Identification and Control of a Nonlinear Bioreactor Plant Using Classical and Dynamical Neural Networks

Mehmet Önder Efe
Electrical and Electronics Engineering
Boğaziçi University, Bebek 80815, Istanbul, Turkey
e-mail : efemond@boun.edu.tr

Okyay Kaynak,
UNESCO Chair On Mechatronics
Boğaziçi University, 80815, Bebek, Istanbul, Turkey
e-mail : kaynak@boun.edu.tr

Abstract - **In this study, the identification and control of a Bioreactor Plant using neural networks is considered with three different control strategies, namely, Inverse Control Strategy, Self-Learning Control and Dynamical Neural Units for control of nonlinear dynamical systems. The performance of these methods are compared using several comparison measures.**

1. INTRODUCTION

In modeling of real life systems we meet challenging difficulties in the form of nonlinearity, time delay, saturation and time varying parameters of the actual plant. Alleviating these kinds of challenging difficulties with a mathematically tractable model is a formidable problem.

Artificial neural networks can identify a system by the use of training data which is obtained from the system inputs and outputs. In this paper, we elaborate the identification and control of a bioreactor plant whose governing equations are coupled and nonlinear.

The bioreactor is a challenging problem for neural network controllers for several reasons. Although the task involves few variables and is easily simulated, its nonlinearity makes it difficult to control. For example, small changes in the values of parameters can cause the bioreactor to become unstable. Significant delays exist between changes in flow rate and the response in cell concentration. Nonlinearities in the dynamics of the bioreactor present a challenge to networks for learning nonlinear models. Additionally, uncontrolled equations exhibit limit cycles. Using neural networks that learn to compensate for deficiencies in the performance of the conventional controllers can alleviate some of these problems [1].

2. PLANT MODEL

The bioreactor is a tank containing water, nutrients, and biological cells as shown in Fig. 2.1. Nutrients and cells are introduced into the tank where the cells mix with nutrients. The state of this process is characterized by the number of cells $c_1(t)$ and the amount of nutrients $c_2(t)$. The volume in the tank is maintained at a constant level by removing tank contents at a rate equal to the incoming rate which is denoted by r(t). This rate is called the flow rate and is the variable by which the bioreactor is controlled. The bioreactor control problem is to keep the amount of cells at a desired level. In the simulation studies presented in this paper, the state variables can take values between zero and one, the flow rate can take values between zero and two. In [1], the stable state of the process is defined to be $c_1 = 0.1207$, $c_2 = 0.8801$, and $r = 0.7500$; the initialization of the state variables and the flow rate is made by considering these values such that the initial value of any state variable is within plus or minus ten percent of its stable value. Moreover, initial values exhibit uniform distribution over their interval and are chosen randomly. Continuous-time equations of the plant dynamics are given by;

$$\dot{c}_1(t) = -c_1(t)r(t) + c_1(t)\left(1-c_2(t)\right)e^{\frac{c_2(t)}{\gamma}} \quad (1)$$

$$\dot{c}_2(t) = -c_2(t)r(t) + c_1(t)\left(1-c_2(t)\right)e^{\frac{c_2(t)}{\gamma}}\frac{1+\beta}{1+\beta - c_2(t)} \quad (2)$$

where $\beta = 0.02$ growth rate parameter, $\gamma = 0.48$ nutrient inhibition parameter. Controller inputs are the state variables. In our simulations, Eqns. (1) and (2) are discretized by the use of first order approximation with a sampling interval of $\Delta = 0.01$ sec. The control interval is taken as 50Δ. The output of the controller is the flow rate r(t). The objective is to achieve and maintain a desired cell amount, by altering the flow rate.

3. SYSTEM IDENTIFICATION USING NEURAL NETWORKS

The system that is to be identified can be represented by a transformation operator T_p, which maps the compact subset $U \in \mathbf{R}^n$ to $Y \in \mathbf{R}^m$. The purpose is to find a class T_i such that T_p is represented by T_i adequately well. The operator T_p is defined by specific input-output pairs that are obtained from the inputs and the outputs of the system to be identified. The objective is expressed in Eqn. (3), for some desired $\varepsilon>0$. $T_i(u)$ denotes the identification model output. In Fig. 3.1, system identification structure is illustrated.

$$\left\| T_i(u) - T_p(u) \right\| \le \varepsilon \quad , u \in U \quad (3)$$

The most important aspect of a neural network based identification scheme is the determination of an adaptive algorithm that minimizes the difference between the actual plant and the outputs of the identified model by using a set of training pairs which represent the approximate behavior of the actual plant.

