

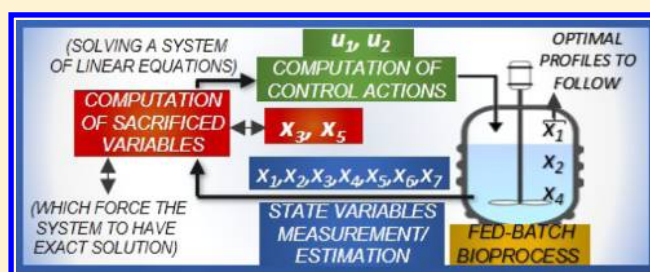
# Multivariable Control for Tracking Optimal Profiles in a Nonlinear Fed-Batch Bioprocess Integrated with State Estimation

María N. Pantano,<sup>\*,†</sup> Mario E. Serrano,<sup>†</sup> María C. Fernández,<sup>†</sup> Francisco G. Rossomando,<sup>‡</sup> Oscar A. Ortiz,<sup>†</sup> and Gustavo J. E. Scaglia<sup>†</sup>

<sup>†</sup>Instituto de Ingeniería Química, Universidad Nacional de San Juan (UNSJ), CONICET, Av. Lib. San Martín Oeste 1109, San Juan J5400ARL, Argentina

<sup>‡</sup>Instituto de Automática, Facultad de Ingeniería, Universidad Nacional de San Juan (UNSJ), CONICET, Av. Lib. San Martín Oeste 1109, San Juan J5400ARL, Argentina

**ABSTRACT:** This paper aims to solve the problem of tracking optimal profiles for a nonlinear multivariable fed-batch bioprocess by a simple but efficient closed-loop control technique based on a linear algebra approach. In the proposed methodology, the control actions are obtained by solving a system of linear equations without the need for state transformations. The optimal profiles to follow are directly those corresponding to output desired variables, therefore, estimation of states for nonmeasurable variables is considered by employing a neural networks method. The efficiency of the proposed controller is tested through several simulations, including process disturbances and operation under parametric uncertainty. The optimal controller parameters are selected through the Montecarlo Randomized Algorithm. In addition, proof of convergence to zero of tracking errors is analyzed and included in this article.



## INTRODUCTION

Many bioprocesses, such as production of proteins, monoclonal antibodies, and other therapeutic drugs, either cannot be produced or are difficult to obtain by conventional manufacturing methods. Most of these are classified as low-volume/high-value products, and there is a tremendous economic potential in this sector of the market.<sup>1</sup> However, in these processes, the more convenient operation is in fed-batch reactors since greater flexibility is provided in terms of product optimization,<sup>2</sup> especially when changing nutrient concentrations affect the productivity and yield of the desired product.<sup>3</sup>

To increase the productivity of the fed-batch bioprocesses, many efforts have been devoted on both modeling and estimation as well as optimization of fed-batch fermentation systems. In this way, mathematical modeling and analysis have become fundamental tools in optimally designing and operating production facilities in the biotechnology industry. Obviously, once the optimal operating policies have been found, the next stage is to implement them, that is, the closed loop control problem to ensure the tracking of the desired values.<sup>4</sup>

From the control engineer's viewpoint, the fed-batch processes present the greatest challenge: the process model usually contains strongly time-varying parameters, changes in initial conditions, input saturation, external disturbances, and the stiffness and nonlinearity of the model equations.<sup>2,5–8</sup> Several control techniques are studied today associated with optimization and control of a bioprocess, such as bioinspired algorithms,<sup>9</sup> genetic algorithms,<sup>10,11</sup> robust control,<sup>6,12</sup> nonlinear fuzzy control,<sup>13</sup> evolutionary algorithms,<sup>14</sup> model predictive control

(MPC)<sup>15</sup> and nonlinear MPC,<sup>16,17</sup> adaptive stochastic algorithms,<sup>18</sup> and neural network model,<sup>19,20</sup> etc. Much of the control literature for fed-batch cultures focuses on open-loop operation owing to the highly nonlinear and inherently difficult dynamic behavior.<sup>21</sup> These methods have good results in biological processes; however, they have limitations regarding the need for advanced specific knowledge, the difficulty of mathematical processing (especially in nonlinear systems), trouble with real-time implementation, and the need for a complicated database of the processes.<sup>22</sup> Besides that, in the open-loop control strategies, the main disadvantage is that no compensation is made for modeling mismatch or random disturbances during the process operation.<sup>1,7,23,24</sup> It is therefore important to design a controller to track the optimal policy for desired variables considering disturbance compensation for the closed-loop control problem. Its noteworthy that there is not much background data in the literature for the design of a closed-loop controller following directly the optimal output profiles achieved.

The aim of this work is to solve the problem of tracking three optimal output profiles with two control actions through a closed-loop control integrated with state estimation of an important nonlinear bioprocess. To achieve the stated objective, the proposed methodology is based on solving a system of linear equations. The simple approach of this methodology suggests

Received: February 26, 2017

Revised: April 25, 2017

Accepted: May 1, 2017

Published: May 1, 2017

that, knowing the values of the desired variables and analyzing the conditions for the system to have an exact solution, a system of linear equations is solved to find the control actions that force the process to get the desired state.

The main advantages of this method are its simplicity, versatility and accuracy even under parametric uncertainty and process disturbances. The methodology for the controller design is simple, it can be implemented with only a basic knowledge of linear algebra.

The main contributions of this work are the proposal of a control technique based on linear algebra approach, which is simple since it does not need to perform state transformations. In each sampling period, a system of linear equations is solved to calculate the control actions. By analyzing the conditions for the system to have an exact solution, the trajectories for the remaining states are obtained. Besides that, proof of convergence to zero of tracking errors is included in this article. Others contributions of this paper are the tuning procedure proposed for the controller parameters that uses a Montecarlo Experiment (MCE) in order to minimize a proposed cost index. Additionally, to implement the proposed control technique, it is necessary to have a good state estimation to feedback all control variables, therefore, a states estimation for unmeasurable output variables of the process by employing neural networks is developed herein.

The case study proposed for control, which has been well documented in the literature, is the Lee–Ramirez fed-batch bioreactor. Lee and Ramirez<sup>25</sup> proposed a mathematical model for the induced foreign protein production by recombinant bacteria in a fed-batch bioreactor. The advantage of using this system is that it has been already used by a number of researchers using different techniques, so the available data can be used to assess other methods.

The paper is organized as follows. In section 2, the mathematical model of the proposed system is presented. Then, the controller design theory is developed in section 3. The state estimation methodology by using neural networks is exposed in section 4. In section 5, the results and discussion of the simulation tests including normal conditions operation, perturbations in control actions and parametric uncertainty, are shown. Finally, section 6 outlines the conclusions of the work.

## 2. BIOPROCESS MODEL

Lee and Ramirez<sup>25</sup> developed a model describing the dynamics of the process of induced foreign protein production by recombinant bacteria. Then, they used it to obtain an optimal control policy to maximize the foreign protein production with a nutrient and inducer feeding strategy.<sup>26</sup> Carrasco and Banga<sup>18</sup> used adaptive stochastic algorithms to obtain better results. Since the performance index exhibits a very low sensitivity with respect to the controls, Tholudur and Ramirez<sup>27</sup> constructed a modified parameter function set to increase the sensitivity to the controls. Balsa-Canto used the same parameter function set.<sup>28</sup> A genetic algorithm to optimize the same system considering multiple control variables was presented in ref 10. The mathematical model used here is taken from Tholudur and Ramirez.<sup>27</sup> Although simple, it can effectively describe the dynamics of the bioprocess. It is important to note that, in this work, the desired trajectories to track are directly the optimal output desired profiles. These trajectories were obtained by an open-loop simulation of the bioprocess using the optimal feeding policies achieved by Balsa-Canto et al.<sup>28</sup>

The operation of the fed-batch bioreactor considering two control variables (nutrient and inducer feed rates) is described by seven differential equations (eq 1).

$$\left. \begin{aligned} \dot{x}_1 &= u_1 + u_2 \\ \dot{x}_2 &= x_2\mu - \frac{u_1 + u_2}{x_1}x_2 \\ \dot{x}_3 &= \frac{u_1N}{x_1} - \frac{u_1 + u_2}{x_1}x_3 - \frac{\mu}{Y}x_2 \\ \dot{x}_4 &= x_2R - \frac{u_1 + u_2}{x_1}x_4 \\ \dot{x}_5 &= \frac{u_2I}{x_1} - \frac{u_1 + u_2}{x_1}x_5 \\ \dot{x}_6 &= -K_1x_6 \\ \dot{x}_7 &= K_2(1 - x_7) \end{aligned} \right\} \quad (1)$$

where

$$\mu = \left( \frac{\mu_{\max}x_3}{K_{CN} + x_3\left(1 + \frac{x_3}{K_S}\right)} \right) \left( x_6 + x_7 \frac{K_{CI}}{K_{CI} + x_5} \right) \quad (2)$$

$$R = \left( \frac{f_{\max}^0x_3}{K_{CN} + x_3\left(1 + \frac{x_3}{K_S}\right)} \right) \left( \frac{f_1^0 + x_5}{K_1 + x_5} \right) \quad (3)$$

$$K_1 = K_2 = \frac{k_{11}x_5}{K_{IX} + x_5} \quad (4)$$

The state variables are the volume of bioreactor  $x_1$  (L), cell density  $x_2$  (g/L), concentration of nutrient  $x_3$  (g/L), protein concentration  $x_4$  (g/L), inducer concentration  $x_5$  (g/L), inducer shock factor on the cell growth rate  $x_6$ , and the inducer recovery factor on the cell growth rate  $x_7$  (both dimensionless). The model parameters set was validated by several simulations and is experiments, which were developed by Lee and Ramirez.<sup>29</sup> The feed nutrient concentration stream is  $N$  (g/L),  $I$  (g/L) is the concentration of inducer in the inducer feed stream, and  $Y$  is the growth yield coefficient. In addition,  $\mu$  is the specific growth rate, and  $R$  is the foreign protein production rate. The two control actions used in the proposed controller are the glucose feeding rate,  $u_1$  (L/h), and inducer feeding rate  $u_2$  (L/h) to the fed-batch bioreactor. The three desired (controlled) variables to follow are the reactor volume  $x_1$ , the cell density  $x_2$ , and the foreign protein concentration  $x_4$ .

