

Research Article

Neural Network-Based State Estimation for a Closed-Loop Control Strategy Applied to a Fed-Batch Bioreactor

Santiago Rómoli,¹ Mario Serrano,¹ Francisco Rossomando,² Jorge Vega,^{3,4} Oscar Ortiz,¹ and Gustavo Scaglia¹

¹*Instituto de Ingeniería Química, Universidad Nacional de San Juan, CONICET, Av. Lib. San Martín Oeste, 1109 San Juan, Argentina*

²*Instituto de Automática, Universidad Nacional de San Juan, CONICET, Av. Lib. San Martín Oeste, 1109 San Juan, Argentina*

³*Facultad Regional Santa Fe, Universidad Tecnológica Nacional, CONICET, Lavaisse 610, Santa Fe, Argentina*

⁴*Instituto de Desarrollo Tecnológico para la Industria Química (INTEC (UNL-CONICET)), Güemes, 3450 Santa Fe, Argentina*

Correspondence should be addressed to Mario Serrano; eserrano@fi.unsj.edu.ar

Received 15 June 2017; Accepted 9 July 2017; Published 5 September 2017

Academic Editor: Chenguang Yang

Copyright © 2017 Santiago Rómoli et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The lack of online information on some bioprocess variables and the presence of model and parametric uncertainties pose significant challenges to the design of efficient closed-loop control strategies. To address this issue, this work proposes an online state estimator based on a Radial Basis Function (RBF) neural network that operates in closed loop together with a control law derived on a linear algebra-based design strategy. The proposed methodology is applied to a class of nonlinear systems with three types of uncertainties: (i) time-varying parameters, (ii) uncertain nonlinearities, and (iii) unmodeled dynamics. To reduce the effect of uncertainties on the bioreactor, some integrators of the tracking error are introduced, which in turn allow the derivation of the proper control actions. This new control scheme guarantees that all signals are uniformly and ultimately bounded, and the tracking error converges to small values. The effectiveness of the proposed approach is illustrated on the basis of simulated experiments on a fed-batch bioreactor, and its performance is compared with two controllers available in the literature.

1. Introduction

In the last decade, several control strategies applied to different bioprocesses have been extensively investigated (e.g., [1–5]). In the particular case of fed-batch bioreactors, some nonlinear control strategies were developed to improve the process efficiency and the “batch-to-batch” reproducibility [6], as well as to deal with the unavoidable uncertainties typically present in bioprocesses [7].

Many articles have investigated the effects of model uncertainties on the performance of model-based nonlinear control strategies typically applied to bioprocesses. For example, the “yield coefficient of cell to glucose” in a bioprocess is a time-varying parameter often assumed as a constant. Also, the “specific growth rate” can be represented through the Monod-Haldane model [8], but it is possible that the model becomes inaccurate because of yeast growth inhibition due to an eventual presence of ethanol. In principle,

three main types of uncertainties can be envisaged [9–13]: (1) uncertain and/or time-varying model parameters, (2) uncertain nonlinearities, and (3) unmodeled dynamics. Additionally, nonmodeled external perturbations can disturb the process. In general, model uncertainties and external perturbations can cause severe risks in bioreactors, such as leading to undesirable operating conditions [14–16]. For instance, inadequate modeling of the inhibitory effect of a substrate can lead to a structural instability in the dynamical behavior of a bioprocess [15].

Adaptive control strategies are a valuable choice to control bioreactors due to their ability to compensate for parametric uncertainties. The Nussbaum gain method has been applied to control a bioprocess of unknown and time-varying control gain and parameters [17]. A fed-batch bioreactor control with simultaneous dynamic identification of unknown state and parameters has been investigated through an algorithm based on the auxiliary model principle with adaptive controllers

[18]. A control strategy was proposed to track a desired profile for the biomass of a yeast culture in a high-density fed-batch bioreactor [19]. This last strategy was based on a model that included an uncertainty term estimated through an adaptive technique. Also, an adaptive nonlinear feedback controller has been designed to automatically estimate uncertain time-varying parameters in order to guarantee the closed-loop control of a waste water plant [20].

Fuzzy control systems have been successfully applied to several bioreactors. For instance, [12] considered an adaptive fuzzy-decentralized control for a system with unknown nonlinear uncertainties and dynamical uncertainties. The fuzzy system was used to estimate the nonlinear uncertainties. Also, a fuzzy controller for the alcoholic fermentation with *Saccharomyces cerevisiae* has been proposed [21]. It combined the capability of handling uncertainties with the ability of predictive control to improve the plant performance, making use of a neural network model of the nonlinear system. Optimal control strategies have also been applied to bioprocesses [22, 23]. In order to compensate for process uncertainties, this type of controllers incorporates an additional term to the bioprocess model (typically, in the specific growth rate), which is estimated through an optimization technique. Alternative strategies for controlling bioreactors under model uncertainties are trajectory-based control [24], hybrid control [25], and model predictive control [26]. Additionally, due to the inherent nature of a bioprocess, a control scheme based on a stochastic model seems to be adequate. The specific growth rate can be modeled as affected by a white noise representing environment uncertainties, and the mathematical model consists of a set of stochastic differential equations [27] (some applications can be consulted in [28, 29]). The application of this approach onto a real plant is quite difficult due to its complex design and implementation.

Based on a linear algebra methodology, a control strategy has been applied for trajectory tracking in a bioprocess [30]. The technique was used to calculate the control actions that ensure the tracking of independently determined optimal trajectories of some process variables. To this effect, a discrete approximation of the original process model must first be derived, and the control actions are then calculated through a least square problem. This methodology has been applied to a fed-batch bioprocess designed to produce a secreted protein, and it was originally modeled by Park and Ramirez [31]. Optimal profiles of such bioprocess have also been published [32].

Many nonlinear control techniques of bioprocesses assume that the feedback variables can be measured online [33, 34]. However, such hypothesis is often unrealistic and therefore some process variables must be estimated through virtual sensors (or soft sensors). There are several kinds of virtual sensors published in the literature which exhibit different abilities to deal with model mismatches and perturbations. In this work, an artificial intelligence-based observer has been chosen because this type of estimator is especially recommended for highly nonlinear systems in the presence of uncertainties [35].

This paper aims at designing a controller for the multivariable tracking of optimal profiles when the process is

affected by uncertainties in the dynamic model and/or in the model parameters. Unmeasured process variables are estimated through a RBF neural network. The proposed approach is evaluated onto a bioprocess originally modeled by [31], with the optimal profiles determined by [32] in absence of model errors. Brief descriptions of the fed-batch bioreactor and the controller derived in [30] are presented in Appendix A.

In comparison with [30], the proposed approach prevents the polynomial uncertainties from affecting the tracking control. To achieve this aim, an integrative term is incorporated in the linear algebra methodology originally proposed. Then, the conditions in order for a system of linear equations to have exact solution are analyzed. The control action is obtained by solving the linear system, even though the original process model is nonlinear.

The main contribution is that the original method proposed herein ensures the convergence to zero of the tracking errors under additive uncertainties. Another important contribution is the inclusion of RBF neural network estimators: the control issue is addressed considering more realistic conditions with nonmeasurable variables. Further, the stability analysis is made using Lyapunov theory in discrete time.

The article is organized as follows. In Section 2, the controller design methodology and the convergence to zero of the tracking errors are presented. Section 3 shows briefly the description of the RBF estimator. A method to tune the controller is presented in Section 4, along with four testing cases designed to check the effectiveness of the new control law. Main conclusions of the work are summarized in Section 5.

2. Controller Design

Derivation of the control strategy is based on the mathematical model presented in Appendix A. The effect of an additive uncertainty (\mathbf{E}_n) is modeled by extending (A.4) as follows:

$$\mathbf{z}_{n+1} = \mathbf{z}_n + \mathbf{c}_n \Delta t - \mathbf{d}_n u_n \Delta t + \mathbf{E}_n, \quad (1)$$

where \mathbf{z}_n is the state vector, \mathbf{c}_n and \mathbf{d}_n are the input vectors, Δt is the sample time, u_n is the control action, and \mathbf{E}_n is an additive uncertainty. In principle, \mathbf{E}_n can be used to model either perturbed systems or a wide class of model mismatches. In what follows, a control law is designed to compensate for the tracking errors introduced by \mathbf{E}_n .

Taking into account the fact that the mismatch might depend on both system states and inputs, we here consider a real plant described by $\mathbf{z}_{n+1} = f(\mathbf{z}_n, \mathbf{u}_n)$. The additive uncertainty can be expressed as $\mathbf{E}_n = f(\mathbf{z}_n, \mathbf{u}_n) - \hat{f}(\mathbf{z}_n, \mathbf{u}_n)$, where $\hat{f}(\mathbf{z}_n, \mathbf{u}_n)$ is the discrete-time nonlinear system model. If \mathbf{z} and \mathbf{u} are assumed to be bounded and f is Lipschitz, then \mathbf{E}_n can be modeled as a bounded uncertainty [36–38].

With the additive uncertainty of (1), the fed-batch bioreactor model of (A.3) becomes

$$\begin{bmatrix} \frac{P_n}{V_n} \\ \frac{T_n}{V_n} \\ \frac{X_n}{V_n} \\ \frac{S_F - S_n}{V_n} \end{bmatrix} u_n^* = \begin{bmatrix} \chi_n (T_n - P_n) - \frac{P_{n+1} - P_n}{\Delta t} \\ \psi_n X_n - \frac{T_{n+1} - T_n}{\Delta t} \\ \mu_n X_n - \frac{X_{n+1} - X_n}{\Delta t} \\ 7.3\mu_n X_n + \frac{S_{n+1} - S_n}{\Delta t} \end{bmatrix} \quad (2)$$

$$+ \begin{bmatrix} E_{P,n} \\ E_{T,n} \\ E_{X,n} \\ E_{S,n} \end{bmatrix}.$$

The control action for the secreted protein concentration, P , given by (A.4) of [30], is

$$u_{n,P} = \frac{\chi (T_n - P_n) - (P_{\text{ref},n+1} - k_p (P_{\text{ref},n} - P_n) - P_n) / \Delta t}{P_n / V_n}. \quad (3)$$

The discrete value of P , P_{n+1} , is calculated by replacing (3) into the first row of (2):

$$P_{n+1} = P_{\text{ref},n+1} - k_p (P_{\text{ref},n} - P_n) + \Delta t (T_n - P_n) (\chi_n - \chi) + \Delta t E_{P,n}, \quad (4)$$

where χ_n and χ depend on S_n and $S_{e_z,n}$, respectively. Thus, the tracking error of P_n is

$$e_{P,n+1} = P_{\text{ref},n+1} - P_{n+1}. \quad (5)$$

By combining (4) and (5), one obtains

$$e_{P,n+1} = k_p e_{P,n} - \Delta t (T_n - P_n) (\chi_n - \chi) - \Delta t E_{P,n}. \quad (6)$$

