters. The performance of our system is depicted in Figs. 20–22. Our result has less ringing in the controlled state when compared with the corresponding results in figure 12 of Ref. [34]. We calculated the MSE as 1.95×10^{-4}.

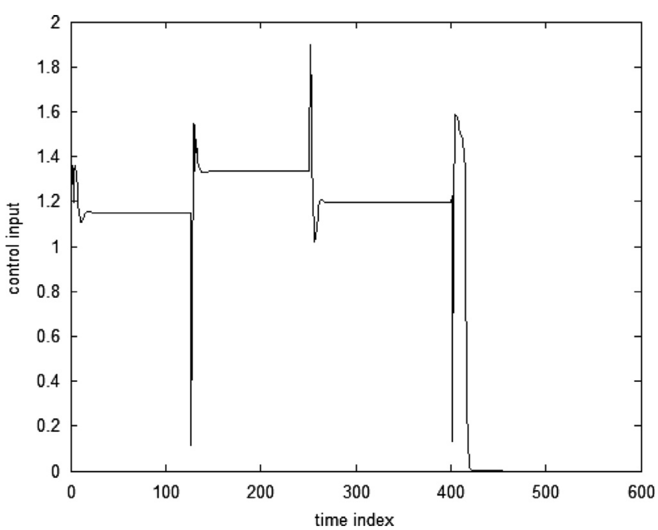

Fig. 15. The control input under fixed gains.

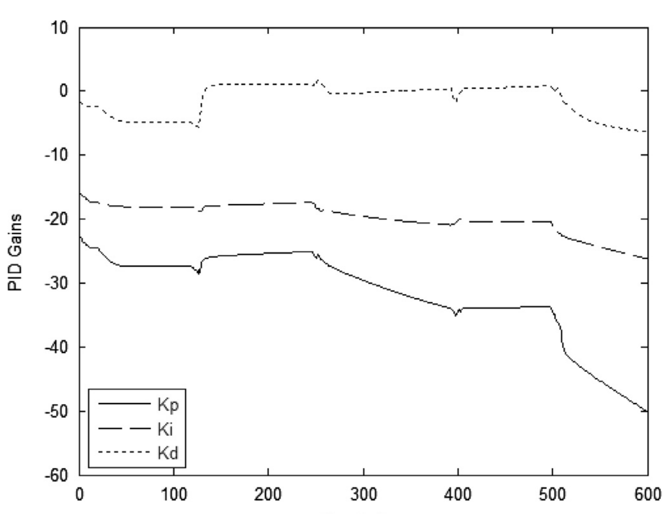

Fig. 16. The variation of the PID controller gains with adaptation.

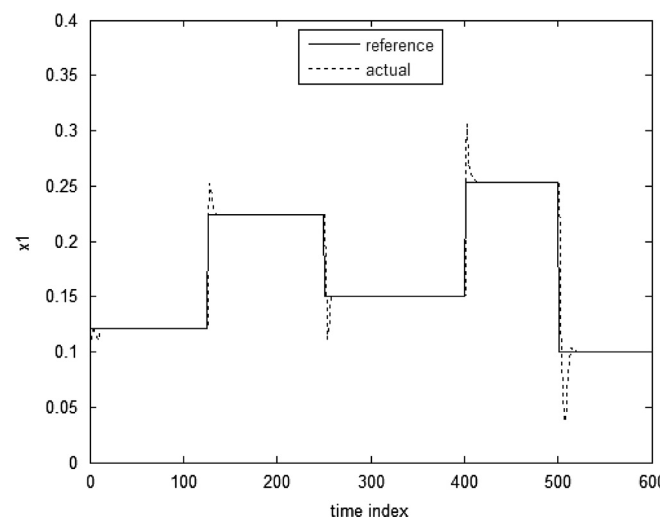

Fig. 17. The control system response for the nominal ($k<400$) and altered ($k>400$) system parameters. Note that in both regions system is nonlinear but with a different set of parameters.