## 3. CONTROLLER DESIGN

“Most control structures are based on the use of a design model. A mathematical model provides a map from inputs to responses. The quality of a model depends on how closely its responses match those of the true plant. Therefore, a model set which includes the true physical plant can never be constructed”.<sup>30</sup> This is one of the reasons why it is necessary that the design of a controller allows tracking, with minimal error, of previously determined profiles (calculated with a model and specific initial conditions) even in the presence of perturbations and parametric uncertainty. The proposed controller methodology is based on approximating the differential equations of the mathematical model (1) through the Euler method; although simple, it works

very well. Hence, the control problem for tracking optimum profiles of volume ( $x_1$ ), cell density ( $x_2$ ), and protein concentration ( $x_4$ ) is reduced to the resolution of a system of linear equations. To achieve the control goal, the feed flow rates of nutrient ( $u_1$ ) and inducer ( $u_2$ ) are available to be used as control actions. Therefore, the aim is to find the values of  $u_1$  and  $u_2$  such that the variables  $x_1$ ,  $x_2$ , and  $x_4$  follow paths desired with minimal tracking error.

**Controller Methodology.** The first step for this technique, is to rearrange the system of eqs 1 in matrix form:

$$\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ (N - x_3) & -x_3 \\ -1 & -1 \\ -1 & (I - x_5) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \frac{x_1}{x_2} - x_1 g \\ \dot{x}_3 x_1 + \frac{\mu}{Y} x_2 x_1 \\ \frac{\dot{x}_4 x_1}{x_4} - \frac{R x_2 x_1}{x_4} \\ \dot{x}_5 x_1 \end{pmatrix} \quad (5)$$

*Remark 1.* The variables  $x_6$  and  $x_7$  are not directly related to the control variables, therefore, are not considered in the controller design.

According to the rule of numerical integration of Euler, the differential equations can be approximated as follow:

$$\dot{x}_i = \frac{x_{i,n+1} - x_{i,n}}{T_s} \quad (6)$$

where  $T_s$  is the sample time,  $x_{i,n}$  represents the state variable  $i$  in  $n$  instant and  $x_{i,n+1}$  the state variable  $i$  in  $(n + 1)$  instant. Defining the following expression:

$$\overbrace{x_{i,\text{ref},n+1} - x_{i,n+1}}^{e_{i,n+1}} = k_i \overbrace{(x_{i,\text{ref},n} - x_{i,n})}^{e_{i,n}} \quad (7)$$

where  $x_{i,\text{ref},n}$  and  $x_{i,\text{ref},n+1}$  are the reference values in the  $n$  instant and the next sample time respectively, the constant  $k_i$  is the controller parameter for the variable  $i$ . Then, the immediately reachable value of each state variable is

$$x_{i,n+1} = x_{i,\text{ref},n+1} - k_i \overbrace{(x_{i,\text{ref},n} - x_{i,n})}^{e_{i,n}} \quad (8)$$

Therefore, the values of the real state variables in the next sample time ( $x_{i,n+1}$ ) are function of the reference profiles, the actual state variable and the controller parameters. So, all values are known. Consequently, substituting (8) in (eq 6):

$$\dot{x}_i = \frac{\overbrace{[x_{i,\text{ref},n+1} - k_i(x_{i,\text{ref},n} - x_{i,n})]}^{x_{i,n+1}}}{T_s} - x_{i,n} \quad (9)$$

Now, replacing eq 9 in each differential expression appearing in eq 5, the process model can be rewritten, see eq 10. The values of  $x_1$ ,  $x_2$ , and  $x_4$  are the references to follow, therefore are known (the reference values as well as the real system values). Note that eq 10 is a system of five equations and two unknown variables, which normally has no solution. Therefore, the unknown variables of this system are defined as “sacrificed variables” and are written as  $x_{i,\text{ez}}$  corresponding in this case to  $x_{3,\text{ez}}$  and  $x_{5,\text{ez}}$ . The key of this technique is that the values adopted by such variables

force the equation system (10) to have an exact solution, which implies that the error be not only minimal, but also equal to zero.

$$\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ (N - x_{3,n}) & -x_{3,n} \\ -1 & -1 \\ -1 & (I - x_{5,n}) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \left( \frac{x_{1,\text{ref},n+1} - k_1(x_{1,\text{ref},n} - x_{1,n}) - x_{1,n}}{T_s} \right) \\ \left( \frac{x_{2,\text{ref},n+1} - k_2(x_{2,\text{ref},n} - x_{2,n}) - x_{2,n}}{T_s} \right) \frac{x_{1,n}}{x_{2,n}} - x_{1,n} \mu \\ \left( \frac{x_{3,\text{ez},n+1} - k_3(x_{3,\text{ez},n} - x_{3,n}) - x_{3,n}}{T_s} \right) x_{1,n} + \frac{\mu}{Y} x_{2,n} x_{1,n} \\ \left( \frac{x_{4,\text{ref},n+1} - k_4(x_{4,\text{ref},n} - x_{4,n}) - x_{4,n}}{T_s} \right) \frac{x_{1,n}}{x_{4,n}} - R x_{2,n} \frac{x_{1,n}}{x_{4,n}} \\ \left( \frac{x_{5,\text{ez},n+1} - k_5(x_{5,\text{ez},n} - x_{5,n}) - x_{5,n}}{T_s} \right) x_{1,n} \end{pmatrix} \quad (10)$$

To simplify the mathematical treatment, the equations system is expressed as follows:

$$\begin{pmatrix} a_{31} & a_{32} \\ a_{51} & a_{52} \\ a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{41} & a_{42} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} b_3 \\ b_5 \\ b_1 \\ b_2 \\ b_4 \end{pmatrix} \quad (11)$$

Note that the order of the rows are altered for the Gaussian elimination because the system presents three linearly dependent rows.

Generally, the equations system (10) has no exact solution, but to accomplish the target of this control methodology, that system must have an exact solution. Therefore, the vector  $\mathbf{b}$  must be contained in the space formed by the columns of  $\mathbf{A}$ , that is, the vector  $\mathbf{b}$  must be a linear combination of the column vectors of matrix  $\mathbf{A}$ . Then, the necessary and sufficient condition for the system to have exact solution is achieved by carrying out a Gaussian elimination process (see Appendix A), leaving:

$$\begin{aligned} 0 &= (a_{31}a_{52} - a_{32}a_{51})(a_{31}b_1 - b_3a_{11}) \\ &\quad - (a_{31}b_5 - b_3a_{51})(a_{31}a_{12} - a_{32}a_{11}) \\ 0 &= (a_{31}a_{52} - a_{32}a_{51})(a_{31}b_2 - b_3a_{21}) \\ &\quad - (a_{31}b_5 - b_3a_{51})(a_{31}a_{22} - a_{32}a_{21}) \\ 0 &= (a_{31}a_{52} - a_{32}a_{51})(a_{31}b_4 - b_3a_{41}) \\ &\quad - (a_{31}b_5 - b_3a_{51})(a_{31}a_{42} - a_{32}a_{41}) \end{aligned} \quad (12)$$

This equations system is solved for each sampling period, where the unknown variables  $x_{3,\text{ez}}$  and  $x_{5,\text{ez}}$  (sacrificed variables) are calculated. Once the values of  $x_{3,\text{ez},n+1}$  and  $x_{5,\text{ez},n+1}$  are found,

the matrix  $A$  and  $b$  are completely known at ( $n$ ) time. Therefore, the system has an exact solution and the control variables  $u_{1,n}$  and  $u_{2,n}$  ( $u$  vector) can be calculated solving the system (10) by the least-squares method:

$$(A^T A)u = A^T b \Rightarrow u = (A^T A)^{-1} A^T b \quad (13)$$

The solution allows finding the control actions ( $u_{1,n}$  and  $u_{2,n}$ ) to be applied at time  $n$  to follow the desired trajectories with a minimal error.

The following constraints on the control variables are considered:<sup>28</sup>  $0 \leq u_1 \leq 1$  and  $0 \leq u_2 \leq 1$ .

With respect to the parameters of the controller, in eq 8 note that if  $k_i = 0$ , the reference trajectory is reached in only one step. The parameters  $k_i$ ,  $i = 1, 2, 3, 4, 5$ , satisfied  $0 < k_i < 1$ , which allows the tracking error to tend to zero (demonstration in Appendix B).

The tracking error is the value of the difference between the reference and real trajectory and is calculated as

$$\|E_n\| = \sqrt{E_{1,n}^2 + E_{2,n}^2 + E_{4,n}^2}; \quad E_T = T_0 \sum_n \|E_n\| \quad (14)$$

$$E_{1,n} = \frac{x_{1,ref,n} - x_{1,n}}{x_{1,ref,max}} \quad (15)$$

$$E_{2,n} = \frac{x_{2,ref,n} - x_{2,n}}{x_{2,ref,max}} \quad (16)$$

$$E_{4,n} = \frac{x_{4,ref,n} - x_{4,n}}{x_{4,ref,max}} \quad (17)$$

where  $x_{1,ref,max} = 1,9 L$ ,  $x_{2,ref,max} = 13,92 \text{ g/L}$ , and  $x_{4,ref,max} = 3,1 \text{ g/L}$ . Note that the tracking error is dimensionless.

*Theorem 1.* If the system behavior is ruled by eq 10 and the controller is designed by eq 13, then,  $e_{i,n} \rightarrow 0$ ,  $n \rightarrow \infty$ , when profile tracking problems are considered. Proof of *Theorem 1* and the convergence to zero of tracking errors can be seen in Appendix B.

Figure 1 shows the architecture of the control system proposed herein. In this work, the optimal profiles are calculated by an

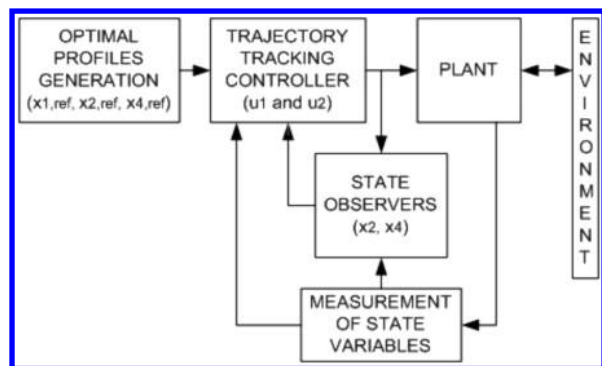


Figure 1. Architecture of the trajectory tracking controller.

open-loop simulation using the optimal policies of feeding obtained by Balsa-Canto et al.,<sup>28</sup> the focus is in the trajectory tracking of such profiles. The controller block can be better understood with Figure 2 showing a flowchart explaining how the controller computes the control actions in the process control system.

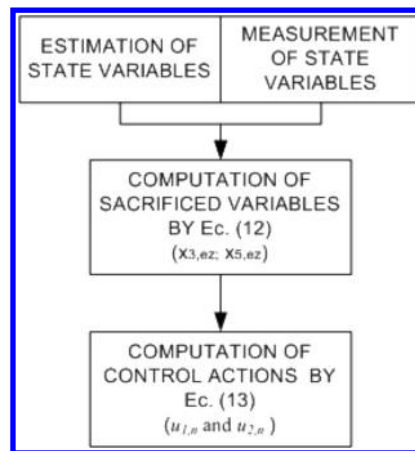


Figure 2. Flowchart of the strategy proposed.