The Taylor interpolation of χ_n at its desired value χ is

$$\chi_n = \chi + \frac{d\chi}{dS} \Big|_{S=S_{e_z,n} + \lambda(S_n - S_{e_z,n})=S_\lambda} (S_n - S_{e_z,n}); \quad (7)$$

$$0 < \lambda < 1.$$

By replacing (7) into (6), the tracking error for P_n becomes

$$e_{P,n+1} = k_p e_{P,n} - \Delta t (T_n - P_n) \left[\chi + \frac{d\chi}{dS} \Big|_{S=S_\lambda} (S_n - S_{e_z,n}) - \chi \right] - \Delta t E_{P,n}, \quad (8)$$

$$e_{P,n+1} = k_p e_{P,n} + \Delta t (T_n - P_n) \frac{d\chi}{dS} \Big|_{S=S_\lambda} (S_{e_z,n} - S_n) - \Delta t E_{P,n}.$$

If a similar procedure is followed for T , X , and S , then the error equation becomes [30]

$$\mathbf{e}_{n+1} = \begin{bmatrix} e_{P,n+1} \\ e_{T,n+1} \\ e_{X,n+1} \\ e_{S,n+1} \end{bmatrix} = \begin{bmatrix} k_p & 0 & 0 & 0 \\ 0 & k_T & 0 & 0 \\ 0 & 0 & k_X & 0 \\ 0 & 0 & 0 & k_S \end{bmatrix} \begin{bmatrix} e_{P,n} \\ e_{T,n} \\ e_{X,n} \\ e_{S,n} \end{bmatrix} + \Delta t \begin{bmatrix} (T_n - P_n) \frac{d\chi}{dS} \Big|_{S=S_\lambda} \\ X_n \frac{d\psi}{dS} \Big|_{S=S_\varphi} \\ X_n \frac{d\mu}{dS} \Big|_{S=S_\delta} \\ 0 \end{bmatrix} e_{S,n} - \begin{bmatrix} E_{P,n} \\ E_{T,n} \\ E_{X,n} \\ E_{S,n} \end{bmatrix}. \quad (9a)$$

Alternatively, in vector form,

$$\mathbf{e}_{n+1} = \mathbf{K} \mathbf{e}_n + \mathbf{h}_n - \mathbf{E}_n, \quad (9b)$$

where \mathbf{h}_n is a nonlinearity that tends to zero when $e_{S,n}$ tends to zero. Equations (9a) and (9b) suggest a direct effect of the additive uncertainty on the tracking error. Therefore, the presence of an uncertainty avoids a convergence of tracking errors to zero (see Theorem 1 of [30]).

2.1. Single Integral Action. In order to reduce the effect of \mathbf{E}_n , some integrators for the tracking errors will be introduced, depending on the time variation hypothesis of \mathbf{E}_n . We will assume that \mathbf{E}_n is unknown and each of its components is an m -order polynomial.

Remark 1. The first-order difference of \mathbf{E}_n is defined as $\delta \mathbf{E}_n = \mathbf{E}_{n+1} - \mathbf{E}_n$, the second-order difference is defined as $\delta^2 \mathbf{E}_n = \delta(\delta \mathbf{E}_n) = \delta(\mathbf{E}_{n+1} - \mathbf{E}_n) = \mathbf{E}_{n+2} - 2\mathbf{E}_{n+1} + \mathbf{E}_n$, and, as a rule, the q -th order difference is defined as $\delta^q \mathbf{E}_n = \delta(\delta^{q-1} \mathbf{E}_n)$.

Remark 2. The q -th difference of a $q - 1$ -order polynomial is zero.

Consider first a constant uncertainty $\mathbf{E}_n = \text{const}$. Then, $\delta \mathbf{E}_n = \mathbf{E}_{n+1} - \mathbf{E}_n = \mathbf{0}$. In this case, an integrator for each state variable will force the error to converge to zero. Call $\mathbf{e}(t)$ the continuous time error in the state vector. Then, the integral of the tracking error is defined as

$$\mathbf{U}_{1,n+1} = \mathbf{U}_{1,n} + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{e}(t) dt \cong \mathbf{U}_{1,n} + \mathbf{e}_n \Delta t. \quad (10)$$

Hence, the control action (A.7) will be computed by assuming a new term in (A.5); that is,

$$\begin{aligned} \mathbf{z}_{n+1} &= \mathbf{z}_{\text{ref},n+1} - \mathbf{K}(\mathbf{z}_{\text{ref},n} - \mathbf{z}_n) + \mathbf{k}_1 \mathbf{U}_{1,n+1}, \\ \mathbf{k}_1 &= \begin{bmatrix} k_{P1} & 0 & 0 & 0 \\ 0 & k_{T1} & 0 & 0 \\ 0 & 0 & k_{X1} & 0 \\ 0 & 0 & 0 & k_{S1} \end{bmatrix}, \end{aligned} \quad (11)$$

where \mathbf{K} and \mathbf{k}_1 are the proportional and integral design parameters, respectively.

By adding one integrator ($\mathbf{k}_1 \mathbf{U}_{1,n+1}$) to (9b) and using (10), one obtains

$$\mathbf{e}_{n+1} + \mathbf{k}_1 (\mathbf{U}_{1,n+1} + \mathbf{e}_n \Delta t) = \mathbf{K} \mathbf{e}_n + \mathbf{h}_n - \mathbf{E}_n. \quad (12)$$

Rewriting (12) for the next sample time,

$$\mathbf{e}_{n+2} + \mathbf{k}_1 (\mathbf{U}_{1,n+2} + \mathbf{e}_{n+1} \Delta t) = \mathbf{K} \mathbf{e}_{n+1} + \mathbf{h}_{n+1} - \mathbf{E}_{n+1}. \quad (13)$$

$$\begin{aligned} u_{1n} &= \frac{P_n V_n (\chi (T_n - P_n) - (P_{\text{ref},n+1} - k_P (P_{\text{ref},n} - P_n) - P_n + k_{P1} U_{P1,n+1}) / \Delta t) + T_n V_n (\psi X_n - (T_{\text{ref},n+1} - k_T (T_{\text{ref},n} - T_n) - T_n + k_{T1} U_{T1,n+1}) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2} \\ &+ \dots \\ &+ \frac{X_n V_n (\mu X_n - (X_{\text{ref},n+1} - k_X (X_{\text{ref},n} - X_n) - X_n + k_{X1} U_{X1,n+1}) / \Delta t) + (20 - S_n) V_n (7.3 \mu X_n + (S_{ez,n+1} - k_S (S_{ez,n} - S_n) - S_n + k_{S1} U_{S1,n+1}) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2}. \end{aligned} \quad (16)$$

2.2. Multiple Integral Actions. Assume that the uncertainty can be modeled by a function with a second-order difference equal to zero; that is, $\delta^2 \mathbf{E}_n = \delta(\delta \mathbf{E}_n) = \delta(\mathbf{E}_{n+1} - \mathbf{E}_n) = \mathbf{E}_{n+2} - 2\mathbf{E}_{n+1} + \mathbf{E}_n = \mathbf{0}$. Similar to (10), a double integrator is now introduced, and new variables \mathbf{U}_1 and \mathbf{U}_2 are defined:

$$\mathbf{U}_{2,n+1} = \mathbf{U}_{2,n} + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{U}_1(t) dt \cong \mathbf{U}_{2,n} + \Delta t \mathbf{U}_{1,n+1}. \quad (17)$$

In this case, the control action (A.7) will include an additional term in (A.5) as follows:

$$\begin{aligned} \mathbf{z}_{n+1} &= \mathbf{z}_{\text{ref},n+1} - \mathbf{K}(\mathbf{z}_{\text{ref},n} - \mathbf{z}_n) + \mathbf{k}_1 \mathbf{U}_{1,n+1} + \mathbf{k}_2 \mathbf{U}_{2,n+1}, \\ \mathbf{k}_2 &= \begin{bmatrix} k_{P2} & 0 & 0 & 0 \\ 0 & k_{T2} & 0 & 0 \\ 0 & 0 & k_{X2} & 0 \\ 0 & 0 & 0 & k_{S2} \end{bmatrix}, \end{aligned} \quad (18)$$

where \mathbf{K} , \mathbf{k}_1 , and \mathbf{k}_2 are the proportional, integral, and double-integral design parameters, respectively. With a similar procedure to that used for deriving (15) and after adding

Combining (10) and (12) and substituting them into (13),

$$\begin{aligned} \mathbf{e}_{n+2} + \mathbf{k}_1 \mathbf{e}_{n+1} \Delta t + \mathbf{h}_n - \mathbf{E}_n - \mathbf{e}_{n+1} + \mathbf{k}_1 \mathbf{e}_n \\ = \mathbf{K} \mathbf{e}_{n+1} + \mathbf{h}_{n+1} - \mathbf{E}_{n+1}. \end{aligned} \quad (14)$$

Alternatively,

$$\begin{aligned} \mathbf{e}_{n+2} + [-\mathbf{K} - \mathbf{I}] \mathbf{e}_{n+1} + \mathbf{k}_1 \mathbf{e}_{n+1} \Delta t + \mathbf{K} \mathbf{e}_n \\ = -(\mathbf{h}_{n+1} - \mathbf{h}_n) - (\mathbf{E}_{n+1} - \mathbf{E}_n). \end{aligned} \quad (15)$$

Therefore, \mathbf{K} and \mathbf{k}_1 are chosen in order to ensure the stability of the linear system represented in the left-hand side of (15). To achieve this condition, the roots of such polynomial (r_i) must lie inside the unit circle. Then, $\mathbf{e}_n \rightarrow 0$ when $n \rightarrow \infty$; that is, the error tends to zero as long as uncertainties are constant.

Once the integral of the error has been added, the control action is calculated following the same design procedure based on linear algebra presented in [30]. Therefore, the new control law u_n can be obtained through the least squares procedure as

two integrators ($\mathbf{k}_1 \mathbf{U}_{1,n+1}$ and $\mathbf{k}_2 \mathbf{U}_{2,n+1}$), the error dynamics can be expressed as

$$\begin{aligned} \mathbf{e}_{n+3} + [-\mathbf{K} - 2\mathbf{I}] \mathbf{e}_{n+2} + (\mathbf{k}_1 + \mathbf{k}_2 \Delta t) \mathbf{e}_{n+2} \Delta t \\ + [2\mathbf{K} + \mathbf{I}] \mathbf{e}_{n+1} - \mathbf{k}_1 \mathbf{e}_{n+1} \Delta t - \mathbf{K} \mathbf{e}_n \\ = \begin{bmatrix} h_{P,n+2} - 2h_{P,n+1} + h_{P,n} \\ h_{T,n+2} - 2h_{T,n+1} + h_{T,n} \\ h_{X,n+2} - 2h_{X,n+1} + h_{X,n} \\ 0 \end{bmatrix} - \delta^2 \mathbf{E}_n. \end{aligned} \quad (19)$$

As suggested by (19), under the assumption of a constant or linear varying uncertainty ($\delta^2 \mathbf{E}_n = \mathbf{0}$), the uncertainty has no influence on the error dynamics. As in the case of the single integrator, the controller parameters \mathbf{K} , \mathbf{k}_1 , and \mathbf{k}_2 should be chosen in order to ensure the stability of the left-hand side of (19).

The previous derivation can be generalized for uncertainties that can be approximated with a $q - 1$ -order polynomial; and, therefore, the influence of \mathbf{E}_n upon \mathbf{e}_n would be cancelled off by addition of q integrators.