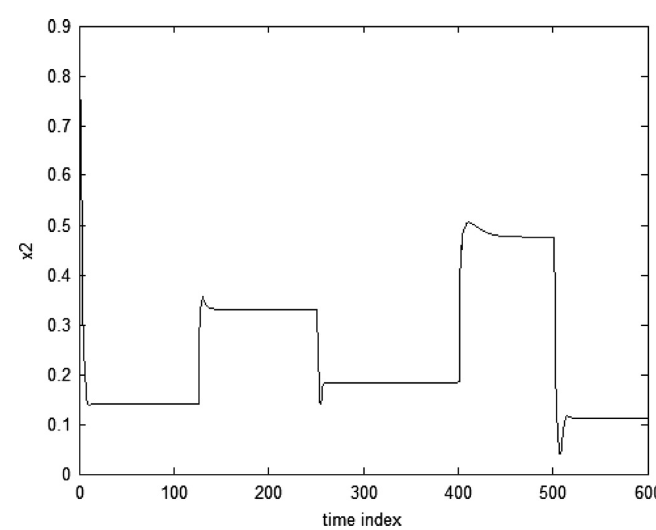

Fig. 18. The trajectory of the state x2 with the nominal ($k<400$) and altered ($k>400$) parameters.

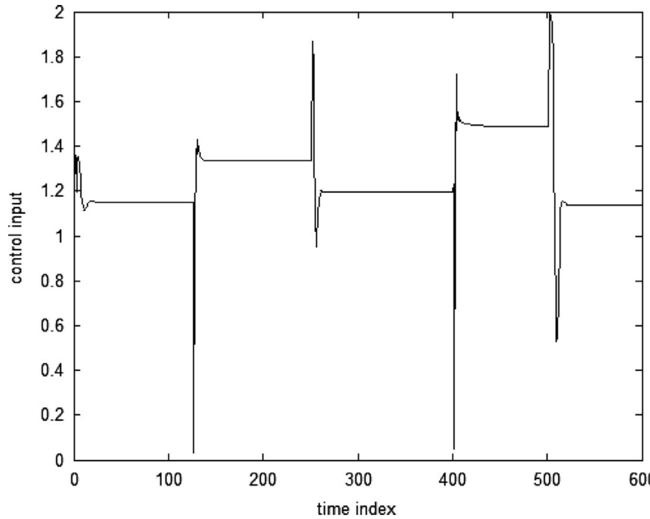

Fig. 19. The control input for the nominal ($k<400$) and altered ($k>400$) parameters.

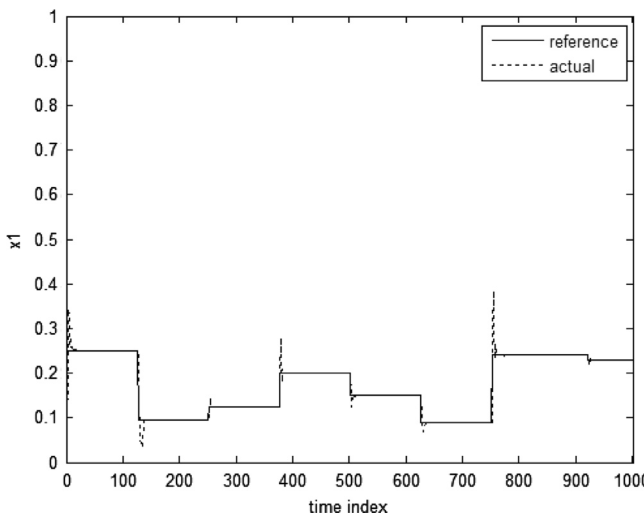

Fig. 20. The control system response for the nominal plant parameters.

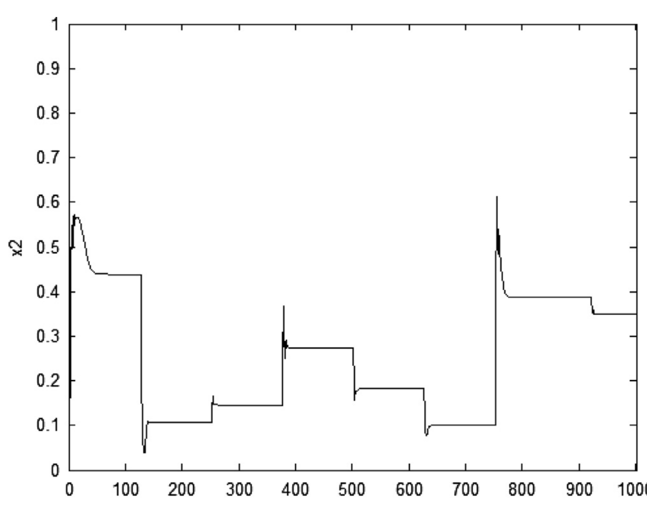

Fig. 21. The trajectory of the uncontrolled state x2 for the nominal plant parameters.