#### 4. STATE ESTIMATION STRATEGY

The closed-loop control strategy supposes that measurements of the state variables are available. As far as the sensors hardware is normally expensive and not always available, an interesting challenge consists in estimating the state variables. At present, several techniques are studied for state and parameter estimation for chemical and biochemical processes.<sup>29,31–36</sup> In this paper, the state estimation is proposed using neural networks. So that the estimation responds well to perturbations and uncertainties, the neural networks have been trained in a range of operations, taking them into account.

In the recombinant protein production, the reactor volume, cell mass, and glucose concentration are among the five important state variables that can be calculated online by using appropriate sensors. The protein concentration is determined off-line. The inducer concentration can be calculated by measuring the amount of inducer injected.<sup>29</sup> Therefore, the variables for estimation are the protein concentration  $x_4$  and cell mass  $x_2$  (expensive sensor).

To implement the neural state estimation, first it is necessary to define the estimated state error  $\tilde{x}_n$  as follows:

$$\tilde{x}_n = x_n - \hat{x}_n; \quad \tilde{\theta}(k) = \theta^* - \theta(k) \quad (18)$$

where  $\hat{x}_i$  is the estimated state variable of the nonlinear fed-batch bioprocess (NFBBP), and  $x_n$  is the off-line measured state, and its components are  $x_n = [x_{n2}, x_{n4}]$ . The nonlinear dynamics of the NFBBP described in ref 1 can be represented by an exact neural estimator, denoted by

$$x_{n+1} = \theta^{*T} \xi(x_n, u_n) + \varepsilon_n \quad x_0 = x(0) \quad (19)$$

where  $u_n$  is the input vector to neural estimator, being  $u_n = [u_n, S_n, V_n]$  and  $\theta^* \in \mathcal{R}^{m \times 1}$  is the optimal weight vector,  $\varepsilon_n \in \mathcal{R}^{b \times 1}$  is the neural approximation error, and  $\xi_i \in \mathcal{R}^{m \times 1}$  is the RBF that represent each neuron in the hidden layers:

$$\xi_i(x_n, u_n) = \exp\left(-\frac{1}{2\sigma_i^2} \left\| \begin{bmatrix} x_n \\ u_n \end{bmatrix} - c_i \right\|^2\right) \quad i = 1, 2, \dots, m \quad (20)$$

The subindex  $i$  indicates the neuron number of RBF functions  $\xi \in \mathcal{R}^{m \times 1}$ . The estimated variables number is  $l = 2$ , and  $m$  is the maximum number of neurons ( $m = 10$ ).

Since the output state vector is nonmeasured or affected with disturbances, there is then a need to estimate their values. A state estimator function based on eq 19 is determined as follows:

$$\hat{\mathbf{x}}_{n+1} = \hat{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{u}_n) \quad (21)$$

Now, making the difference between eq 19 and eq 21, the neural identification error can be described by

$$\begin{aligned} \tilde{\mathbf{x}}_{n+1} &= \mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1} \\ &= \boldsymbol{\theta}^{*T} \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{u}_n) - \hat{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{u}_n) + \boldsymbol{\varepsilon}_n \\ &= \dots = \tilde{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{u}_n) + \boldsymbol{\varepsilon}_n \end{aligned} \quad (22)$$

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

$$\tilde{\boldsymbol{\theta}}_n = \boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_n \quad (23)$$

To train the neural network for identification, several off-line data sets of the pair  $\mathbf{x}, \mathbf{u}$  were used. The learning rule to train the neural network is stated in the next theorem:

**Theorem 2:** The NFBPP defined by eq 1, can be approximated by the neural network eq 21 using a neuronal adjustment law defined by

$$\Delta \tilde{\boldsymbol{\theta}}_n = -\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{u}_n) \tilde{\mathbf{x}}_n^T \mathbf{P} \quad (24)$$

where  $\mathbf{P}$  is a diagonal definite positive matrix and  $\gamma$  is an arbitrary positive constant.

*Proof:* the demonstration of this theorem has been developed in Appendix C. The convergence of this estimator is very important because an exact representation of the variables to be estimated ( $\hat{\mathbf{x}}_n = [\hat{x}_{2,n}, \hat{x}_{4,n}]$ ) is necessary.

The performance of the states estimation is shown in the next section.

## 5. RESULTS AND DISCUSSION

To evaluate the performance of the controller, various simulation tests employing Matlab were carried out:

- A Montecarlo algorithm to tuning the optimal controller parameters ( $k_i$ ).
- Simulation under normal operating conditions with and without state estimation.
- Simulation under perturbations in the control actions.
- Simulation considering uncertainty in the parameters.

**Optimal Controller Parameters.** The performance of a controller depends directly on the correct selection of its parameters  $k_i$ . Therefore, in this subsection a Monte Carlo randomized experiment is proposed to find the optimal values of the controller parameters.

In the field of the processes and control systems, Monte Carlo methods have been found effective to manage problems related to robustness of uncertain systems.<sup>37</sup>

The Monte Carlo randomized algorithm (MCRA) is, by definition, a randomized algorithm that may produce an incorrect result, but the probability of such an incorrect result is bounded.<sup>38</sup> The number of simulations necessary to guarantee a reliable degree of accuracy and confidence (confidence boundaries) is achieved by using the following expression:<sup>37</sup>

$$N \geq \left\lceil \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\varepsilon}} \right\rceil \quad (25)$$

where  $\delta$  = confidence and  $\varepsilon$  = accuracy. The procedure carried out to implement this technique is as follows:

First, a confidence and accuracy degree is selected, then, the number of trials ( $N$ ) is computed according to eq 25. Below, random values are selected for the controller parameters and a simulation of the controlled system is carried out. Then, the total tracking error ( $E_T$ ) is calculated for this set of parameters. This procedure is repeated for  $N$  trials, then, the optimal controller parameters set is one for which  $E_T$  is minimal. For the simulations, it is fixed  $\delta = 0.01$  and  $\varepsilon = 0.005$ . Therefore,  $N \geq 920$ .

The initial conditions needed for the simulations are shown in Table 1. Parameter values and feeding concentrations can be seen in Table 2.

**Table 1. Initial Conditions for the State Variables (g/L)**

$x_{1,0}$	$x_{2,0}$	$x_{3,0}$	$x_{4,0}$	$x_{5,0}$	$x_{6,0}$	$x_{7,0}$
1.0	0.1	40.0	0.01	0.01	1.0	0.01

**Table 2. Feeding Concentrations and Parameters<sup>27</sup>**

$N$ (g/L)	$I$ (g/L)	$Y$	$k_{11}$ (h <sup>-1</sup> )
40	100	0.51	0.09

The final time for the process is  $T_f = 10$  h and the sample time for simulations is  $T_s = 0.1$  h.

The  $k_i$  values found for the minimum total tracking error after 1000 simulations (total simulation time, 70 min) are presented in Table 3.

**Table 3. Optimal Controller Parameters**

$k_1$	$k_2$	$k_3$	$k_4$	$k_5$
0.2805	0.6761	0.6520	0.1694	0.1266
$E_T = 0.0006$ – iteration no. 386				

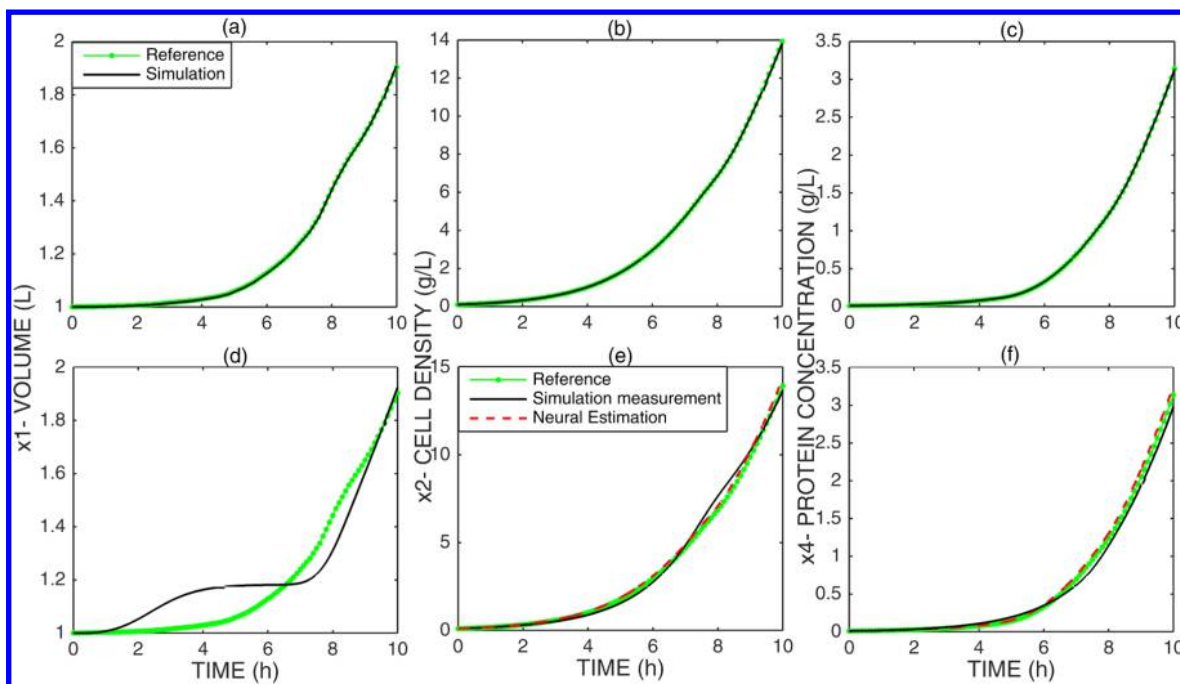
**Simulation in Normal Operating Conditions.** This section shows the performance of the proposed controller system without environmental disturbances. The set of optimal controller parameters found previously is used. First, a simulation without state estimation is carried out. Then, the state estimation designed in a previous section is integrated with the closed-loop control system.

A very good controller performance can be observed in Figure 3. The optimal desired profiles are successfully tracking as can be seen in Figure 3(a–c). The performance of the state estimation designed in a previous section is good, see Figure 3(d–f). The control actions profiles (without state estimation) are similar those obtained in Balsa-Canto et al.,<sup>28</sup> see Figure 4a. Figure 4b show the same profiles with the integrated state estimation.