The linear algebra-based procedure enables the calculation of the following control law:

$$\begin{aligned}
u_{2n} = & \frac{P_n V_n (\chi (T_n - P_n) - (P_{\text{ref},n+1} - k_P (P_{\text{ref},n} - P_n) - P_n + k_{P1} U_{P1,n+1} + k_{P2} U_{P2,n+1}) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2} + \dots \\
& + \frac{T_n V_n (\psi X_n - (T_{\text{ref},n+1} - k_T (T_{\text{ref},n} - T_n) - T_n + k_{T1} U_{T1,n+1} + k_{T2} U_{T2,n+1}) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2} + \dots \\
& + \frac{X_n V_n (\mu X_n - (X_{\text{ref},n+1} - k_X (X_{\text{ref},n} - X_n) - X_n + k_{X1} U_{X1,n+1} + k_{X2} U_{X2,n+1}) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2} + \dots \\
& + \frac{(20 - S_n) V_n (7.3 \mu X_n + (S_{\text{ez},n+1} - k_S (S_{\text{ez},n} - S_n) - S_n + k_{S1} U_{S1,n+1} + k_{S2} U_{S2,n+1}) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2}.
\end{aligned} \tag{20}$$

3. Neural State Estimator

The proposed feedback control technique requires reliable estimates of several state variables. The state estimation error, $\tilde{\mathbf{x}}_n$, is defined as

$$\tilde{\mathbf{x}}_n = \mathbf{x}_n - \hat{\mathbf{x}}_n, \tag{21}$$

where $\hat{\mathbf{x}}_n$ is the vector of estimated state variables and $\mathbf{x}_n = [\hat{P}_n, \hat{T}_n, \hat{X}_n]$ is the vector of online measured states. The following neural estimator represents the nonlinear dynamic of the bioreactor described in (7):

$$\begin{aligned}
\mathbf{x}_{n+1} &= \boldsymbol{\theta}^{*T} \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) + \boldsymbol{\varepsilon}_n, \\
\mathbf{x}_0 &= \mathbf{x}(0),
\end{aligned} \tag{22}$$

where $\mathbf{v}_n = [u_n, S_n, V_n]$ is the input vector to the neural estimator, $\boldsymbol{\theta}^*$ is the optimal weight vector, $\boldsymbol{\varepsilon}$ is the neural approximation error, ξ_i is the RBF that represents each neuron in the hidden layers, subindex i indicates the neuron number of the RBF, and m is the maximum number of neurons. Every neuron of the hidden layer is modeled as

$$\begin{aligned}
\xi_i(\mathbf{x}_n, \mathbf{v}_n) &= \exp\left(-\frac{1}{2\sigma_i^2} \left\| \begin{bmatrix} \mathbf{x}_n \\ \mathbf{v}_n \end{bmatrix} - \mathbf{c}_i \right\|^2\right), \\
i &= 1, 2, \dots, m,
\end{aligned} \tag{23}$$

where \mathbf{c}_i and σ_i are two parameter vectors that represent the centers and widths of the RBF, respectively.

In the bioreactor under study, biomass (X) cannot be measured online and the protein concentrations (P, T) cannot be measured at all; then a state estimator function based on (22) is determined as follows:

$$\hat{\mathbf{x}}_{n+1} = \hat{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n). \tag{24}$$

From the difference between (22) and (24), the neural identification error can be described by

$$\begin{aligned}
\tilde{\mathbf{x}}_{n+1} &= \mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1} = \boldsymbol{\theta}^{*T} \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) - \hat{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) + \boldsymbol{\varepsilon}_n \\
&= \tilde{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) + \boldsymbol{\varepsilon}_n,
\end{aligned} \tag{25}$$

where $\tilde{\boldsymbol{\theta}}_n$ represents the neural weight vector estimation error and it is defined as

$$\tilde{\boldsymbol{\theta}}_n = \boldsymbol{\theta}^* - \boldsymbol{\theta}_n. \tag{26}$$

In order to train the neural network for identification, an offline data set of (\mathbf{x}, \mathbf{u}) will be used. This training data will be obtained after following the same procedure presented in [39] and then applied in [40, 41]. A similar procedure was also used in [42]. The learning rule utilized to train the neural network is demonstrated in the next theorem.

Theorem 3. *The model of the fed-batch bioreactor (see (A.1) and (2)) can be approximated by the neural network (24) using a neuronal adjustment law defined by*

$$\Delta \tilde{\boldsymbol{\theta}}_n = -\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \boldsymbol{\Lambda}. \tag{27}$$

Proof. The demonstration of this theorem is presented in Appendix B. \square

As recommended by [35], the estimator was evaluated to test its performance and avoid biased estimates. Table 1 summarizes the implemented tests, and their results were evaluated through the average values of the mean squared errors (MSE) corresponding to P, T , and X . The obtained results show an acceptable performance of the RBF neural network for estimating the nonmeasurable states. Note that none of the MSE values exceeds the thresholds proposed by [43], 1×10^{-3} . Figure 1 shows the proposed closed-loop architecture. Note that the controller requires information on both measured and estimated variables.

4. Results and Discussion

All simulation examples are based on the bioprocess originally modeled by [31]. The available optimal profiles for such process are also adopted [32]. Equation (A.1) represents the fed-batch bioreactor model. The effectiveness of the new control law is tested through some simulation examples. First, the optimal controller parameters are tuned through a Monte Carlo experiment. Then, a sinusoidal perturbation is added to some model parameters. Also, the controller performance

TABLE I: Performance evaluation of the RBF neural network.

Test	MSE _T	MSE _p	MSE _X
Normal operating conditions	1.58×10^{-6}	2.14×10^{-6}	7.37×10^{-4}
Continuous variations in the substrate concentration (15%)	2.00×10^{-6}	2.47×10^{-6}	8.17×10^{-4}
20% of error in model parameters (S_F and $Y_{S/X}$)	1.98×10^{-6}	2.70×10^{-6}	9.51×10^{-4}
20% lesser in the initial nominal operating conditions in V , S , and X	1.83×10^{-6}	2.25×10^{-6}	3.80×10^{-4}
Normally distributed noise with zero mean and variance of 0.2 in u and S	1.93×10^{-6}	2.35×10^{-6}	7.51×10^{-4}

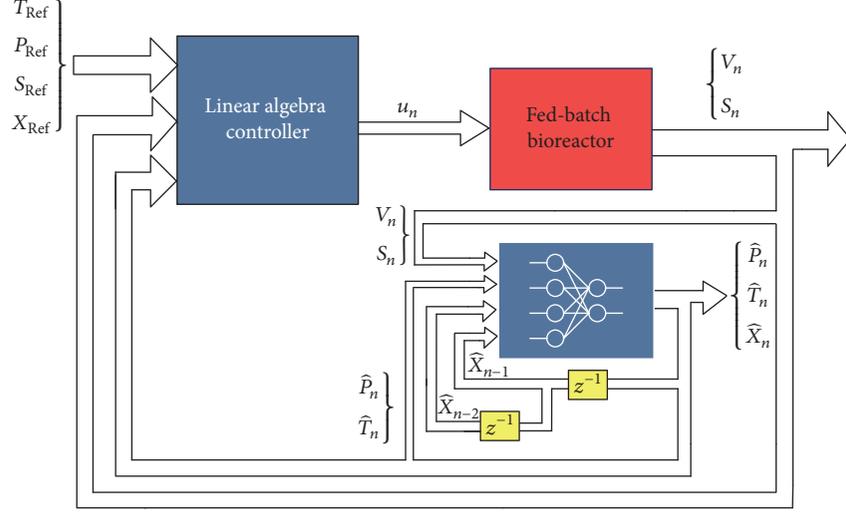


FIGURE 1: The proposed closed-loop control architecture: the linear algebra-based controller and the Radial Basis Function (RBF) neural network estimator.

under uncertain time-varying parameters is verified through a Monte Carlo Randomized Algorithm. In other examples, the functionality of the specific growth rate is changed. Finally, some parameters are contaminated by additive white noise.

As previously discussed, the behavior of the controlled system depends on parameters $\mathbf{k}, \mathbf{K}_1, \dots, \mathbf{K}_p$, where p is the number of integrators. For $p = 0, 1, 2$, the controller parameters are determined through a Monte Carlo method, where the number of simulations is determined by application of Theorem 2 of [30]. All simulations are implemented in Matlab™ (MathWorks, Inc.), with a sample time $\Delta t = 0.01$ h. For each controller (i.e., for each p), 1,000 simulations are run; and the controller parameters are randomly chosen, such that the roots (r_i) of the linear systems defined in the right-hand side of (9a) and (9b) and left-hand side of (15) and (19) verify $r_i \in (0, 1)$. More specifically, for $p = 0$, the controller parameters are directly taken from [30]. For $p = 1$, 1,000 new simulations are performed, and each controller parameter is chosen according to

$$\begin{aligned} k^i &= r_1 r_2, \\ K_1^i &= \frac{-r_1 - r_2 + r_1 r_2 + 1}{\Delta t}, \end{aligned} \quad (28)$$

where the superscript i refers to each state variable and r_1 and r_2 are the roots of the linear system in the left-hand side of (15). For $p = 2$, 1,000 simulations are performed, and each controller parameter is chosen according to

$$\begin{aligned} k^i &= r_1 r_2 r_3, \\ K_1^i &= \frac{2r_1 r_2 r_3 + 1 - r_1 r_2 - r_1 r_3 - r_2 r_3}{\Delta t}, \\ K_2^i &= \frac{-r_1 - r_2 - r_3 - r_1 r_2 r_3 + 1 + r_1 r_2 + r_1 r_3 + r_2 r_3}{\Delta t^2}, \end{aligned} \quad (29)$$

where r_1 , r_2 , and r_3 are the roots of the linear system in (19).

Table 2 summarizes the results obtained with each controller. For comparison purposes, the performance of the proposed strategy is compared with that provided by a Parallel Two-Degree-of-Freedom PID Controller (2-DOF PID), which computes a weighted difference signal for each of the proportional, integral, and derivative actions according to the set point weights specified by the user [44]. In this case, the culture cell density has been chosen as the controlled variable. The 2-DOF PID was tuned with the automatic tuning of the Simulink™ Control Design software that provides the proportional, integral, and derivative gains, as well as the filter coefficient.