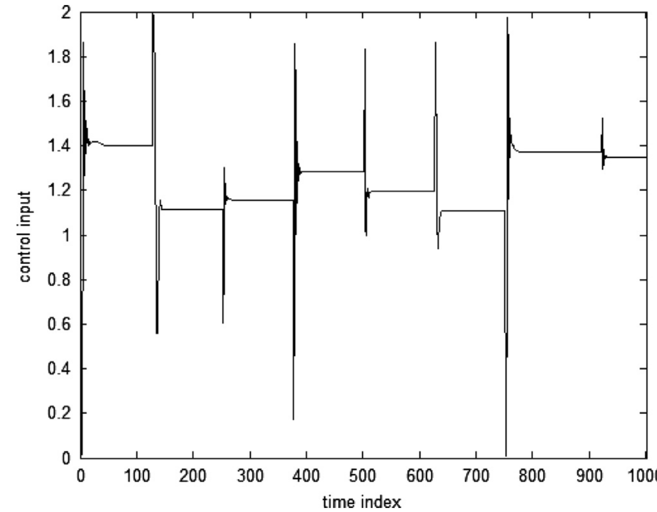

Fig. 22. The control input for the nominal plant parameters.

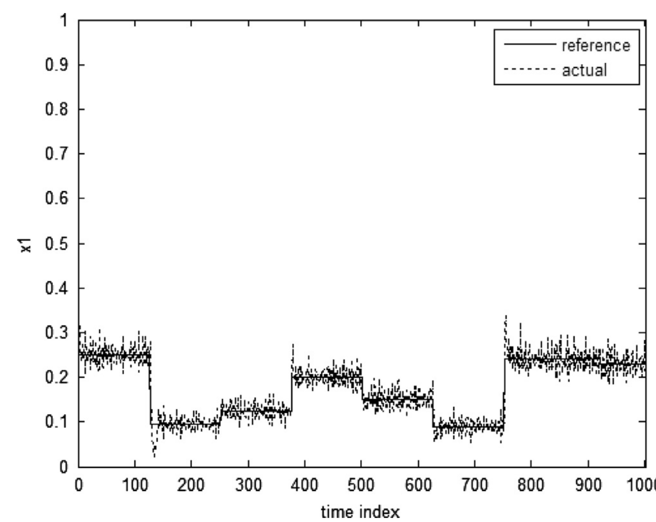

Fig. 23. The control system response for the altered plant parameters with Gaussian measurement noise of standard deviation 0.01.

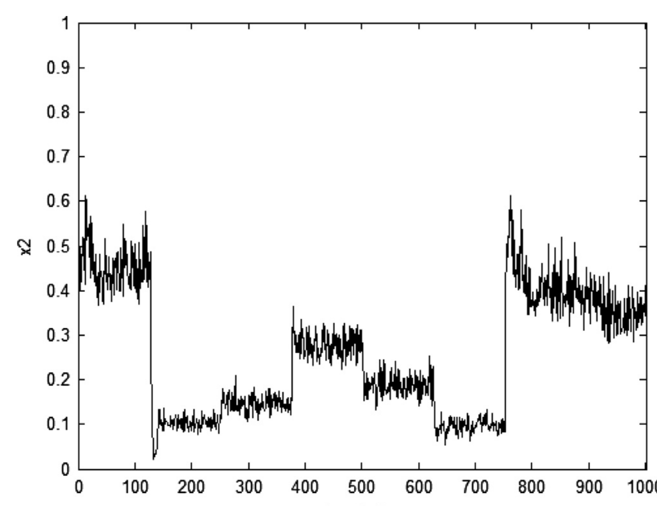

Fig. 24. The trajectory of the uncontrolled state x2 for the altered plant parameters with Gaussian measurement noise of standard deviation 0.01.