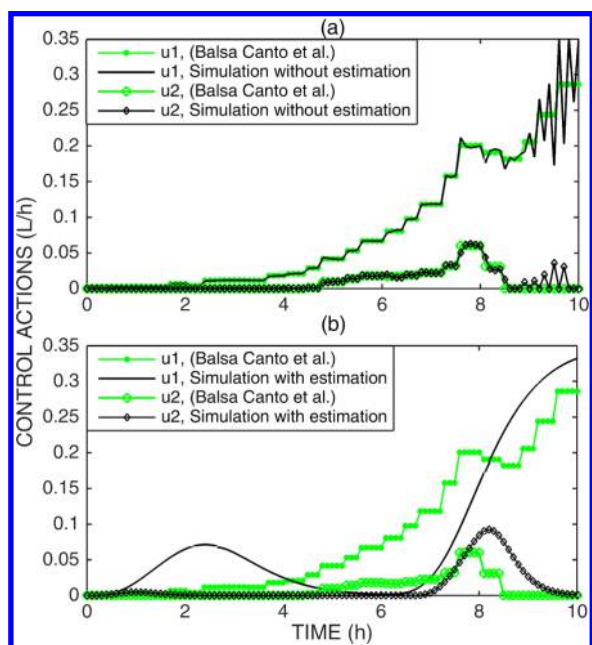
It is important to note that the final optimum volume is successfully achieved, which is very difficult since it is the sum of the two control actions used. Note that the tracking error defined by eq 14, remains low and acceptably bounded as shown in Figure 5. It should be noted too, that while Euler's discretization is very simple, it is useful enough for excellent controller performance.

The main objective of the closed-loop control developed in this work is following the optimal profiles of  $\times 1$ ,  $\times 2$ , and  $\times 4$  obtained in Balsa-Canto et al.,<sup>28</sup> which allow maximization of the profitability of the process.

The evaluation of the performance index  $J$  (26) is shown in Table 4.



**Figure 3.** Tracking optimal profiles of desired variables ( $x_1$ ,  $x_2$ , and  $x_4$ ) in normal operation conditions. (a–c) without state estimation; (d–f) with state estimation integrated in closed-loop control.

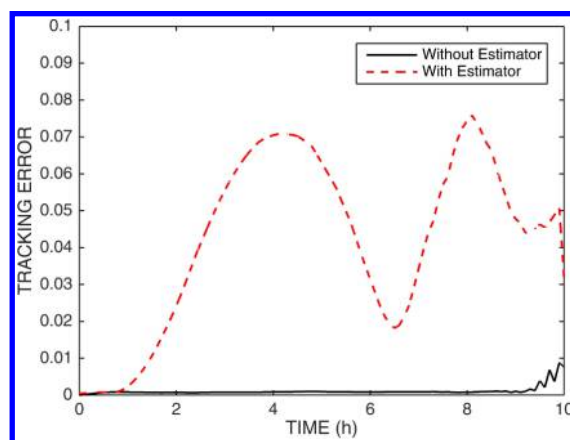


**Figure 4.** Control actions compared with those presented by Balsa-Canto et al.<sup>28</sup> (a) without state estimation, (b) with state estimation.

$$J = x_4(T_f)x_1(T_f) - Q \int_{T_0}^{T_f} u_2(t) dt \quad (26)$$

where  $Q = 2.5$ , and it is the ratio of the cost of the inducer to the value of the protein product.

It is important to remark that in the referenced work in said table, no closed-loop control law to track the optimal profiles achieved was proposed. The only precedent for closed-loop control of this bioprocess is that proposed by Lee and Ramirez;<sup>29</sup> they developed an online optimization of the fed-batch bioreactor. Unlike this work, its object was to keep the glucose



**Figure 5.** Tracking error ( $E_n$ ) in normal operation conditions.

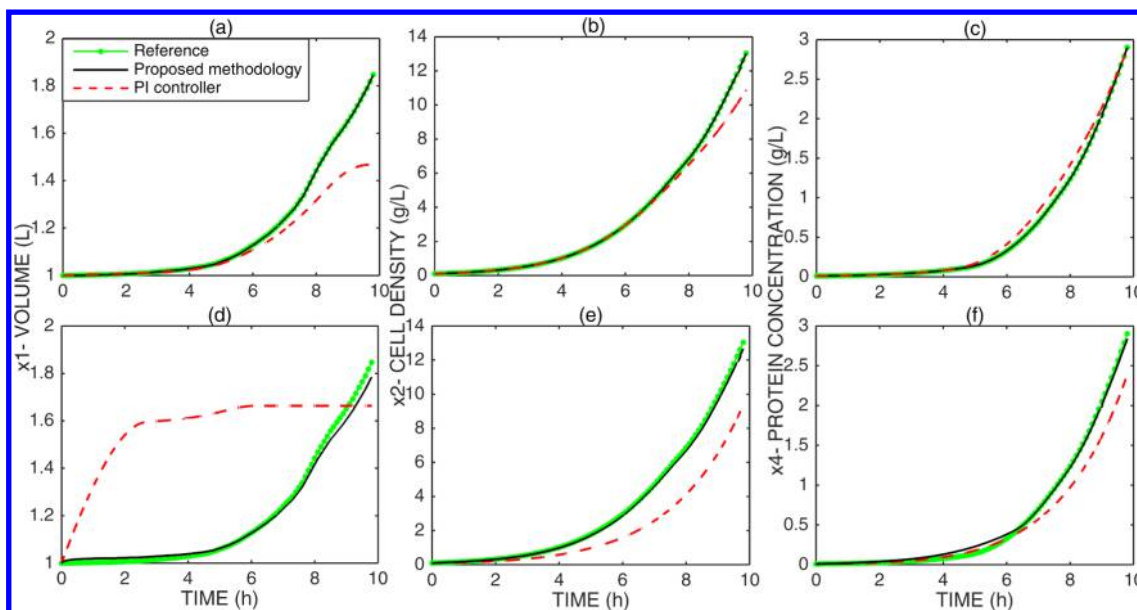
**Table 4.** Comparison of Results with that in the Literature

parameter/variable	this work	Balsa Canto et al. <sup>28</sup>
$x_1(T_f)$ (L)	1.9250	1.9039
$x_2(T_f)$ (g/L)	14.0670	13.9272
$x_4(T_f)$ (g/L)	3.2255	3.1396
$J$	5.4439	5.7569
$E$	0.05	N.D. <sup>a</sup>

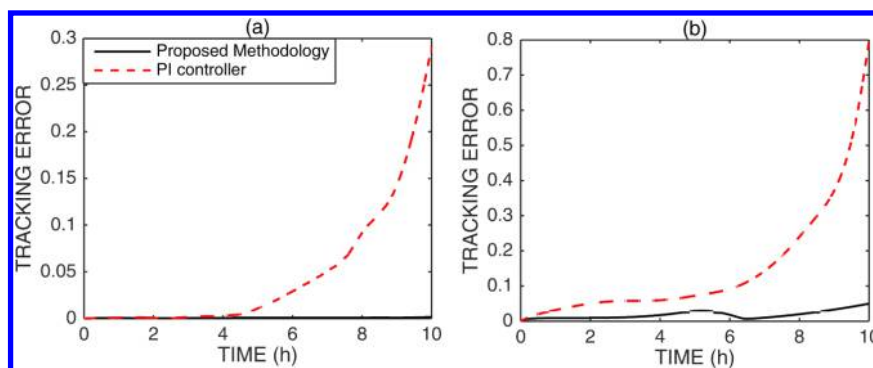
<sup>a</sup>N.D. means not developed by the authors.

and inducer concentrations at their optimal values using a combined PI–anticipatory regulatory control strategy.

The controller performance is also compared with a PI controller, which is widely used thanks to its versatility and facility of implementation. The best combination between the control actions and desired variables for the PI controller design is  $x_2$  controlled with  $u_1$  and  $x_4$  controlled with  $u_2$ . The optimal parameters of the PI controller are chosen by MCRA. Two tests



**Figure 6.** Comparison between the proposed control law and a PI controller. (a–c) tracking trajectory in normal operation conditions and (d–f) with the addition of constant perturbation in the control actions.



**Figure 7.** Comparison between the proposed control law and a PI controller. Tracking error ( $E_n$ ) in (a) normal operation conditions and (b) with the addition of constant perturbation in the control actions.

were carried out: a simulation in normal operating conditions and another with a constant perturbation in the control actions.

Figures 6 and 7 show the comparison between the control actions of the two controllers. Under normal operating conditions, Figure 6(a–c), the performance of the controller proposed herein is better than that of the PI controller, the total tracking error ( $E_T$ ) of the current technique is much lower than that of the PI. Now, considering the perturbations in the control actions, the performance of the controller proposed is still much better than that of the PI controller (see Figures 6(d–f) and 7b).

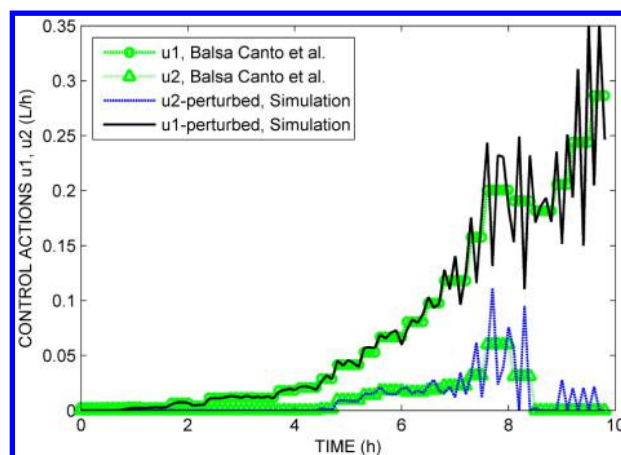
**Perturbation in the Control Actions.** The controller performance is tested now adding a perturbation in the control actions. To achieve it, a random function using MATLAB is employed. The control actions  $u_1$  and  $u_2$  are perturbed in a 20% of its normal values.

The function used was “ $random(unif,0,0.2)$ ” and the control actions are calculated as follow:

$$u_{1,perturbed} = u_{1,unperturbed}(1 + random(unif, 0, 0.2) + 1)$$

$$u_{2,perturbed} = u_{2,unperturbed}(1 + random(unif, 0, 0.2) + 1)$$

Figure 8 shows the perturbed control actions and the controller adaptation to keep the minimal tracking error. Figure



**Figure 8.** Perturbed control actions.

9 shows the tracking profiles in the perturbed system. Again, the controller performance is very good even considering disturbances in control actions.