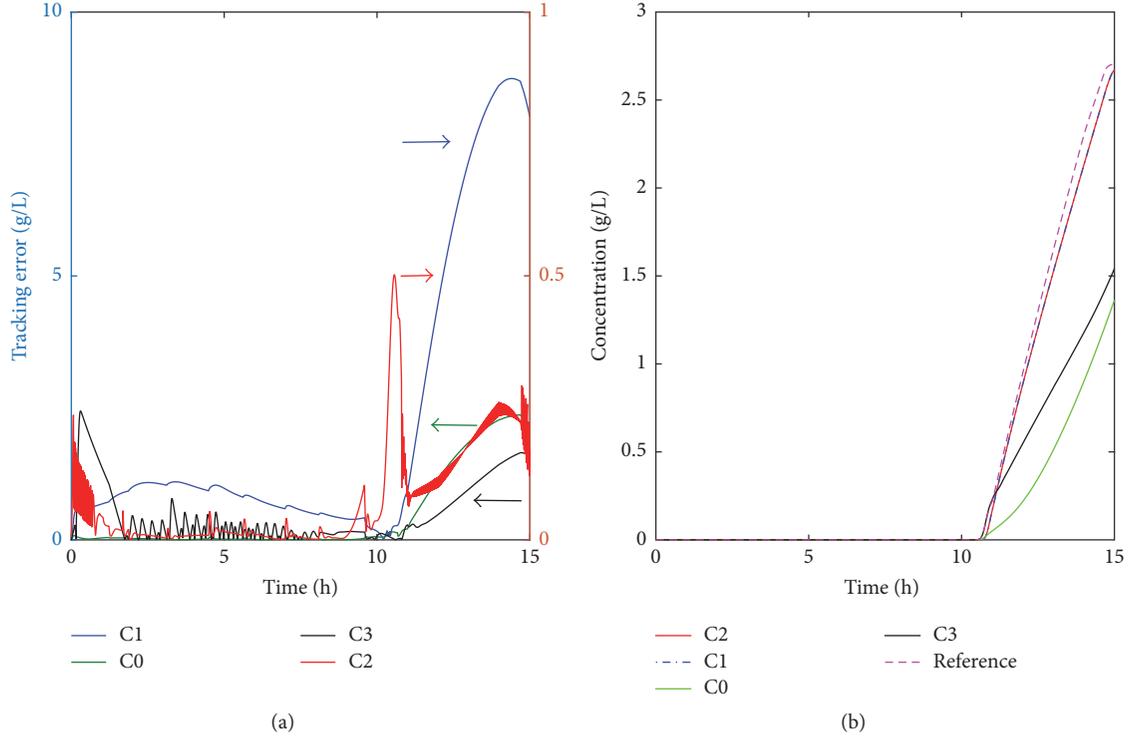


FIGURE 2: (a) Comparison of the tracking errors (see (31)) (left ordinate axis: C0 and C3; right axis: C2 and C1). (b) Total protein concentration profiles: C1 and C2 successfully reach the maximum production of protein in the presence of time-varying parameters.

TABLE 2: Main simulation results in the Monte Carlo experiments.

Controller	Controller parameters		
$p = 0$	$K_p = 0.23$		
	$K_T = 0.14$		
	$K_X = 0.52$		
	$K_S = 0.23$		
$p = 1$	$K_p = 0.7758$	$k_{p1} = 0.043$	
	$K_T = 0.6181$	$k_{T1} = 0.083$	
	$K_X = 0.8173$	$k_{X1} = 0.008$	
	$K_S = 0.0356$	$ks_1 = 0.013$	
$p = 2$	$K_p = 0.7758$	$k_{p1} = 0.053$	$k_{p2} = 0.037$
	$K_T = 0.6181$	$k_{T1} = 0.084$	$k_{T2} = 0.052$
	$K_X = 0.8173$	$k_{X1} = 0.097$	$k_{X2} = 0.065$
	$K_S = 0.0356$	$ks_1 = 0.099$	$ks_2 = 0.015$

4.1. *Time-Varying Parameters and Input Disturbances.* As in [18, 45], the following expressions are adopted for simulating uncertainties in the parameters and the feed flow rate:

$$\begin{aligned}
 \hat{Y}_{S/X} &= 7.3 + 0.05 \sin(t), \\
 \hat{S}_F &= 20 + 0.05 \sin(t), \\
 \hat{u} &= u + 0.05 \sin(t).
 \end{aligned} \tag{30}$$

The tracking error (e_n) is defined as

$$\begin{aligned}
 \|e_n\|_2 &= \sqrt{(P_{\text{ref},n} - P_n)^2 + (T_{\text{ref},n} - T_n)^2 + (X_{\text{ref},n} - X_n)^2 + (S_{e_z,n} - S_n)^2}.
 \end{aligned} \tag{31}$$

In what follows, we shall call C0, C1, and C2 the controllers that use $p = 0$, $p = 1$, and $p = 2$ integrators, respectively, and C3 the 2-DOF PID controller. Particularly note that C0 corresponds to the controller designed in [30]. In spite of the assumed time-varying parameters, Figure 2(a) shows that C2 produces the minimum tracking error. According to Figure 2(b), both C2 and C1 are effective to reach the desired protein concentration profile, while C0 and C3 produce large errors.

In several research fields, probabilistic methods have been found to be useful for dealing with problems related to robustness of systems affected by uncertainties [46]. Particularly, the Monte Carlo Randomized Algorithm has been used in many applications such as radioactive decay, power system generation, and traffic on roads [47]. In the control area, Monte Carlo methods are effective tools for the analysis of probabilistically robust control schemes [30, 46]. A Monte Carlo simulation is here performed to demonstrate the effectiveness of C2 from a statistical point of view [48–50], when uncertainties are introduced by time-varying parameter. To this effect, 1,000 random values of S_F and $Y_{S/X}$ are chosen in a range of $\pm 25\%$ of their nominal values. Figures 3(a), 3(c), 3(e), and 3(g) show that only C2 and C1 were able to adequately track the reference profile. In fact, both controller

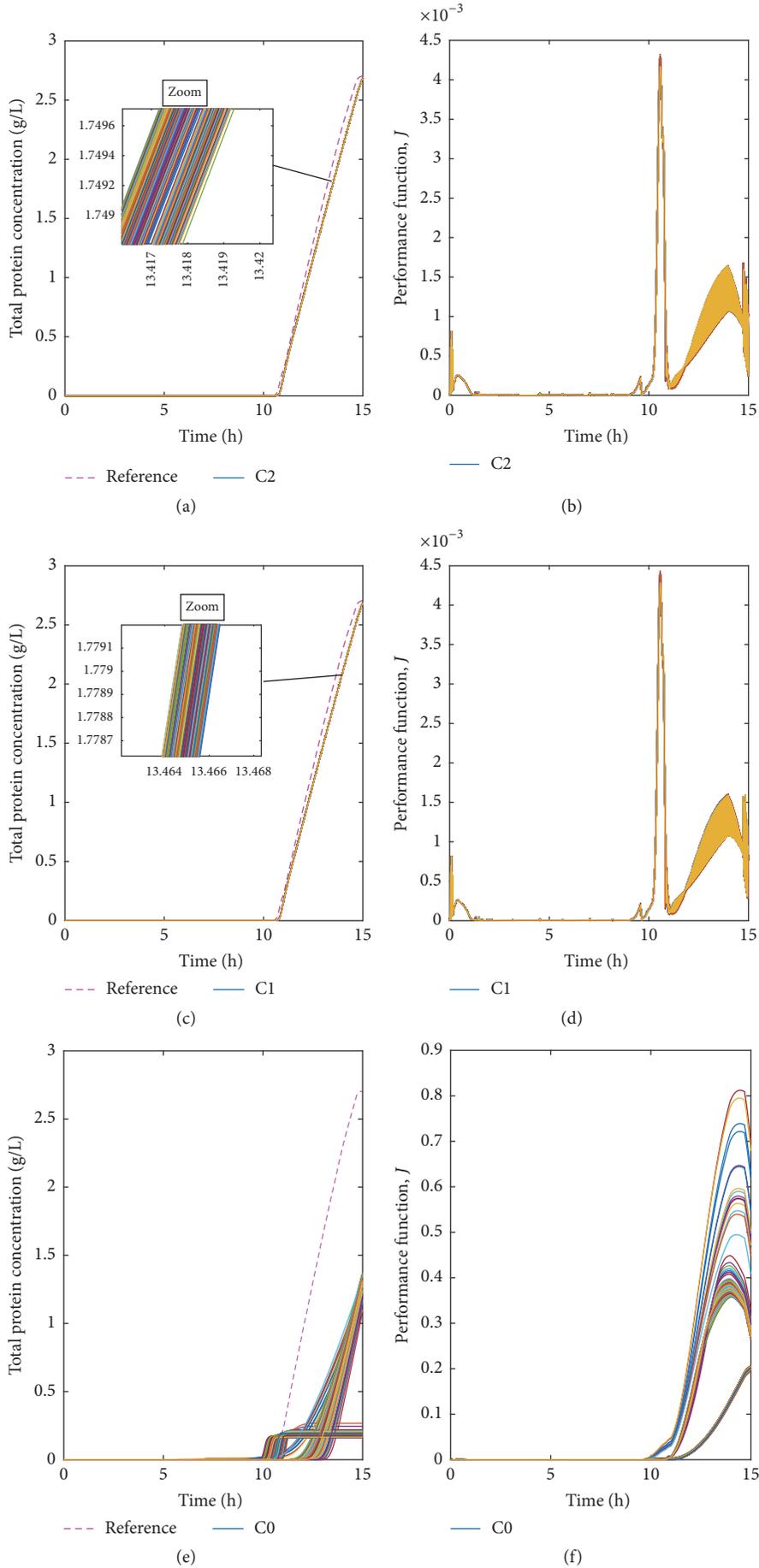


FIGURE 3: Continued.

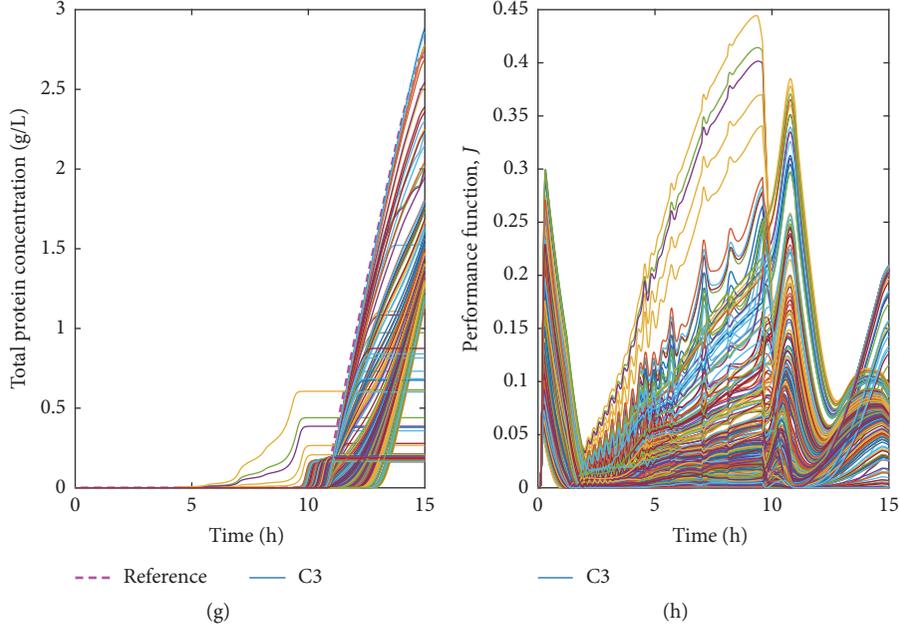


FIGURE 3: Total protein concentration ((a), (c), (e), and (g)) and performance index J ((b), (d), (f), and (h)) for 1,000 random values of S_F and $Y_{S/X}$ (controllers are indicated in each figure).

responses are quite similar, thus suggesting that the time-varying parameter has not affected their performances. For

comparison, the tracking error was evaluated through the following performance index J :

$$J = \frac{0.25 \sum_{n=1}^{N_{\text{ref}}} ((P_{\text{ref},n} - P_n)^2 + (T_{\text{ref},n} - T_n)^2 + (X_{\text{ref},n} - X_n)^2 + (S_{\text{ez},n} - S_n)^2)}{\Delta t N_{\text{ref}}}, \quad (32)$$

where N_{ref} represents the number of points of the reference profile. Figures 3(b), 3(d), 3(f), and 3(h) confirm a better performance of C2 and C1 over C0 and C3. In fact, J is quite small for C2 and C1, even though 25% of uncertainty in the main bioreactor parameters was adopted. Since 1,000 simulations with random values of S_F and $Y_{S/X}$ were carried out and the tracking errors remained bounded, the performance of C2 and C1 will be acceptable with 99% of probability whatever the parameter values are in a range of $\pm 25\%$ [46].