Please cite this article as: Savran A, Kahraman G. A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes. ISA Transactions (2013), http://dx.doi.org/10.1016/j.isatra.2013.09.020

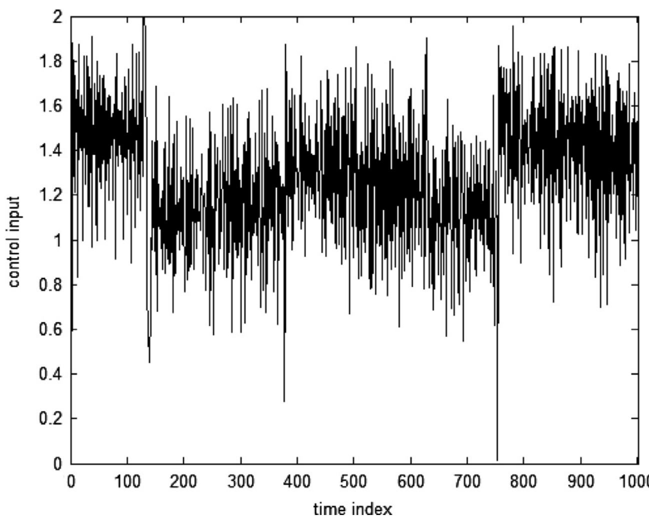

Fig. 25. The control input for the altered plant parameters with Gaussian measurement noise of standard deviation 0.01.

Secondly, the performance of the controller is tested on a plant with altered parameters with Gaussian measurement noise as stated above. This is the most difficult bioreactor control problem that is handled in [34]. Figs. 23–25 show the performance of our design. The control system is able to track the reference trajectory with quite less ringing and better tracking performance compared to those of figure 14 in [34]. Lower ringing appears on the uncontrolled state in our case as well. We calculated the MSE as 4.98×10^{-4}.

5. Conclusions

In this paper, a novel adaptive tuning method for the classical PID controller is developed to control nonlinear processes as well as adapt to abrupt parametrical changes. The adaptive control scheme invokes a PID controller in cascade with a fuzzy predictor, where the controller gains are adjusted online based on the predictions of a fuzzy predictor. The fuzzy predictor is constructed from the input–output data of the actual process, and thus does not require a first-principal model of the process. In the application part, a PID controller identical to that of the adaptation block is used to control the actual plant. The adjusted gain values are transferred to this identical controller at each control cycle.

The performance of the control system is successfully tested on several kinds of nonlinear problems based on the bioreactor, only three of which we presented. First, we showed that the controller tracked very well a bioreactor process that switches from a stable to an unstable region. In the second simulation, noise is added on the states of the first problem, and the control system proved to be robust under noise. The third simulation showed that the adaptive nonlinear control was properly achieved even when the system parameters are abruptly changed at a given time. This last problem involved a system that is both nonlinear and time-variant, and the control was excellent, while the fixed gain controller failed. Finally, we performed a comparative study on those of a plant with altered parameters with measurement noise, and obtained less ringing and better tracking. Our classic PID design allows computing the parameters from the measured input–output data (without requiring the first-principle based model) of the nonlinear process more precisely than other known designs by virtue of its multi-step ahead prediction, on-line predictor training, adaptive controller tuning, and fast training features. We think that our proposed classic PID-based method with its user-friendly realization will find wide-spread use in industrial applications involving nonlinear and uncertain processes.