**Parametric Uncertainty.** In this subsection, the system is analyzed in the case of appearing errors of modeling using the

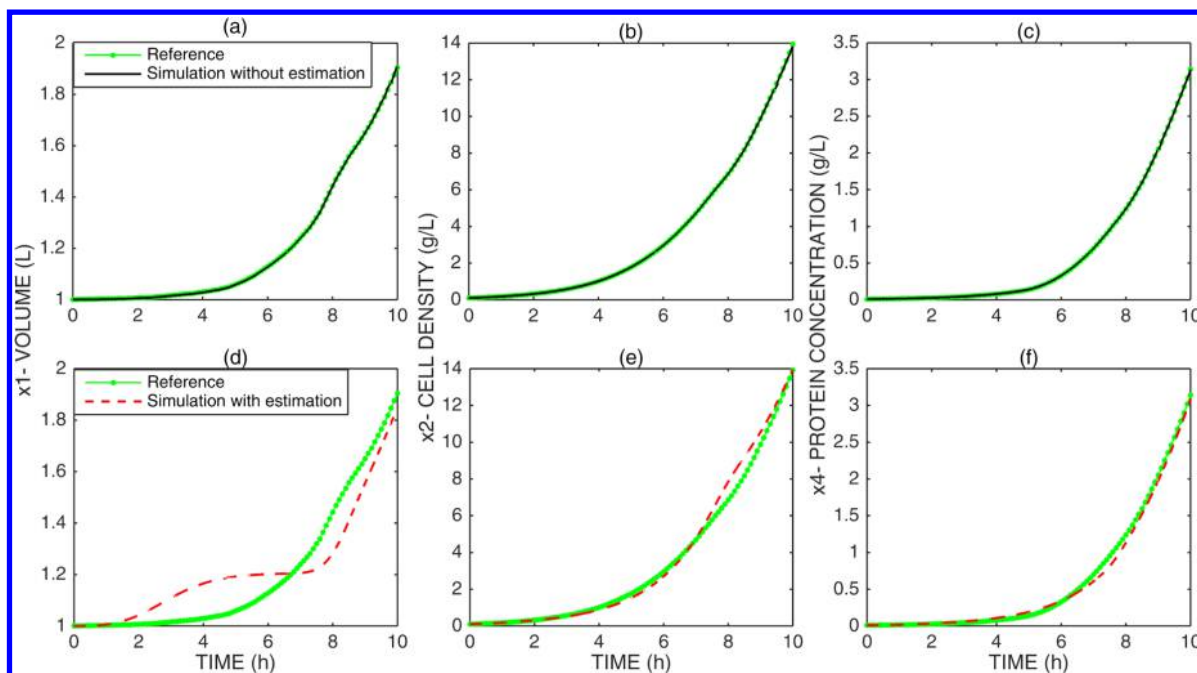


Figure 9. Tracking optimal profile with perturbed control actions (20%). (a–c) without state estimation, (d–f) with estimation.

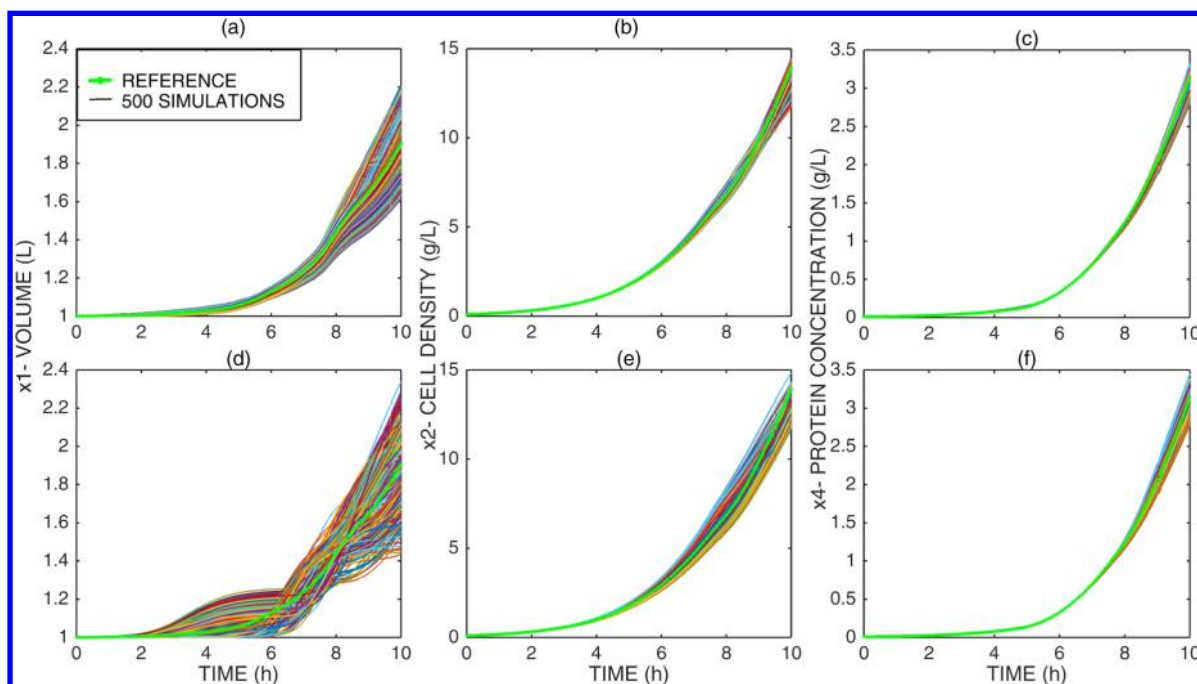


Figure 10. Tracking optimal profiles (desired variables  $x_1$ ,  $x_2$ ,  $x_4$ ) under parametric uncertainty ( $\pm 10\%$ , 500 simulations). (a–c) without estimation, (d–f) with state estimation.

MCRA method. Then, a determined error is introduced in the model parameters (10%), and 500 simulations ( $N = 500$ ) are performed. For each simulation, the parameters are chosen in a random way by the MCRA experiment.<sup>39</sup>

Considering the worst case, practically all the model parameters were varied:  $u_{\max}$ ,  $K_{CN}$ ,  $K_s$ ,  $K_{CV}$ ,  $f_{\max}^0$ ,  $f_1^0$ ,  $K_D$ ,  $K_{IX}$  and  $k_{11}$ . The initial conditions and the sampling time are the same used in the previous subsection.

The number of simulations taken to carry out this test considers the confidence  $\delta = 0.01$  and accuracy  $\epsilon = 0.01$ . Therefore, replacing in eq 25,  $N \geq 500$ .

Figure 10 shows the behavior of the system with the controller proposed in this work, which presents a good response against parametric uncertainty, both for the simulation without estimators (a–c) and for that with the integrated neural network (d–f). It is noteworthy that the controller design is focused in the tracking optimal profile of the output desired variables. This is one of the advantages, since it allows feedback control of the system even under disturbances.

The above figures show that the controller proposed in this work presents a certain level of robustness to parametric



uncertainty. The simulation tests carried out show the good performance of the control law.

6. CONCLUSION

An effective closed-loop controller integrated with a neural network state estimation has been designed for tracking three optimal profiles of an important nonlinear biological process by manipulation of two control actions. The advantages of the methodology employed is its simplicity and accuracy since it reduces the controller design to a resolution of a linear equations system for the calculation of the control actions. Furthermore, in contrast to conventional open-loop techniques mentioned herein,<sup>11,26-28</sup> this methodology proposes a closed-loop control strategy that achieves the tracking of optimal trajectories of desired variables with a minimal tracking error, even under

process disturbances and parametric uncertainties. Besides that, it can be implemented with only a basic knowledge of linear algebra. The proposed controller was compared with a PI controller, showing a better control response. The optimal controller parameters were successfully found through a Monte Carlo experiment. Moreover, through several closed-loop simulation tests, this proposed control structure was shown to be not only simple and efficient with the neural state estimation, but also sufficiently robust to compensate the mismatches in the model parameters as well as internal and external perturbations of the system.

■ APPENDIX A

The Gauss elimination process proposed in section 3 is shown below:

$a_{31}$	$a_{32}$	$b_3$	(A.1)
$a_{51}$	$a_{52}$	$b_5$	
$a_{11}$	$a_{12}$	$b_1$	
$a_{21}$	$a_{22}$	$b_2$	
$a_{41}$	$a_{42}$	$b_4$	
$a_{31}$	$a_{32}$	$b_3$	
0	$a_{31}a_{52} - a_{32}a_{51}$	$a_{31}b_5 - b_3a_{51}$	
0	$a_{31}a_{12} - a_{32}a_{11}$	$a_{31}b_1 - b_3a_{11}$	
0	$a_{31}a_{22} - a_{32}a_{21}$	$a_{31}b_2 - b_3a_{21}$	
0	$a_{31}a_{42} - a_{32}a_{41}$	$a_{31}b_4 - b_3a_{41}$	
$a_{31}$	$a_{32}$	$b_3$	
0	$a_{31}a_{52} - a_{32}a_{51}$	$a_{31}b_5 - b_3a_{51}$	
0	0	$(a_{31}a_{52} - a_{32}a_{51})(a_{31}b_1 - b_3a_{11}) - (a_{31}b_5 - b_3a_{51})(a_{31}a_{12} - a_{32}a_{11})$	
0	0	$(a_{31}a_{52} - a_{32}a_{51})(a_{31}b_2 - b_3a_{21}) - (a_{31}b_5 - b_3a_{51})(a_{31}a_{22} - a_{32}a_{21})$	
0	0	$(a_{31}a_{52} - a_{32}a_{51})(a_{31}b_4 - b_3a_{41}) - (a_{31}b_5 - b_3a_{51})(a_{31}a_{42} - a_{32}a_{41})$	

■ APPENDIX B

Theorem 1 proposes that if the system behavior is ruled by eq 10 and the controller is designed by eq 13, then,  $e_{i,n} \rightarrow 0, n \rightarrow \infty$ , when profile tracking problems are considered.

The demonstration of this theorem is shown for each variable  $x_i$ , for  $i = 1, 2, 3, 4, 5$  as follow.