4.2. Uncertain Nonlinearities. Uncertain nonlinearities are often present in many bioprocesses. For instance, the specific growth rate, μ , can suffer from important changes from batch to batch, and unfortunately such variability is impossible to be accurately modeled. There are a number of models in the literature for μ [8], for example, the third expression of (A.2), for the bioprocess currently under study [31]. To demonstrate the controller performance, an additional term can be included in the calculation of μ as follows [18, 51]:

$$\hat{\mu} = \frac{(21.87 + 0.28 \sin(t)) S}{(S + 0.4)(S + 62.5)}. \quad (33)$$

Figure 4(a) compares the performances reached by C2, C1, C0, and C3 under the simulated uncertain nonlinearities in

μ . The lower tracking errors correspond to C2. Figure 4(b) shows that only C2 and C1 are able to reach the maximum concentration of the total protein.

4.3. Unmodeled Dynamics. The presence of white noise in μ_{max} leads to a mathematical model governed by a set of stochastic differential equations. In order to properly deal with this type of uncertainties, a kind of stochastic control approach should be used. As an alternative to the complex theory of stochastic control, a simpler deterministic strategy is here proposed. To this effect, note first that all parameters in (A.1) and (A.2) are truly affected by environmental noise, and consequently they always fluctuate at long times around some average values. For simplicity, we will consider that only μ_{max} is affected by randomness and changes according to the following expression [52, 53]:

$$d\mu_{\text{max}} = \mu_{\text{max}} + \alpha dB(t), \quad (34)$$

where $dB(t)$ is a standard Brownian motion defined on a complete probability space and $\alpha \geq 0$ represents the intensity of the noisy signal.

A white noise can be approximated through a band-limited white noise (BLWN), which is simply implemented through a proper block of Simulink. Theoretically, a white

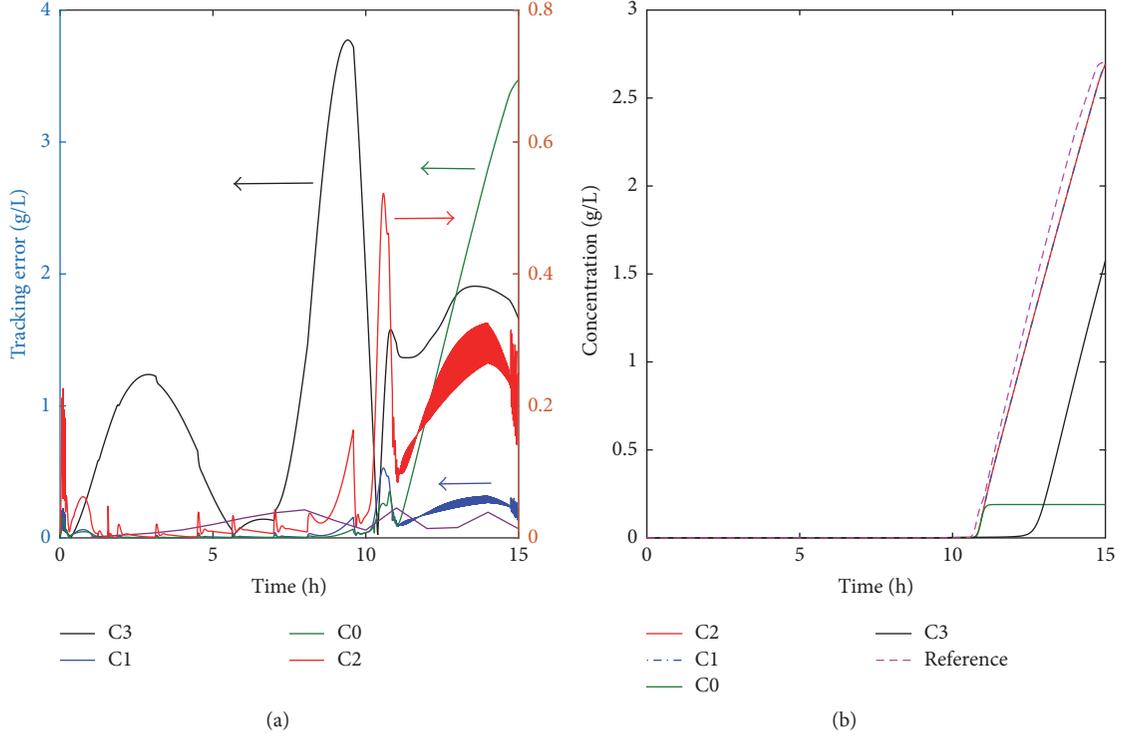


FIGURE 4: (a) Comparison of the tracking errors (see (31)) under uncertain nonlinearities in μ (left ordinate axis: C0, C1, and C3; right axis: C2). (b) Total protein concentration profiles.

noise is uncorrelated; it exhibits a flat power spectral density, and therefore it is characterized by an infinite energy. The BLWN block can simulate the effect of white noise by using a random sequence with a correlation time much smaller than the shortest time constant of the system. The correlation time of the noise is the sample rate of the block [54] and can be estimated as

$$t_c \approx \frac{2\pi}{100} \frac{1}{f_{\max}} = \frac{2\pi}{100} \frac{1}{0.5/\Delta t_{\max}}, \quad (35)$$

where t_c is the correlation time, f_{\max} is the bandwidth of the system, and Δt_{\max} is the maximum sample time for which the system is stable. For the feedback control presented in this work, $\Delta t_{\max} = 0.0145$ h. Then, from (35), $t_c = 0.0018$ h.

Many researchers have utilized this methodology to prove the robustness of the bioreactors controllers [19, 22–24, 55]. In the current simulations, a BLWN with a noise power of 0.9 and $\alpha = 0.5$ is introduced along with the feedback signal. Thus, the maximum specific growth rate affected with BLWN is calculated as

$$\tilde{\mu}_{\max} = \mu_{\max} + 0.05\text{BLWN}. \quad (36)$$

Figure 5(a) indicates that C2 and C1 adequately work in the presence of unmodeled dynamics. In fact, the tracking errors are quite small in comparison with C0 and C3. Figure 5(b) shows the profile tracking for each controller. Again, C2 and C1 are able to follow the reference profile, while C0 and C3 fail to reach the maximum protein concentration.

5. Conclusions

A new control law for tracking an optimal concentration profile in a fed-batch bioreactor has been presented. The proposed method allows controlling a nonlinear system in presence of unavoidable uncertainties in bioreactors: time-varying parameters, uncertain nonlinearities, and unmodeled dynamics.

In order to deal with these uncertainties, some integrators have been added to the original controller proposed in [30], and a Radial Basis Function (RBF) neural network was employed to estimate X , P , and T . The integrators have been chosen based on the variation hypothesis of the modeling error (E_n): if a q -grade polynomial can approximate E_n , then $q + 1$ integrators will be added in order to avoid its influence on the tracking error (e_n).

Several tests were carried out to demonstrate the effectiveness of the proposed methodology. The controllers are tuned through a Monte Carlo experiment and under stability conditions. To evaluate the performance of the designed controllers, different uncertainties were simulated. It was proven that the proposed methodology successfully compensates the uncertainties, and the tracking errors decrease as long as the number of integrators increases. The performance of the proposed controller with two integrators (C2) is remarkable, and the complexity of the control algorithm is relatively small. Moreover, the addition of integrators does not increase significantly the calculation time.

In comparison to previous strategies that deal with similar control problems (e.g., [17, 21, 22, 24, 25, 27]), the proposed

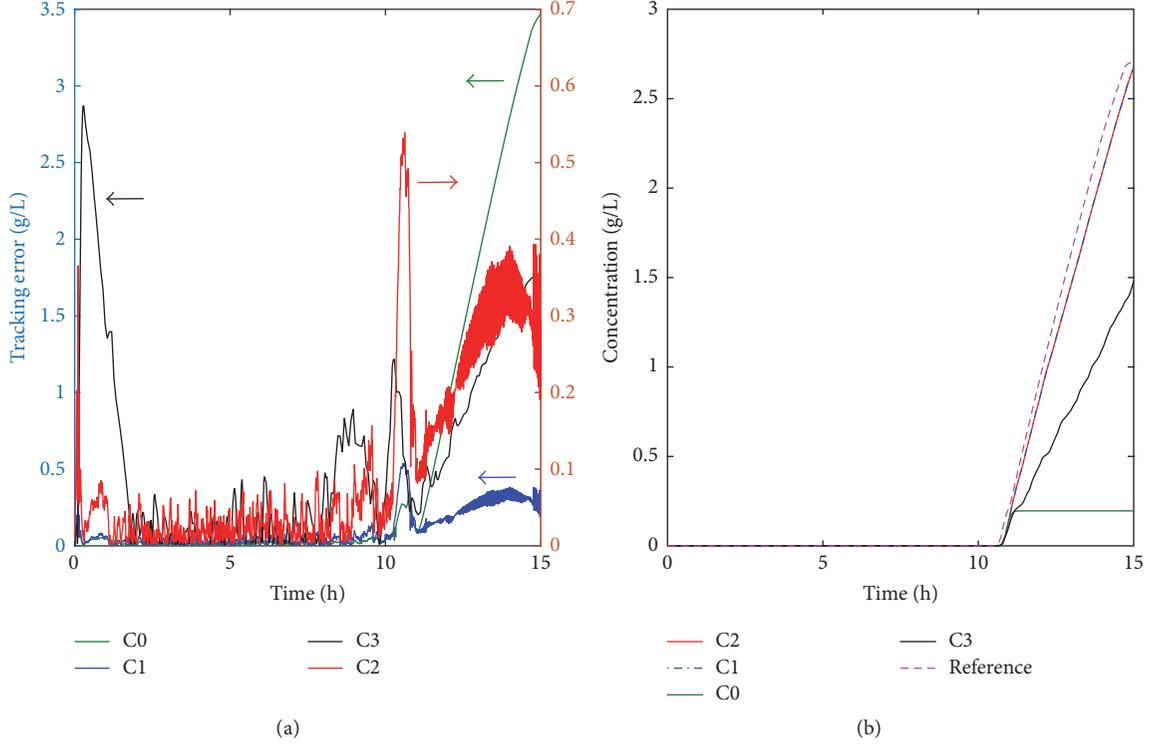


FIGURE 5: (a) Comparison of the tracking errors (see (31)) (right ordinate axis: C2; left axis: C0, C1, and C3). (b) Total protein concentration profiles.

controllers present the advantage of avoiding a stochastic modeling, as it is typically required to deal with systems where the parameters are contaminated by white noise. The trade-off between the response speed and the stability is solved through a simple Monte Carlo experiment. In addition, the controller design is mathematically simple. Currently, the developed methodology for the controller design is being applied to another bioprocess. Besides, the inclusion of saturation of the control signals and sliding mode control schemes in the formulation of the problem will be addressed in future contributions.