References

[1] Åström KJ, Hägglund T. The future of PID control. Control Engineering Practice 2001;9(11):1163–75.
[2] Ang KH, Chong G, Li Y. PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology 2005;13(4):559–76.
[3] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. Transactions of the ASME 1942;64(8) (759/768–759/768).
[4] Marsh P. Turn on, tune in. New Electron 1998;31(4):31–2.
[5] Segovia JP, Sbarbaro D, Ceballos E. An adaptive pattern based nonlinear PID controller. ISA Transactions 2004;43:271–81.
[6] Gyongy IJ, Clarke DW. On the automatic tuning and adaptation of PID controllers. Control Engineering Practice 2006;14:149–63.
[7] Yamamoto T, Takao K, Yamada T. Design of a data-driven PID controller. IEEE Transactions on Control Systems Technology 2009;17(1):29–39.
[8] Deya C, Mudi RK. An improved auto-tuning scheme for PID controllers. ISA Transactions 2009;48:396–409.
[9] Garcia CE, Prett DM, Morari M. Model predictive control: theory and practice – a survey. Automatica 1989;25(3):335–48.
[10] Camacho EF, Bordons C. Model Predictive Control. London, Berlin, New York: Springer Verlag; 1999.
[11] Gu B, Gupta YP. Control of nonlinear processes by using linear model predictive control algorithms. ISA Transactions 2008;47:211–6.
[12] Zadeh LA. Fuzzy sets. Information and Control 1965;8:338–53.
[13] Zadeh LA. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics 1973;3:28–44.
[14] Kukolj D, Levi E. Identification of complex systems based on neural and Takagi–Sugeno fuzzy model. IEEE Transactions On Systems, Man, And Cybernetics – Part B: Cybernetics 2004;34(1):272–82.
[15] Mann GKI, Hu BG, Gosine RG. Analysis of direct action fuzzy PID controllers structures. IEEE Transactions on Systems, Man, and Cybernetics B 1999;29 (3):371–88.
[16] Takagi T, Sugeno M. Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics 1985;15(1):116–32.
[17] Sugeno M, Kang GT. Fuzzy modeling and control of multiplayer incinerator. Fuzzy Sets and Systems 1986;18:329–46.
[18] Huang YL, Lou HH, Gong JP, Edgar TF. Fuzzy model predictive control. IEEE Transactions on Fuzzy Systems 2000;8(6):665–78.
[19] Mahfouf M, Abbod MF, Linkens DA. Online elicitation of Mamdani-type fuzzy rules via TSK-based generalized predictive control. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 2003;33(3):465–75.
[20] Goodrich MA, Stirling WC, Frost RL. Model predictive satisficing fuzzy logic control. IEEE Transactions on Fuzzy Systems 1999;7(3):319–32.
[21] Mollov S, Babuska R, Abonyi J, Verbruggen HB. Effective optimization for fuzzy model predictive control. IEEE Transactions on Fuzzy Systems 2004;12 (5):661–75.
[22] Zhao YM, Xie WF, Tu XW. Performance-based parameter tuning method of model-driven PID control systems. ISA Transactions 2012;51:393–9.
[23] Padhya PK, Majhi S. Improved automatic tuning of PID controller for stable processes. ISA Transactions 2009;48:423–7.
[24] Maghade DK, Patre BM. Decentralized PI/PID controllers based on gain and phase margin specifications for TITO processes. ISA Transactions 2012;51:550–8.
[25] Hu W, Xiao G, Li X. An analytical method for PID controller tuning with specified gain and phase margins for integral plus time delay processes. ISA Transactions 2011;50:268–76.
[26] Vijayan V, Panda RC. esign of PID controllers in double feedback loops for SISO systems with set-point filters. ISA Transactions 2012;51:514–21.
[27] Savran A. A multivariable predictive fuzzy PID control system. Applied Soft Computing 2013;13:2658–67.
[28] Wang LX. Adaptive Fuzzy Systems and Control. Englewood Cliffs, NJ: Prentice-Hall; 1994.
[29] Wang LX. A Course in Fuzzy Systems and Control. Upper Saddle River, NJ: Prentice-Hall; 1997.
[30] Mills PM, Zomaya AY, Tade MO. Adaptive model-based control using neural networks. International Journal of Control 1994;60(6):1163–92.
[31] Scales LE. Introduction to Non-Linear Optimization. New York: Springer-Verlag New York Inc.; 1985; 115–8.
[32] Savran A. Multi-feedback-layer neural network. IEEE Transactions on Neural Networks 2007;18(2):373–84.
[33] Ungar LH. A bioreactor benchmark for adaptive network-based process control. In: Miller III WT, Sutton RS, Werbos PJ, editors. Neural Networks for Control. London: MIT Press; 1990. p. 476–80 (Chapter 16, 387–402 and Appendix A).
[34] Puskorius GV, Feldkamp LA. Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. IEEE Transactions on Neural Networks 1994;5(2):279–97.
[35] Savran A. An adaptive recurrent fuzzy system for nonlinear identification. Applied Soft Computing 2007;7:593–600.
[36] Efe MO. MIMO variable structure controller design for a bioreactor benchmark process. ISA Transactions 2007;46:459–69.