**Demonstration 1:** The tracking error in the  $x_1$  variable tends to zero:

Consider eq 1,

$$u_1 + u_2 = \dot{x}_1 \tag{B.1}$$

then take the first and second row respectively of the reduced matrix results of the Gaussian elimination (see eq A.1),

$$u_1 = \frac{b_3}{a_{31}} - \frac{a_{32}}{a_{31}}u_2 \tag{B.2}$$

$$u_2 = \frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} \tag{B.3}$$

Note that the control actions in the above equations depend on  $x_{3,ez}$  and  $x_{5,ez}$ , the sacrificed variables found in  $b_3$  and  $b_5$ , respectively. Adding eq B.3 into eq B.2 gives

$$u_1 = \frac{b_3}{a_{31}} - \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} \right) \tag{B.4}$$

Now, consider that the third row of the reduced matrix results in Gaussian elimination (see eq A.1):

$$\frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} = \frac{a_{31}b_1 - a_{11}b_3}{a_{31}a_{12} - a_{32}a_{11}} \tag{B.5}$$

Combining eq B.5 and B.4 gives

$$u_1 = \frac{b_3}{a_{31}} - \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_1 - a_{11}b_3}{a_{31}a_{12} - a_{32}a_{11}} \right) \tag{B.6}$$

Summing the control actions  $u_1$  and  $u_2$  gives

$$u_1 + u_2 = \frac{b_3}{a_{31}} - \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_1 - a_{11}b_3}{a_{31}a_{12} - a_{32}a_{11}} \right) + \left( \frac{a_{31}b_1 - a_{11}b_3}{a_{31}a_{12} - a_{32}a_{11}} \right) \tag{B.7}$$

Performing a mathematical operation gives

$$u_1 + u_2 = b_1 \frac{(a_{31} - a_{32})}{(a_{31}a_{12} - a_{32}a_{11})} = b_1 \quad (\text{B.8})$$

Then eq B.8 and eq B.1 can be equated:

$$u_1 + u_2 = \dot{x}_1 = b_1 \quad (\text{B.9})$$

Consider eq 6 and eq 10:

$$\frac{\overbrace{x_{1,n+1} - x_{1,n+1}}^{\dot{x}_1}}{T_0} = \frac{\overbrace{(x_{1,\text{ref},n+1} - k_1(x_{1,\text{ref},n} - x_{1,n})) - x_{1,n}}^{b_1}}{T_0} \quad (\text{B.10})$$

Then,

$$\overbrace{x_{1,\text{ref},n+1} - x_{1,n+1}}^{e_{1,n+1}} = k_1 \overbrace{(x_{1,\text{ref},n} - x_{1,n})}^{e_{1,n}} \quad (\text{B.11})$$

Note that if the equations system 10 did not have a solution, that is, the eq 12 is not fulfilled, then  $u_1 + u_2 = b_1$  would not be fulfilled, therefore neither would eq 7, and the error would not tend to 0.

Equation B.9 proves that the equations system (10) has an exact solution and the control actions are calculated from  $x_{3,\text{ez}}$  and  $x_{5,\text{ez}}$ . Therefore,  $0 < k_1 < 1 \Rightarrow e_{1,n} \rightarrow 0$  when  $n \rightarrow \infty$ .

**Demonstration 2:** The tracking error in the  $x_5$  variable tends to zero.

Following from eq 1,

$$-u_1 + (I - x_5)u_2 = \dot{x}_5 x_1 \quad (\text{B.12})$$

and considering eq B.2 and eq B.3:

$$\begin{aligned} -u_1 + (I - x_5)u_2 &= -\frac{b_3}{a_{31}} + \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} \right) + \dots \\ &+ (I - x_5) \left( \frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} \right) \end{aligned} \quad (\text{B.13})$$

Mathematical manipulation and considering the value of  $a_{52} = (I - x_5)$  and  $a_{51} = (-1)$  (see eq 10) gives

$$-u_1 + (I - x_5)u_2 = \frac{b_5 a_{32}}{(a_{31}a_{52} - a_{32}a_{51})} + \frac{b_5 a_{31}(I - x_5)}{(a_{31}a_{52} - a_{32}a_{51})} \quad (\text{B.14})$$

$$\begin{aligned} -u_1 + (I - x_5)u_2 &= b_5 \frac{a_{32} + a_{31}(I - x_5)}{(a_{31}a_{52} - a_{32}a_{51})} = b_5 \\ \frac{a_{32} + a_{31}(I - x_5)}{(a_{32} + a_{31}(I - x_5))} &= b_5 \end{aligned} \quad (\text{B.15})$$

Then, eq B.12 and eq B.15 can be equated:

$$\dot{x}_5 x_1 = b_5 \quad (\text{B.16})$$

Considering eq 6 and eq 10:

$$\frac{\overbrace{x_{5,n+1} - x_{5,n+1}}^{\dot{x}_5}}{T_0} x_1 = \frac{\overbrace{(x_{5,\text{ez},n+1} - k_5(x_{5,\text{ez},n} - x_{5,n})) - x_{5,n}}^{b_5}}{T_0} x_1 \quad (\text{B.17})$$

Then,

$$\overbrace{x_{5,\text{ez},n+1} - x_{5,n+1}}^{e_{5,\text{ez},n+1}} = k_5 \overbrace{(x_{5,\text{ez},n} - x_{5,n})}^{e_{5,\text{ez},n}} \quad (\text{B.18})$$

Therefore, if  $0 < k_5 < 1 \Rightarrow e_{5,n} \rightarrow 0$  when  $n \rightarrow \infty$ .

**Demonstration 3:** The tracking error in the  $x_3$  variable tends to zero. Considering eq 1,

$$(N - x_3)u_1 - x_3 u_2 = \dot{x}_3 x_1 + \frac{x_2 x_1}{Y} \mu(x_3, x_5) \quad (\text{B.19})$$

and bearing in mind eqs B.2, B.3, and B.4:

$$\begin{aligned} (N - x_3)u_1 - x_3 u_2 &= (N - x_3) \frac{b_3}{a_{31}} + \dots \\ &+ \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_1 - a_{11}b_3}{a_{31}a_{12} - a_{32}a_{11}} \right) - x_3 \left( \frac{a_{31}b_1 - a_{11}b_3}{a_{31}a_{12} - a_{32}a_{11}} \right) \end{aligned} \quad (\text{B.20})$$

Mathematical manipulation gives

$$(N - x_3)u_1 - x_3 u_2 = b_3 \left( \frac{a_{31}a_{12} - a_{11}a_{32}}{a_{31}a_{12} - a_{32}a_{11}} \right) = b_3 \quad (\text{B.21})$$

Then, eqs B.19 and B.21 can be equated, and taking into account eq 6 and 10:

$$\begin{aligned} \frac{(x_{3,n+1} - x_{3,n})}{T_0} x_1 + \frac{x_2 x_1}{Y} \mu(x_{3,n}, x_{5,n}) \\ = \frac{(x_{3,\text{ez},n+1} - k_3(x_{3,\text{ez},n} - x_{3,n})) - x_{3,n}}{T_0} x_1 \\ + \frac{x_2 x_1}{Y} \mu(x_{3,n}, x_{5,n}) \end{aligned} \quad (\text{B.22})$$

then performing mathematical operations gives

$$\frac{(x_{3,\text{ez},n+1} - k_3(x_{3,\text{ez},n} - x_{3,n})) - x_{3,n}}{T_0} x_1 = -\frac{x_{3,n+1} - x_{3,n}}{T_0} x_1 \quad (\text{B.23})$$

$$\overbrace{x_{3,\text{ez},n+1} - x_{3,n+1}}^{e_{3,\text{ez},n+1}} = k_3 \overbrace{(x_{3,\text{ez},n} - x_{3,n})}^{e_{3,\text{ez},n}} \quad (\text{B.24})$$

$$e_{3,\text{ez},n+1} = k_3 e_{3,\text{ez},n} \quad (\text{B.25})$$

Therefore, if  $0 < k_3 < 1 \Rightarrow e_{3,n} \rightarrow 0$  when  $n \rightarrow \infty$ .

**Demonstration 4:** The tracking error in the  $x_2$  variable tends to zero. Following eq 1,

$$u_1 + u_2 = -\dot{x}_2 \frac{x_1}{x_2} + \mu(x_3, x_5) x_1 \quad (\text{B.26})$$

Now, consider the fourth row of the reduced matrix results of the Gaussian elimination (see eq A.1):

$$\frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} = \frac{a_{31}b_2 - a_{21}b_3}{a_{31}a_{22} - a_{32}a_{21}} \quad (\text{B.27})$$

Combining eq B.27 into eq B.2 and eq B.3, and then summing the control actions:

$$\begin{aligned} u_1 + u_2 &= \frac{b_3}{a_{31}} - \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_2 - a_{21}b_3}{a_{31}a_{22} - a_{32}a_{21}} \right) \\ &+ \left( \frac{a_{31}b_2 - a_{21}b_3}{a_{31}a_{22} - a_{32}a_{21}} \right) \end{aligned} \quad (\text{B.28})$$

Performing mathematical operations and considering the value of  $a_{22} = a_{21} = (-1)$  (see eq 10) gives

$$u_1 + u_2 = b_2 \frac{(a_{31} - a_{32})}{(a_{31}a_{22} - a_{32}a_{21})} = -b_2 \frac{(a_{31} - a_{32})}{(a_{31} - a_{32})} = -b_2 \tag{B.29}$$

Equating eqs B.26 and B.29, then replacing according to eq 6 and 10 gives

$$\begin{aligned} & - \left( \frac{\overset{\dot{x}_2}{x_{2,n+1} - x_{2,n}}}{T_0} \right) \frac{x_1}{x_2} + \mu(x_3, x_5)x_1 = \\ & - \left( \frac{(x_{2,ref,n+1} - k_2(x_{2,ref,n} - x_{2,n})) - x_{2,n}}{T_0} \right) \frac{x_1}{x_2} + \dots \\ & + \mu(x_{3,ez,n}, x_{5,ez,n})x_1 = -b_2 \end{aligned} \tag{B.30}$$

The Taylor approximation of  $\mu(x_{3,n}, x_{5,n})$  in the desired value  $x_{3,ez}$  and  $x_{5,ez}$  is

$$\begin{aligned} \mu(x_{3,n}, x_{5,n}) &= \mu(x_{3,ez,n}, x_{5,ez,n}) + \dots \\ &+ \left( x_{3,n} - x_{3,ez,n} \right) \frac{\partial \mu}{\partial x_{3,n}} \Bigg|_{x_{3,n}=x_{3,n}+\lambda(x_{3,n}-x_{3,ez,n})} + \dots \\ &+ \left( x_{5,n} - x_{5,ez,n} \right) \frac{\partial \mu}{\partial x_{5,n}} \Bigg|_{x_{5,n}=x_{5,n}+\theta(x_{5,n}-x_{5,ez,n})} \end{aligned} \tag{B.31}$$

Then,

$$\mu(x_{3,n}, x_{5,n}) = \mu(x_{3,ez,n}, x_{5,ez,n}) - e_{3,ez,n} \mu_\lambda - e_{5,ez,n} \mu_\theta \tag{B.32}$$

where  $0 < \lambda < 1$  and  $0 < \theta < 1$ . Combining eq B.32 into B.30:

$$e_{2,n+1} = k_2 e_{2,n} - T_0 x_2 e_{3,ez,n} \mu_\lambda - T_0 x_2 e_{5,ez,n} \mu_\theta \tag{B.33}$$

Therefore, if  $0 < k_2 < 1 \Rightarrow e_{2,n} \rightarrow 0$  when  $n \rightarrow \infty$ .