Appendix

A. Linear Algebra-Based Controller

Below is a brief description of the fed-batch bioreactor and the controller derived in [30]. The state variables in a fed-batch reactor can be described by the following set of equations proposed in [31]:

$$\begin{aligned}
 \frac{dP}{dt} &= \chi(T - P) - \frac{uP}{V}, \\
 \frac{dT}{dt} &= \psi X - \frac{uT}{V}, \\
 \frac{dX}{dt} &= \mu X - \frac{uX}{V}, \\
 \frac{dS}{dt} &= -Y_{S/X}\mu X + \frac{u(S_F - S)}{V}
 \end{aligned} \tag{A.1}$$

with

$$\begin{aligned}
 \chi &= \frac{4.75\mu}{0.12 + \mu}, \\
 \psi &= \frac{Se^{-5S}}{0.1 + S}, \\
 \mu &= \frac{\mu_{\max}S}{(S + 0.4)(S + 62.5)}, \\
 u &= \frac{dV}{dt}.
 \end{aligned} \tag{A.2}$$

In (A.1) and (A.2), the following variables and parameters are used: the amount of secreted protein per culture volume unit (P), the total protein amount per culture volume unit (T), the culture cell density (X), the culture glucose concentration (S), and the culture volume (V); u is the feed flow rate, S_F is the glucose concentration of the feed stream, $Y_{S/X}$ is the yield of glucose per cell mass, and ψ , μ , and χ are the protein expression rate, the specific growth rate of the host cell, and the protein secretion rate, respectively. The optimal profiles for the state variables calculated in [32] were chosen as the reference profiles $[P_{\text{ref}}, T_{\text{ref}}, X_{\text{ref}}, S_{\text{ref}}]^T$ to be tracked by the control algorithm. The nominal values of the model parameters are $S_F = 20 \text{ gL}^{-1}$ and $Y_{S/X} = 7.3$. Initial conditions for the state variables are $P_0 = 0 \text{ gL}^{-1}$, $T_0 = 0 \text{ gL}^{-1}$, $X_0 = 1 \text{ gL}^{-1}$, $S_0 = 5 \text{ gL}^{-1}$, and $V_0 = 1 \text{ L}$.

In order to follow the reference profiles, the required control action is calculated from the process model. First,

(A.1) can be rewritten in a discrete version using the Euler approximation:

$$\begin{bmatrix} \frac{P_n}{V_n} \\ \frac{T_n}{V_n} \\ \frac{X_n}{V_n} \\ \frac{S_F - S_n}{V_n} \end{bmatrix} u_n = \begin{bmatrix} \chi_n (T_n - P_n) - \frac{P_{n+1} - P_n}{\Delta t} \\ \psi_n X_n - \frac{T_{n+1} - T_n}{\Delta t} \\ \mu_n X_n - \frac{X_{n+1} - X_n}{\Delta t} \\ 7.3\mu_n X_n + \frac{S_{n+1} - S_n}{\Delta t} \end{bmatrix}. \quad (\text{A.3})$$

Let us denote by \mathbf{z}_n the state vector and by \mathbf{c}_n and \mathbf{d}_n the input vectors. Then, the state vector in the sample time $(n + 1)$ can be expressed as

$$\begin{aligned} \mathbf{z}_{n+1} &= \mathbf{z}_n + \mathbf{c}_n \Delta t - \mathbf{d}_n u_n \Delta t, \\ \mathbf{z}_n &= \begin{bmatrix} P_n \\ T_n \\ X_n \\ S_n \end{bmatrix}; \\ \mathbf{c}_n &= \begin{bmatrix} \chi(T_n - P_n) \\ \psi X_n \\ \mu X_n \\ -7.3\mu X_n \end{bmatrix}; \\ \mathbf{d}_n &= \begin{bmatrix} \frac{P_n}{V_n} & \frac{T_n}{V_n} & \frac{X_n}{V_n} & \frac{S_n - 20}{V_n} \end{bmatrix}^T. \end{aligned} \quad (\text{A.4})$$

The following replacements should be done in (A.3):

$$\mathbf{z}_{n+1} = \mathbf{z}_{\text{ref},n+1} - \mathbf{K}(\mathbf{z}_{\text{ref},n} - \mathbf{z}_n),$$

$$\begin{aligned} u_n &= \frac{P_n V_n (\chi(T_n - P_n) - (P_{\text{ref},n+1} - k_P(P_{\text{ref},n} - P_n) - P_n) / \Delta t) + T_n V_n (\psi X_n - (T_{\text{ref},n+1} - k_T(T_{\text{ref},n} - T_n) - T_n) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2} \\ &+ \dots \\ &+ \frac{X_n V_n (\mu X_n - (X_{\text{ref},n+1} - k_X(X_{\text{ref},n} - X_n) - X_n) / \Delta t) + (20 - S_n) V_n (7.3\mu X_n + (S_{\text{ez},n+1} - k_S(S_{\text{ez},n} - S_n) - S_n) / \Delta t)}{P_n^2 + T_n^2 + X_n^2 + (20 - S_n)^2}. \end{aligned} \quad (\text{A.7})$$

B. Proof of Theorem 3

Assumption B.1. The optimal approximation error ε is bounded by a constant: $\|\varepsilon\| \leq \bar{\varepsilon}$.

Proof of Theorem 3. Define a candidate function of Lyapunov, with V_m being a positive definite function given by

$$V_m = \tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + \gamma^{-1} \text{tr}(\tilde{\boldsymbol{\theta}}_{n-1}^T \tilde{\boldsymbol{\theta}}_{n-1}). \quad (\text{B.1})$$

$$\mathbf{K} = \begin{bmatrix} k_P & 0 & 0 & 0 \\ 0 & k_T & 0 & 0 \\ 0 & 0 & k_X & 0 \\ 0 & 0 & 0 & k_S \end{bmatrix}, \quad (\text{A.5})$$

where \mathbf{K} is the controller parameter matrix and $k_P, k_T, k_X, k_S \in (0, 1)$. The Monte Carlo Randomized Algorithm was applied to select the controller parameter matrix. This procedure does not ensure reaching the global optimum. Nevertheless, based on Theorem 2 of [30], a large number of trials were made in order to guarantee a large confidence interval. Then, (A.3) can be rewritten as

$$\begin{bmatrix} \frac{P_n}{V_n} \\ \frac{T_n}{V_n} \\ \frac{X_n}{V_n} \\ \frac{20 - S_n}{V_n} \end{bmatrix} u_n = \begin{bmatrix} \chi(T_n - P_n) - \frac{P_{\text{ref},n} - k_P(P_{\text{ref},n} - P_n) - P_n}{\Delta t} \\ \psi X_n - \frac{T_{\text{ref},n+1} - k_T(T_{\text{ref},n} - T_n) - T_n}{\Delta t} \\ \mu X_n - \frac{X_{\text{ref},n+1} - k_X(X_{\text{ref},n} - X_n) - X_n}{\Delta t} \\ 7.3\mu X_n + \frac{S_{\text{ref},n+1} - k_S(S_{\text{ref},n} - S_n) - S_n}{\Delta t} \end{bmatrix}. \quad (\text{A.6})$$

Equation (A.6) has an exact solution when the culture glucose concentration is adopted as a ‘‘sacrificed variable,’’ S_{ez} [30]. Then, the control action u_n can be obtained through the least squares procedure:

The difference in discrete time ΔV_m is adopted as follows:

$$\begin{aligned} \Delta V_m &= (\tilde{\mathbf{x}}_{n+1}^T \Lambda \tilde{\mathbf{x}}_{n+1} - \tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n) \\ &+ \gamma^{-1} \text{tr}(\tilde{\boldsymbol{\theta}}_n^T \tilde{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}_{n-1}^T \tilde{\boldsymbol{\theta}}_{n-1}). \end{aligned} \quad (\text{B.2})$$

And $\Delta \boldsymbol{\theta}_i$ is defined as

$$\Delta \boldsymbol{\theta}_i = \gamma_1^{-1} \text{tr}(\tilde{\boldsymbol{\theta}}_n^T \tilde{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}_{(n-1)}^T \tilde{\boldsymbol{\theta}}_{(n-1)}). \quad (\text{B.3})$$

By replacing (25) into (B.4), ΔV_m is written as

$$\begin{aligned}\Delta V_m &= (\tilde{\mathbf{x}}_{n+1}^T \Lambda \tilde{\mathbf{x}}_{n+1} - \tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n) + \Delta \boldsymbol{\theta} \\ &= ((\tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n)^T \Lambda (\tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n) - \tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n) + \Delta \boldsymbol{\theta} \quad (\text{B.4}) \\ &= \dots = 2\tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n + \Delta \boldsymbol{\theta}.\end{aligned}$$

By replacing (25) into (B.4), ΔV_m is written as

$$\begin{aligned}\Delta V_m &= 2\tilde{\mathbf{x}}_n^T \Lambda (\tilde{\mathbf{x}}_{n+1} - \tilde{\mathbf{x}}_n) + \Delta \tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n + \Delta \boldsymbol{\theta} \\ &= 2\tilde{\mathbf{x}}_n^T \Lambda (\tilde{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) + \boldsymbol{\varepsilon}_n - \tilde{\mathbf{x}}_n) + \Delta \tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n \quad (\text{B.5}) \\ &\quad + \Delta \boldsymbol{\theta}.\end{aligned}$$

From (B.3) and rewriting $\Delta \boldsymbol{\theta}$, one obtains

$$\begin{aligned}\Delta \boldsymbol{\theta} &= \gamma^{-1} \text{tr} \left(\tilde{\boldsymbol{\theta}}_n^T \tilde{\boldsymbol{\theta}}_n - [\tilde{\boldsymbol{\theta}}_n - \Delta \tilde{\boldsymbol{\theta}}_n]^T [\tilde{\boldsymbol{\theta}}_n - \Delta \tilde{\boldsymbol{\theta}}_n] \right), \quad (\text{B.6}) \\ \Delta \boldsymbol{\theta} &= 2\gamma^{-1} \text{tr} \left(\tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n \right) - \gamma^{-1} \text{tr} \left(\Delta \tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n \right).\end{aligned}$$

Then, (B.6) is replaced into (B.5) and, after reorganizing some terms, one obtains

$$\begin{aligned}\Delta V_m &= -2\tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \Lambda \tilde{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{u}_n) + \Delta \tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n \\ &\quad + 2\tilde{\mathbf{x}}_n^T \Lambda \boldsymbol{\varepsilon}_n + \Delta \boldsymbol{\theta} \\ &= -2\tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \Lambda \boldsymbol{\varepsilon}_n \quad (\text{B.7}) \\ &\quad + \text{tr} \left(2\tilde{\boldsymbol{\theta}}_n^T [\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \Lambda + \gamma^{-1} \Delta \tilde{\boldsymbol{\theta}}_n] \right) \\ &\quad - \gamma^{-1} \text{tr} \left(\Delta \tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n \right).\end{aligned}$$

Replacing the adjustment laws (see (27)), $\Delta \tilde{\boldsymbol{\theta}}$, in (B.7), ΔV_m is represented by

$$\begin{aligned}\Delta V_m &= -2\tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n^T \Lambda \Delta \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \Lambda \boldsymbol{\varepsilon}_n \\ &\quad - \gamma^{-1} \text{tr} \left(\Delta \tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n \right).\end{aligned} \quad (\text{B.8})$$

In (B.8), the increment of the model error of $\Delta \tilde{\mathbf{x}}$ is unknown, and it can be approximated by the following equation:

$$\Delta \tilde{\mathbf{x}}_{ni} = \left(\frac{\partial \tilde{\mathbf{x}}_{(n+1)i}}{\partial \tilde{\boldsymbol{\theta}}_{ni}} \right)^T \Delta \tilde{\boldsymbol{\theta}}_{ni}. \quad (\text{B.9})$$

The partial derivative of the model error depends only on the weights of the neural network and can be rewritten as

$$\Delta \tilde{\mathbf{x}}_{ni} = - \left(\frac{\partial \tilde{\mathbf{x}}_{(n+1)i}}{\partial \tilde{\boldsymbol{\theta}}_{ni}} \right)^T \Delta \tilde{\boldsymbol{\theta}}_{ni}. \quad (\text{B.10})$$

Changing the values of the weights according to (B.6) and considering that $\boldsymbol{\theta}^*$ is constant, ideal weight vectors are

required only for analytical purpose. Now reorganizing $\Delta \tilde{\boldsymbol{\theta}}_{ni}$, it can be rewritten as

$$\begin{aligned}\Delta \tilde{\boldsymbol{\theta}}_{ni} &= \tilde{\boldsymbol{\theta}}_{ni} - \tilde{\boldsymbol{\theta}}_{(n-1)i} = (\boldsymbol{\theta}_i^* - \hat{\boldsymbol{\theta}}_{ni}) - (\boldsymbol{\theta}_i^* - \hat{\boldsymbol{\theta}}_{(n-1)i}) \\ &= -\hat{\boldsymbol{\theta}}_{ni} + \hat{\boldsymbol{\theta}}_{(n-1)i} = -\Delta \hat{\boldsymbol{\theta}}_{ni}.\end{aligned} \quad (\text{B.11})$$

Equation (B.10) can be written as

$$\Delta \tilde{\mathbf{x}}_{ni} = - \left(\frac{\partial \tilde{\mathbf{x}}_{(n+1)i}}{\partial \hat{\boldsymbol{\theta}}_{ni}} \right)^T (\gamma \tilde{\mathbf{x}}_{ni} \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)). \quad (\text{B.12})$$

Considering the value of the partial derivatives of the neural network (see (27)) and by substitution in (B.10), it yields

$$\Delta \tilde{\mathbf{x}}_{ni} = -\gamma \tilde{\mathbf{x}}_{ni} \boldsymbol{\xi}^T(\mathbf{x}_n, \mathbf{v}_n) \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n). \quad (\text{B.13})$$

The increase in the model error is defined as

$$\Delta \tilde{\mathbf{x}}_{ni} \leq \gamma |\tilde{\mathbf{x}}_{ni}|, \quad (\text{B.14})$$

where

$$\gamma = \gamma \|\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)\|^2, \quad (\text{B.15})$$

$$\max(\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)) \leq 1.$$

Consider that $\|\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)\|$ is a bounded function. The value γ is a learning factor of the neural network ($0 < \gamma < 1$) and it can be arbitrarily defined.

Expressing (B.14) in vectorial form,

$$\|\Delta \tilde{\mathbf{x}}_n\| \leq \gamma \|\tilde{\mathbf{x}}_n\|. \quad (\text{B.16})$$

Now, substituting the increment value of the model error in (B.16), the Lyapunov discrete difference is defined as

$$\begin{aligned}\Delta V_m &= \tilde{\mathbf{x}}_n^T \gamma^2 \Lambda \tilde{\mathbf{x}}_n - 2\tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \Lambda \boldsymbol{\varepsilon}_n \\ &\quad - \gamma^{-1} \text{tr} \left(\Delta \tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n \right),\end{aligned} \quad (\text{B.17})$$

$$\begin{aligned}\Delta V_m &= -\tilde{\mathbf{x}}_n^T (\Lambda (1 - \gamma^2)) \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \Lambda \boldsymbol{\varepsilon}_n \\ &\quad - \gamma^{-1} \text{tr} \left(\Delta \tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n \right).\end{aligned}$$

Now replacing the learning rule in (B.17) and applying norm yield

$$\begin{aligned}\Delta V_m &= -\tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + \gamma^2 \tilde{\mathbf{x}}_n^T \Lambda \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \Lambda \boldsymbol{\varepsilon}_n \\ &\quad - \gamma^{-1} \text{tr} \left((\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \Lambda)^T (\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \Lambda) \right), \\ \Delta V_m &\leq -\|\Lambda\| \|\tilde{\mathbf{x}}_n\|^2 + \gamma^2 \|\Lambda\| \|\tilde{\mathbf{x}}_n\|^2 + 2\lambda_{\max}(\Lambda) \|\tilde{\mathbf{x}}_n\| \bar{\varepsilon} \\ &\quad - \gamma^2 \|\Lambda\| \|\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)\|^2 \|\tilde{\mathbf{x}}_n\|^2,\end{aligned} \quad (\text{B.18})$$

$$\Delta V_m \leq \|\Lambda\| \|\tilde{\mathbf{x}}_n\|^2 + 2\|\Lambda\| \|\tilde{\mathbf{x}}_n\| \bar{\varepsilon},$$

where $\|\xi(\mathbf{x}_n, \mathbf{v}_n)\| \leq 1$; then (B.18) can be expressed as

$$\begin{aligned} \Delta V_m &\leq -\|\Lambda\| \|\tilde{\mathbf{x}}_n\|^2 + 2\|\Lambda\| \|\tilde{\mathbf{x}}_n\| \bar{\varepsilon} \\ &= -\lambda_{\min}(\Lambda) (\|\tilde{\mathbf{x}}_n\| - 2\bar{\varepsilon}) \|\tilde{\mathbf{x}}_n\|. \end{aligned} \quad (\text{B.19})$$

From (B.19), it follows that when $\varepsilon = 0$, x_n tends to zero when n tends to infinity. If the error norm is $\|\tilde{\mathbf{x}}_n\| < 2\bar{\varepsilon}$, we can obtain that $V > 0$ and the neural weights could tend to infinity. To prevent the above situation, the next update rule is used.

$$\Delta\theta = \begin{cases} \gamma \xi(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \Lambda & \text{if } \|\tilde{\mathbf{x}}_n\| \geq \bar{\mathbf{x}}_0, \|\theta\| \leq M \\ 0 & \text{somewhere else,} \end{cases} \quad (\text{B.20})$$

where $\bar{\mathbf{x}}_0$ and M are design parameters. \square

Nomenclature

Symbol

\mathbf{c}_i :	Vector of centers of each RBF function, —
dB :	Standard Brownian motion, h^{-1}
\mathbf{e}_n :	Tracking error, $\text{g}^2\text{L}^{-2}\text{h}^{-1}$
\mathbf{E} :	Additive uncertainty, gL^{-1}
f_{\max} :	System bandwidth, h^{-1}
J :	Performance index, $\text{gL}^{-1}\text{h}^{-1}$
k, K :	Controller parameters, —
M :	Neural design parameter, —
m :	Maximum number of neurons, —
N_{ref} :	Number of points of the reference profile, —
P :	Secreted protein concentration, gL^{-1}
r :	Root of a linear system, —
S :	Glucose concentration, gL^{-1}
S_F :	Glucose concentration of the feed, gL^{-1}
T :	Total protein concentration, gL^{-1}
t_c :	Correlation time, h
Δt :	Sample time, h
Δt_{\max} :	Maximum sample time, h
u :	Feed flow rate, Lh^{-1}
U :	Integral of the tracking error, g^2L^{-2}
V :	Bioreactor volume, L
V_m :	Lyapunov candidate function, —
X :	Biomass, gL^{-1}
$\hat{\mathbf{x}}_n$:	Estimated variables vector $(\hat{P}_n, \hat{T}_n, \hat{X}_n)$, gL^{-1}
$\tilde{\mathbf{x}}_0$:	Neural design parameter, —
$Y_{S/X}$:	Yield of glucose per cell mass, —
z :	State vector, gL^{-1} .

Subscripts

ez :	Sacrificed variable, —
n :	Actual sample time, —
ref:	Reference profile, —
0:	Initial condition, —

Greek Letters

α :	Intensity of the noisy signal, —
λ, δ, φ :	Random number between 0 and 1, —

Λ :	Diagonal definite positive matrix, —
$\xi(\cdot)$:	RBFs vector, —
ε_n :	Neural approximation error, —
σ_i :	Vector of widths of each RBF function, —
θ_n :	Neural weight matrix, —
θ^* :	Optimal weight matrix, —
$\tilde{\theta}_n$:	Neural error weight matrix, —
$\Delta\theta_i$:	Auxiliary vector, —
μ :	Specific growth rate, h^{-1}
μ_{\max} :	Maximum specific growth rate, h^{-1}
ψ :	Protein expression rate, —
χ :	Protein secretion rate, —
\mathbf{v}_n :	Input vector to the neural estimator $[u_n, S_n, V_n]$, —
γ :	Learning factor of the neural network, —.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

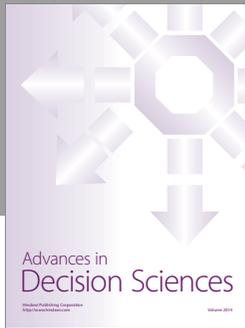
This work was partially funded by the following Argentine institutions: National Council of Scientific and Technological Research (CONICET), Universidad Nacional de San Juan (UNSJ), and Universidad Tecnológica Nacional.