**Demonstration 5:** The tracking error in the  $x_4$  variable tends to zero. Considering eq 1,

$$u_1 + u_2 = -\dot{x}_4 \frac{x_1}{x_4} + \frac{x_1}{x_4} R(x_3, x_5) \tag{B.34}$$

Now, consider the fifth row of the reduced matrix results of the Gaussian elimination (see eq A.1):

$$\frac{a_{31}b_5 - a_{51}b_3}{a_{31}a_{52} - a_{32}a_{51}} = \frac{a_{31}b_4 - a_{41}b_3}{a_{31}a_{42} - a_{32}a_{41}} \tag{B.35}$$

Combining eq B.35 into eqs B.2 and B.3, and then summing the control actions:

$$\begin{aligned} u_1 + u_2 &= \frac{b_3}{a_{31}} - \frac{a_{32}}{a_{31}} \left( \frac{a_{31}b_4 - a_{41}b_3}{a_{31}a_{42} - a_{32}a_{41}} \right) \\ &+ \left( \frac{a_{31}b_4 - a_{41}b_3}{a_{31}a_{42} - a_{32}a_{41}} \right) \end{aligned} \tag{B.36}$$

Mathematically operating and considering the value of  $a_{42} = a_{41} = (-1)$  (see eq 10): considering the value of  $a_{42} = a_{41} = (-1)$  (see eq 10):

$$u_1 + u_2 = -b_4 \frac{(a_{32} - a_{31})}{(-a_{31} + a_{32})} = -b_4 \tag{B.37}$$

Equating eqs B.34 and B.37, then replacing according to eq 6 and 10:

$$\begin{aligned} & -\dot{x}_4 \frac{x_1}{x_4} + \frac{x_2 x_1}{x_4} R(x_{3,n}, x_{5,n}) \\ &= \frac{(x_{4,ref,n+1} - k_4(x_{4,ref,n} - x_{4,n})) - x_{4,n} \frac{x_1}{x_4}}{T_0} \\ &+ \frac{x_2 x_1}{x_4} R(x_{3,ez,n}, x_{5,ez,n}) \end{aligned} \tag{B.38}$$

The Taylor approximation of  $R(x_{3,n}, x_{5,n})$  in the desired value  $x_{3,ez}$  and  $x_{5,ez}$  is

$$\begin{aligned} R(x_{3,n}, x_{5,n}) &= R(x_{3,ez,n}, x_{5,ez,n}) + \dots \\ &+ \left( x_{3,n} - x_{3,ez,n} \right) \frac{\partial R}{\partial x_{3,n}} \Bigg|_{x_{3,n}=x_{3,n}+\lambda(x_{3,n}-x_{3,ez,n})} + \dots \\ &+ \left( x_{5,n} - x_{5,ez,n} \right) \frac{\partial R}{\partial x_{5,n}} \Bigg|_{x_{5,n}=x_{5,n}+\theta(x_{5,n}-x_{5,ez,n})} \end{aligned} \tag{B.39}$$

where  $0 < \lambda < 1$  and  $0 < \theta < 1$ . Then,

$$R(x_{3,n}, x_{5,n}) = R(x_{3,ez,n}, x_{5,ez,n}) - e_{3,ez,n} R_\lambda - e_{5,ez,n} R_\theta \tag{B.40}$$

Combining eq B.40 into B.38:

$$\begin{aligned} & -\dot{x}_4 \frac{x_1}{x_4} + \frac{x_2 x_1}{x_4} R(x_{3,ez,n}, x_{5,ez,n}) - \frac{x_2 x_1}{x_4} e_{3,ez,n} R_\lambda - \frac{x_2 x_1}{x_4} e_{5,ez,n} R_\theta \\ &= \frac{(x_{4,ref,n+1} - k_4(x_{4,ref,n} - x_{4,n})) - x_{4,n} \frac{x_1}{x_4}}{T_0} \\ &+ \frac{x_2 x_1}{x_4} R(x_{3,ez,n}, x_{5,ez,n}) \end{aligned} \tag{B.41}$$

Mathematically operating:

$$\begin{aligned} (x_{4,ref,n+1} - x_{4,n+1}) &= k_4(x_{4,ref,n} - x_{4,n}) + \dots \\ &- T_0 x_2 e_{3,ez,n} R_\lambda - T_0 x_2 e_{5,ez,n} R_\theta \end{aligned} \tag{B.42}$$

$$e_{4,n+1} = k_4 e_{4,n} - T_0 x_2 e_{3,ez,n} R_\lambda - T_0 x_2 e_{5,ez,n} R_\theta \tag{B.43}$$

Therefore, if  $0 < k_4 < 1 \Rightarrow e_{4,n} \rightarrow 0$  when  $n \rightarrow \infty$ . Finally, combining eqs B.11, B.18, rB.25, B.33, B.43:

$$\begin{pmatrix} e_{1,n+1} \\ e_{2,n+1} \\ e_{3,n+1} \\ e_{4,n+1} \\ e_{5,n+1} \end{pmatrix} = \overbrace{\begin{pmatrix} k_1 & 0 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 & 0 \\ 0 & 0 & k_3 & 0 & 0 \\ 0 & 0 & 0 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_5 \end{pmatrix}}^{\text{linear system}} \begin{pmatrix} e_{1,n} \\ e_{2,n} \\ e_{3,n} \\ e_{4,n} \\ e_{5,n} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 \\ -x_2 e_{3,ez,n} \mu_\lambda & -x_2 e_{5,ez,n} \mu_\theta \\ 0 & 0 \\ -x_2 e_{3,ez,n} R_\lambda & -x_2 e_{5,ez,n} R_\theta \\ 0 & 0 \end{pmatrix}}_{\text{bounded nonlinearity} \rightarrow \text{zero}} \dots + T_s \begin{pmatrix} 0 & 0 \\ -x_2 e_{3,ez,n} \mu_\lambda & -x_2 e_{5,ez,n} \mu_\theta \\ 0 & 0 \\ -x_2 e_{3,ez,n} R_\lambda & -x_2 e_{5,ez,n} R_\theta \\ 0 & 0 \end{pmatrix} \quad (\text{B.44})$$

Equation B.44 represents a linear system and a bounded nonlinearity that tends to zero when  $0 < k_i < 1, i = 1, 2, 3, 4, 5$ , and  $n \rightarrow \infty$ , thus proving that the tracking errors tend to zero. The demonstration of the tracking error tending to zero for robotic systems can be seen in ref 40.

### APPENDIX C

The stability analysis of the neural network estimator proposed in this work is presented below.

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

*Proof:* Defining a candidate function of Lyapunov, where  $V_m$  is a positive definite function given by

$$L_m(k) = \tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + \gamma^{-1} \text{tr}(\tilde{\boldsymbol{\theta}}_{n-1}^T \tilde{\boldsymbol{\theta}}_{n-1}) \quad (\text{C.1})$$

where  $\text{tr}(\cdot)$  is a trace operator. Now, taking the difference in discrete time  $\Delta L_m$  as follows,

$$\Delta L_m = (\tilde{\mathbf{x}}_{n+1}^T \mathbf{P} \tilde{\mathbf{x}}_{n+1} - \tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n) + \gamma^{-1} \text{tr}(\tilde{\boldsymbol{\theta}}_n^T \tilde{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}_{n-1}^T \tilde{\boldsymbol{\theta}}_{n-1}) \quad (\text{C.2})$$

Defining  $\Delta \boldsymbol{\theta}$ ; as

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

Organizing eq C.2 and considering  $\Delta \tilde{\mathbf{x}}_n = \tilde{\mathbf{x}}_{n+1} - \tilde{\mathbf{x}}_n$ :

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

Next, replacing eq 22 into C.4,  $\Delta L_m$  is written as

$$\begin{aligned} \Delta L_m &= 2\tilde{\mathbf{x}}_n^T \mathbf{P} (\tilde{\mathbf{x}}_{n+1} - \tilde{\mathbf{x}}_n) + \Delta \tilde{\mathbf{x}}_n^T \mathbf{P} \Delta \tilde{\mathbf{x}}_n + \Delta \boldsymbol{\theta} = \dots \\ &= 2\tilde{\mathbf{x}}_n^T \mathbf{P} (\tilde{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) + \varepsilon_n - \tilde{\mathbf{x}}_n) + \Delta \tilde{\mathbf{x}}_n^T \mathbf{P} \Delta \tilde{\mathbf{x}}_n + \Delta \boldsymbol{\theta} \end{aligned} \quad (\text{C.5})$$

From eq C.3, and rewriting  $\Delta \boldsymbol{\theta}$ , it yields

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

Then, eq C.6, is combined into eq C.5, where upon reorganizing the terms gives

$$\begin{aligned} \Delta L_m &= -2\tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\boldsymbol{\theta}}_n^T \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) + \Delta \tilde{\mathbf{x}}_n^T \mathbf{P} \Delta \tilde{\mathbf{x}}_n \\ &+ 2\tilde{\mathbf{x}}_n^T \mathbf{P} \varepsilon_n + \Delta \boldsymbol{\theta} = \dots = -2\tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n^T \mathbf{P} \Delta \tilde{\mathbf{x}}_n \\ &+ 2\tilde{\mathbf{x}}_n^T \mathbf{P} \varepsilon_n + \dots + \text{tr}(2\tilde{\boldsymbol{\theta}}_n^T [\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P} + \gamma^{-1} \Delta \tilde{\boldsymbol{\theta}}_n]) \\ &- \gamma^{-1} \text{tr}(\Delta \tilde{\boldsymbol{\theta}}_n^T \Delta \tilde{\boldsymbol{\theta}}_n) \end{aligned} \quad (\text{C.7})$$

Replacing the adjustment laws eq 24,  $\Delta \tilde{\boldsymbol{\theta}}$ , in eq C.7,  $\Delta L_m$  is represented by,

$$\begin{aligned} \Delta L_m &= -2\tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + \Delta \tilde{\mathbf{x}}_n^T \mathbf{P} \Delta \tilde{\mathbf{x}}_n + 2\tilde{\mathbf{x}}_n^T \mathbf{P} \varepsilon_n + \dots \\ &- \gamma^{-1} \text{tr}((\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})^T (\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})) \end{aligned} \quad (\text{C.8})$$

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

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

Where the subindex  $i$  denotes the  $i$ th row of  $\Delta \tilde{\boldsymbol{\theta}}$ . The partial derivative of the model error depends only on the weights of the neural network and can be rewritten as

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

Change the values of the weights according to eq C.6, and consider that  $\theta^*$  is constant, noting that ideal weight vectors are required only for analytical purposes. Now reorganize  $\Delta \hat{\boldsymbol{\theta}}_{ni}$ , so it can be rewritten as

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

Equation C.10 can be written as

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

Consider the value of the partial derivatives of the neural network, eq 24 and combine in eq C.10, yielding

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

The increase in the model error is defined as

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

where

$$\begin{aligned} \gamma &= \gamma \|\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)\|^2 \\ \max(\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)) &\leq 1 \end{aligned} \quad (\text{C.15})$$

where  $\|\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)\|$  a bounded function. The value  $\gamma$  is a learning factor of the neural network ( $0 < \gamma < 1$ ) and it can be arbitrarily defined. Equation C.14 is expressed in vectorial form:

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

Substituting the increment value of the model error in eq C.16 gives the Lyapunov discrete difference, defined as

$$\begin{aligned} \Delta L_m &= \tilde{\mathbf{x}}_n^T \gamma^2 \mathbf{P} \tilde{\mathbf{x}}_n - 2 \tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + 2 \tilde{\mathbf{x}}_n^T \mathbf{P} \mathbf{e}_n + \dots \\ &\quad - \gamma^{-1} \text{tr}((\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})^T (\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})) \\ \Delta L_m &= -\tilde{\mathbf{x}}_n^T (\mathbf{P}(1 - \gamma^2)) \tilde{\mathbf{x}}_n + 2 \tilde{\mathbf{x}}_n^T \mathbf{P} \mathbf{e}_n + \dots \\ &\quad - \gamma^{-1} \text{tr}((\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})^T (\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})) \end{aligned} \quad (\text{C.17})$$

Now replacing the learning rule in eq C.17 and applying norm yields

$$\begin{aligned} \Delta L_m &= -\tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + \gamma^2 \tilde{\mathbf{x}}_n^T \mathbf{P} \tilde{\mathbf{x}}_n + 2 \tilde{\mathbf{x}}_n^T \mathbf{P} \mathbf{e}_n + \dots \\ &\quad - \gamma^{-1} \text{tr}((\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})^T (\gamma \boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n) \tilde{\mathbf{x}}_n^T \mathbf{P})) \\ \Delta L_m &\leq -\|\mathbf{P}\| \|\tilde{\mathbf{x}}_n\|^2 + \gamma^2 \|\mathbf{P}\| \|\tilde{\mathbf{x}}_n\|^2 + 2 \lambda_{\max}(\mathbf{P}) \|\tilde{\mathbf{x}}_n\| \bar{\varepsilon} + \dots \\ &\quad - \gamma^2 \|\mathbf{P}\| \|\boldsymbol{\xi}(\mathbf{x}_n, \mathbf{v}_n)\|^2 \|\tilde{\mathbf{x}}_n\|^2 \\ \Delta L_m &\leq \|\mathbf{P}\| \|\tilde{\mathbf{x}}_n\|^2 + 2 \|\mathbf{P}\| \|\tilde{\mathbf{x}}_n\| \bar{\varepsilon} \end{aligned} \quad (\text{C.18})$$

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

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

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

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

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

## AUTHOR INFORMATION

### Corresponding Author

\*E-mail: npantano@unsj.edu.ar.

### ORCID

María N. Pantano: 0000-0003-2549-6535

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

This work was funded by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET- National Council for Scientific Research) and Instituto de Ingeniería Química, Facultad de Ingeniería—Universidad Nacional de San Juan, Argentina.

## REFERENCES

- (1) Soni, A. S.; Parker, R. S. Closed-loop control of fed-batch bioreactors: a shrinking-horizon approach. *Ind. Eng. Chem. Res.* **2004**, *43*, 3381–3393.
- (2) Rani, K. Y.; Rao, V. R. Control of fermenters—a review. *Bioprocess Eng.* **1999**, *21*, 77–88.
- (3) Yamanè, T.; Shimizu, S. *Bioprocess Parameter Control*; Springer, 1984; pp 147–194.
- (4) Banga, J. R.; Balsa-Canto, E.; Moles, C. G.; Alonso, A. A. Dynamic optimization of bioreactors: a review. *Proc. Ind. Natl. Sci. Acad. Part A* **2003**, *69*, 257–266.
- (5) Bayen, T.; Mairet, F. Minimal time control of fed-batch bioreactor with product inhibition. *Bioprocess Biosyst. Eng.* **2013**, *36*, 1485–1496.

(6) Renard, F.; Wouwer, A. V.; Valentinotti, S.; Dumur, D. A practical robust control scheme for yeast fed-batch cultures—An experimental validation. *J. Process Control* **2006**, *16*, 855–864.

(7) Lee, J.; Lee, S. Y.; Park, S.; Middelberg, A. P. Control of fed-batch fermentations. *Biotechnol. Adv.* **1999**, *17*, 29–48.

(8) Johnson, A. The control of fed-batch fermentation processes—a survey. *Automatica* **1987**, *23*, 691–705.

(9) Rocha, M.; Mendes, R.; Rocha, O.; Rocha, I.; Ferreira, E. C. Optimization of fed-batch fermentation processes with bio-inspired algorithms. *Expert Systems with Applications* **2014**, *41*, 2186–2195.

(10) Sarkar, D.; Modak, J. M. Optimisation of fed-batch bioreactors using genetic algorithms. *Chem. Eng. Sci.* **2003**, *58*, 2283–2296.

(11) Sarkar, D.; Modak, J. M. Optimization of fed-batch bioreactors using genetic algorithm: multiple control variables. *Comput. Chem. Eng.* **2004**, *28*, 789–798.

(12) Renard, F.; Vande Wouwer, A. Robust adaptive control of yeast fed-batch cultures. *Comput. Chem. Eng.* **2008**, *32*, 1238–1248.

(13) Cosenza, B.; Galluzzo, M. Nonlinear fuzzy control of a fed-batch reactor for penicillin production. *Comput. Chem. Eng.* **2012**, *36*, 273–281.

(14) Ronen, M.; Shabtai, Y.; Guterman, H. Optimization of feeding profile for a fed-batch bioreactor by an evolutionary algorithm. *J. Biotechnol.* **2002**, *97*, 253–263.

(15) Ashoori, A.; Moshiri, B.; Khaki-Sedigh, A.; Bakhtiari, M. R. Optimal control of a nonlinear fed-batch fermentation process using model predictive approach. *J. Process Control* **2009**, *19*, 1162–1173.

(16) Craven, S.; Whelan, J.; Glennon, B. Glucose concentration control of a fed-batch mammalian cell bioprocess using a nonlinear model predictive controller. *J. Process Control* **2014**, *24*, 344–357.

(17) Santos, L. O.; Dewasme, L.; Coutinho, D.; Wouwer, A. V. Nonlinear model predictive control of fed-batch cultures of microorganisms exhibiting overflow metabolism: assessment and robustness. *Comput. Chem. Eng.* **2012**, *39*, 143–151.

(18) Carrasco, E. F.; Banga, J. R. Dynamic optimization of batch reactors using adaptive stochastic algorithms. *Ind. Eng. Chem. Res.* **1997**, *36*, 2252–2261.

(19) Saint-Donat, J.; Bhat, N.; McAvoy, T. J. Neural net based model predictive control. *Int. J. Control* **1991**, *54*, 1453–1468.

(20) Tholudur, A.; Ramirez, W. F. Optimization of Fed-Batch Bioreactors Using Neural Network Parameter Function Models. *Biotechnol. Prog.* **1996**, *12*, 302–309.

(21) Berber, R. Control of Batch Reactors—A Review (Reprinted from Methods of Model Based Process Control, 1995). *Chem. Eng. Res. Des.* **1996**, *74*, 3–20.

(22) Alford, J. S. Bioprocess control: Advances and challenges. *Comput. Chem. Eng.* **2006**, *30*, 1464–1475.

(23) Chang, D.-M. The Snowball Effect in Fed-Batch Bioreactions. *Biotechnology progress* **2003**, *19*, 1064–1070.

(24) Chung, Y.-C.; Chien, I.-L.; Chang, D.-M. Multiple-model control strategy for a fed-batch high cell-density culture processing. *J. Process Control* **2006**, *16*, 9–26.

(25) Lee, J.; Ramirez, W. F. Mathematical modeling of induced foreign protein production by recombinant bacteria. *Biotechnol. Bioeng.* **1992**, *39*, 635–646.

(26) Lee, J.; Ramirez, W. F. Optimal fed-batch control of induced foreign protein production by recombinant bacteria. *AIChE J.* **1994**, *40*, 899–907.

(27) Tholudur, A.; Ramirez, W. F. Obtaining smoother singular arc policies using a modified iterative dynamic programming algorithm. *Int. J. Control* **1997**, *68*, 1115–1128.

(28) Balsa-Canto, E.; Banga, J. R.; Alonso, A. A.; Vassiliadis, V. S. Efficient optimal control of bioprocesses using second-order information. *Ind. Eng. Chem. Res.* **2000**, *39*, 4287–4295.

(29) Lee, J.; Ramirez, W. F. On-line optimal control of induced foreign protein production by recombinant bacteria in fed-batch reactors. *Chem. Eng. Sci.* **1996**, *51*, 521–534.

(30) Zhou, K.; Doyle, J. C.; Glover, K. *Robust and Optimal Control*; Prentice Hall: NJ, 1996; Vol. 40.

- (31) Dochain, D. State and parameter estimation in chemical and biochemical processes: a tutorial. *J. Process Control* **2003**, *13*, 801–818.
- (32) De Battista, H.; Picó, J.; Garelli, F.; Vignoni, A. Specific growth rate estimation in (fed-) batch bioreactors using second-order sliding observers. *J. Process Control* **2011**, *21*, 1049–1055.
- (33) Svrcek, W.; Elliott, R.; Zajic, J. The extended Kalman filter applied to a continuous culture model. *Biotechnol. Bioeng.* **1974**, *16*, 827–846.
- (34) Bastin, G.; Dochain, D. On-line estimation and adaptive control of bioreactors: Elsevier, Amsterdam, 1990 (ISBN 0-444-88430-0). *Anal. Chim. Acta* **1991**, *243*, 324.
- (35) Thibault, J.; Van Breusegem, V.; Chéruey, A. On-line prediction of fermentation variables using neural networks. *Biotechnol. Bioeng.* **1990**, *36*, 1041–1048.
- (36) Karim, M.; Rivera, S. *Modern Biochemical Engineering*; Springer, 1992; pp 1–33.
- (37) Tempo, R.; Ishii, H. Monte Carlo and Las Vegas Randomized Algorithms for Systems and Control\*: An Introduction. *European journal of control* **2007**, *13*, 189–203.
- (38) Motwani, R.; Raghavan, P. *Randomized Algorithms*; Cambridge International Series on Parallel Computation; Cambridge University Press, 1995.
- (39) Cheein, F. A. A.; Pereira, F. M. L.; Di Sciascio, F.; Carelli, R. Autonomous simultaneous localization and mapping driven by Monte Carlo uncertainty maps-based navigation. *Knowledge Engineering Review* **2013**, *28*, 35–57.
- (40) Scaglia, G.; Rosales, A.; Quintero, L.; Mut, V.; Agarwal, R. A linear-interpolation-based controller design for trajectory tracking of mobile robots. *Control Engineering Practice* **2010**, *18*, 318–329.