References

- [1] N. Dimitrova and M. Krastanov, "Nonlinear adaptive control of a model of an uncertain fermentation process," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 9, pp. 1001–1009, 2010.
- [2] L. M. Beltrán, C. L. Garzon-Castro, F. Garcés, and M. Moreno, "Monitoring and control system used in microalgae crop," *IEEE Latin America Transactions*, vol. 10, no. 4, pp. 1993–1998, 2012.
- [3] M. Cárcamo, P. A. Saa, J. Torres et al., "Effective dissolved oxygen control strategy for high-cell-density cultures," *IEEE Latin America Transactions*, vol. 12, no. 3, pp. 389–394, 2014.
- [4] K. C. Veluvolu, Y. C. Soh, and W. Cao, "Robust discrete-time nonlinear sliding mode state estimation of uncertain nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 9, pp. 803–828, 2007.
- [5] S. Rómoli, A. N. Amicarelli, O. A. Ortiz, G. J. E. Scaglia, and F. di Sciascio, "Nonlinear control of the dissolved oxygen concentration integrated with a biomass estimator for production of *Bacillus thuringiensis* δ -endotoxins," *Computers and Chemical Engineering*, vol. 93, pp. 13–24, 2016.
- [6] S. Gnath, M. Jenzsch, R. Simutis, and A. Lübbert, "Control of cultivation processes for recombinant protein production: a review," *Bioprocess and Biosystems Engineering*, vol. 31, no. 1, pp. 21–39, 2008.
- [7] S. Tokat, "Sliding mode controlled bioreactor using a time-varying sliding surface," *Transactions of the Institute of Measurement & Control*, vol. 31, no. 5, pp. 435–456, 2009.
- [8] T. Egli, "Growth kinetics, bacterial," in *Encyclopedia of Microbiology*, M. Schaechter, Ed., pp. 180–193, Academic Press, Oxford, UK, 3rd edition, 2009.
- [9] S. C. Patwardhan, S. Narasimhan, P. Jagadeesan, B. Gopaluni, and S. L. Shah, "Nonlinear Bayesian state estimation: a review

- of recent developments,” *Control Engineering Practice*, vol. 20, no. 10, pp. 933–953, 2012.
- [10] H. O. Méndez-Acosta, D. U. Campos-Delgado, R. Femat, and V. González-Alvarez, “A robust feedforward/feedback control for an anaerobic digester,” *Computers and Chemical Engineering*, vol. 29, no. 7, pp. 1613–1623, 2005.
- [11] J. Alvarez-Ramírez, “Adaptive control of feedback linearizable systems: a modelling error compensation approach,” *International Journal of Robust and Nonlinear Control*, vol. 9, no. 6, pp. 361–377, 1999.
- [12] S. Tong, C. Liu, and Y. Li, “Fuzzy-adaptive decentralized output-feedback control for large-scale nonlinear systems with dynamical uncertainties,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 845–861, 2010.
- [13] Z. Chen and T. Zhang, “Dynamics of a stochastic model for continuous flow bioreactor with Contois growth rate,” *Journal of Mathematical Chemistry*, vol. 51, no. 3, pp. 1076–1091, 2013.
- [14] J. Hess and O. Bernard, “Design and study of a risk management criterion for an unstable anaerobic wastewater treatment process,” *Journal of Process Control*, vol. 18, no. 1, pp. 71–79, 2008.
- [15] G. Lara-Cisneros, R. Femat, and E. a. Pérez, “On dynamical behaviour of two-dimensional biological reactors,” *International Journal of Systems Science. Principles and Applications of Systems and Integration*, vol. 43, no. 3, pp. 526–534, 2012.
- [16] H. L. Smith and P. Waltman, *The Theory of the Chemostat: Dynamics of Microbial Competition*, vol. 13, Cambridge University Press, 1995.
- [17] A. Rincón, D. Piarpuzán, and F. Angulo, “A new adaptive controller for bio-reactors with unknown kinetics and biomass concentration: guarantees for the boundedness and convergence properties,” *Mathematics and Computers in Simulation*, vol. 112, pp. 1–13, 2015.
- [18] K. J. Keesman and V. I. Maksimov, “On feedback identification of unknown characteristics: a bioreactor case study,” *International Journal of Control*, vol. 81, no. 1, pp. 134–145, 2008.
- [19] A. E. Rodríguez, J. A. T. Muñoz, R. Luna et al., “Robust control for cultivation of microorganisms in a high density fed-batch bioreactor,” *IEEE Latin America Transactions*, vol. 13, no. 6, pp. 1927–1933, 2015.
- [20] L. Mailleret, O. Bernard, and J.-P. Steyer, “Nonlinear adaptive control for bioreactors with unknown kinetics,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 40, no. 8, pp. 1379–1385, 2004.
- [21] C. Bartolomeo and G. Mosè, “Type-2 fuzzy control of a bioreactor,” in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent System (ICIS '09)*, pp. 700–704, November 2009.
- [22] G. Lara-Cisneros, R. Femat, and D. Dochain, “An extremum seeking approach via variable-structure control for fed-batch bioreactors with uncertain growth rate,” *Journal of Process Control*, vol. 24, no. 5, pp. 663–671, 2014.
- [23] M. Guay, D. Dochain, and M. Perrier, “Adaptive extremum seeking control of continuous stirred tank bioreactors with unknown growth kinetics,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 40, no. 5, pp. 881–888 (2005), 2004.
- [24] C. Cimander and C.-F. Mandenius, “Bioprocess control from a multivariate process trajectory,” *Bioprocess and Biosystems Engineering*, vol. 26, no. 6, pp. 401–411, 2004.
- [25] F. Mairet and J.-L. Gouzé, “Hybrid control of a bioreactor with quantized measurements,” *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 61, no. 5, pp. 1385–1390, 2016.
- [26] F. Liping, Z. Jun, H. Xing, and H. Dong, “The design of the MISO model predictive controller for bioreactor,” *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 10, no. 6, 2012.
- [27] M. Elenchezhiyan and J. Prakash, “State estimation of stochastic non-linear hybrid dynamic system using an interacting multiple model algorithm,” *ISA Transactions*, vol. 58, pp. 520–532, 2015.
- [28] G. Mannina, G. Di Bella, and G. Viviani, “Uncertainty assessment of a membrane bioreactor model using the GLUE methodology,” *Biochemical Engineering Journal*, vol. 52, no. 2–3, pp. 263–275, 2010.
- [29] J.-P. Vila and V. Wagner, “Predictive neuro-control of uncertain systems: design and use of a neuro-optimizer,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 39, no. 5, pp. 767–777, 2003.
- [30] S. Rómoli, M. E. Serrano, O. A. Ortiz, Vega J. R., and G. J. E. Scaglia, “Tracking control of concentration profiles in a fed-batch bioreactor using a linear algebra methodology,” *ISA Transactions*, vol. 57, no. 0, pp. 162–171, 2015.
- [31] S. Park and W. F. Ramirez, “Optimal production of secreted protein in fed-batch reactors,” *AIChE Journal*, vol. 34, no. 9, pp. 1550–1558, 1988.
- [32] X. Chen, W. Du, H. Tianfield, R. Qi, W. He, and F. Qian, “Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1289–1299, 2014.
- [33] W. Gong, J. Liang, X. Kan, L. Wang, and A. M. Dobaie, “Robust state estimation for stochastic complex-valued neural networks with sampled-data,” *Neural Computing and Applications*, 2017.
- [34] S. Romoli, G. J. E. Scaglia, M. E. Serrano, S. A. Godoy, O. A. Ortiz, and J. R. Vega, “Control of a fed-batch fermenter based on a linear algebra strategy,” *IEEE Latin America Transactions*, vol. 12, no. 7, pp. 1206–1213, 2014.
- [35] J. M. Ali, N. H. Hoang, M. A. Hussain, and D. Dochain, “Review and classification of recent observers applied in chemical process systems,” *Computers and Chemical Engineering*, vol. 76, pp. 27–41, 2015.
- [36] H. Michalska and D. Q. Mayne, “Robust receding horizon control of constrained nonlinear systems,” *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, 1993.
- [37] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: stability and optimality,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 36, no. 6, pp. 789–814, 2000.
- [38] M. E. Serrano, S. A. Godoy, S. Rómoli, and G. J. E. Scaglia, “A numerical approximation-based controller for mobile robots with velocity limitation,” *Asian Journal of Control*, 2017.
- [39] S. G. Kulkarni, A. K. Chaudhary, S. Nandi, S. S. Tambe, and B. D. Kulkarni, “Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN),” *Biochemical Engineering Journal*, vol. 18, no. 3, pp. 193–210, 2004.
- [40] M. Von Stosch, R. Oliveira, J. Peres, and S. Feyer De Azevedo, “A novel identification method for hybrid (N)PLS dynamical systems with application to bioprocesses,” *Expert Systems with Applications*, vol. 38, no. 9, pp. 10862–10874, 2011.

- [41] Z. Xiong and J. Zhang, "Modelling and optimal control of fed-batch processes using a novel control affine feedforward neural network," *Neurocomputing*, vol. 61, no. 1-4, pp. 317–337, 2004.
- [42] F. G. Rossomando and C. M. Soria, "Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural PID," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1179–1191, 2015.
- [43] R. F. Escobar, C. M. Astorga-Zaragoza, A. C. Tllez-Anguiano, D. Jurez-Romero, J. A. Hernandez, and G. V. Guerrero-Ramrez, "Sensor fault detection and isolation via high-gain observers: application to a double-pipe heat exchanger," *ISA Transactions*, vol. 50, no. 3, pp. 480–486, 2011.
- [44] Matlab™, *User Guide, 7.6.0.324*, The MathWorks, Inc, Torrance, CA, USA, 2008.
- [45] M. P. Aghababa, "Adaptive control of nonlinear complex Holling II predator-prey system with unknown parameters," *Complexity*, vol. 21, no. 6, pp. 260–266, 2016.
- [46] R. Tempo and H. Ishii, "Monte Carlo and Las Vegas randomized algorithms for systems and control," *European Journal of Control*, vol. 13, no. 2-3, pp. 189–203, 2007.
- [47] A. Noruzi, T. Banki, O. Abedinia, and N. Ghadimi, "A new method for probabilistic assessments in power systems, combining Monte Carlo and stochastic-algebraic methods," *Complexity*, vol. 21, no. 2, pp. 100–110, 2015.
- [48] H. Hao, K. Zhang, S. X. Ding, Z. Chen, and Y. Lei, "A data-driven multiplicative fault diagnosis approach for automation processes," *ISA Transactions*, vol. 53, no. 5, pp. 1436–1445, 2014.
- [49] C. D. Karlgaard and H. Shen, "Robust state estimation using desensitized divided difference filter," *ISA Transactions*, vol. 52, no. 5, pp. 629–637, 2013.
- [50] K. Michail, A. C. Zolotas, and R. M. Goodall, "Optimised sensor selection for control and fault tolerance of electromagnetic suspension systems: a robust loop shaping approach," *ISA Transactions*, vol. 53, no. 1, pp. 97–109, 2014.
- [51] M.-C. Pai, "Robust tracking and model following for uncertain time-delay systems with input nonlinearity," *Complexity*, vol. 21, no. 2, pp. 66–73, 2015.
- [52] Z. Chen and T. Zhang, "Long time behaviour of a stochastic model for continuous flow bioreactor," *Journal of Mathematical Chemistry*, vol. 51, no. 2, pp. 451–464, 2013.
- [53] C. Xu and S. Yuan, "An analogue of break-even concentration in a simple stochastic chemostat model," *Applied Mathematics Letters. An International Journal of Rapid Publication*, vol. 48, pp. 62–68, 2015.
- [54] MathWorks, 2017, <http://www.mathworks.com/help/simulink/slref/bandlimitedwhitenoise.html>.
- [55] V. Rajinikanth and K. Latha, "Tuning and retuning of PID controller for unstable systems using evolutionary algorithm," *ISRN Chemical Engineering*, vol. 2012, Article ID 693545, 11 pages, 2012.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