4. INVERSE CONTROL STRATEGY (ICS)

In this approach, the neural controller directly interpolates the inverse dynamics of the plant by using the training data obtained from the plant itself. Assume that the governing equation of the plant is given by Eqn. (4),

$$x(k+1) = F\,[x(k), u(k)] \quad (4)$$

$$u(k) = G[x(k+1), x(k)] \quad (5)$$

where $x \in R^m$ and $u \in R^n$. The plant performs a nonlinear mapping from R^{m+n} to R^m. Our objective is to solve the plant equation for the input vector. If a controller realizes the nonlinear mapping G, then it is able to control the plant. From a systems theoretical point of view, the controller realizes the inverse dynamics of the plant. Given the state values and the desired next state, it generates the input which will cause the desired state transition. As long as the states are observable, we can generate the training data for the neural controller so that it realizes the mapping G which is from R^{2m} to R^n.

The control system architecture for this control strategy is shown in Fig.4.1 and the corresponding simulation results are given in Fig.4.2.

5. SELF-LEARNING CONTROL (SLC)

In this section, the self-learning control scheme which was originally proposed by Nguyen and Widrow [3] is elaborated upon. The method differs from the other methods in that this scheme is goal directed. The controller is trained to keep the plant output at a desired and previously defined level. The approach utilizes the well-known backpropagation method in the training of the controller.

The first step of the control procedure is obtaining the neural identification model of the plant. In Fig. 5.1, identification model is represented by E (Emulator) boxes. This architecture implies that after K time steps, plant output could be pulled from its initial state y_0 to the desired state y_d. K is determined by the designer and represents the length of the chain-like architecture. The objective of the strategy is minimization of the cost function defined by Eqn. (6).

$$J = E\left(\| y_d - y_K \|^2 \right) \tag{6}$$

The minimization of Eqn. (6) is carried out through evaluating the output error at the K^{th} step and propagating it back through the structure illustrated in Fig. 5.1. Utilizing the plant identification model makes the training process easy. An important point in the weight updating is the succession that appears in the controller training structure. In fact, the weight change at the K^{th} stage influences the $(K-1)^{th}$ stage. This obviously requires the saving of each individual weight change evaluated at the corresponding stage. Nguyen and Widrow [3], in their work, say that the weight changes could be applied immediately as they are evaluated because they are the accumulated effects that improve the performance of the controller and this does not affect the final performance significantly. Having trained the neural controller, it is installed to the control system as shown in Fig. 5.2. The simulation results are presented in Fig. 5.3.

Identification model	:	3-25-16-2	MSE : 88e-6
Controller	:	2-8-4-1	MSE : 1e-6

As is stated previously, the control strategy adopted in this section is a goal directed one, the desired cell (c_1) and nutrient (c_2) concentrations being taken as 0.1000 and 0.8800 respectively. Simulation results presented indicate that the desired c_1 value is reached with a fast response, without the sharp deviation that is observed in the case of ICS (Fig. 4.2) even in the regions when the desired trajectory is constant. The SLC strategy therefore results in a better and faster performance.

6. DYNAMICAL NEURAL UNITS (DNU) FOR CONTROL OF NONLINEAR SYSTEMS

In this section, a new neuron model which adopts a second order system with an output sigmoidal nonlinearity as a single neuron is studied. The model is comprised of synaptic and somatic parts and adaptation is carried out on the coefficients of this second order block (synaptic part) and the on the slope of its nonlinear activation function (somatic part).

The topology of a single dynamical neural unit consists delay elements, feedforward and feedback synaptic weights and a nonlinear somatic operator. The architecture of the DNU model is illustrated in Fig 6.1. The difference equation which describes the behavior of the second order dynamical structure is given in Eqn. (7) in which $v_1(k)$, $x(k) \in R^1$. Similarly, the pulse transfer function of this part can be given by Eqn. (8).

$$v_1(k) = - b_1 v_1(k-1) - b_2 v_1(k-2) + a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) \tag{7}$$

$$T(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{8}$$

The output of the DNU can be evaluated as follows;

$$v(k) = g_s v_1(k) \tag{9}$$

$$u(k) = \Psi(v(k)) = \tanh(g_s v_1(k)) \tag{10}$$

The objective is based on the minimization of the instantaneous error evaluated at the output of the system. The cost function which is to be minimized is defined by Eqn. (12) in which E denotes the expectation operator. If w denotes the parameters of a single DNU, the update rule can be given by;

$$w_{k+1} = w_k - \mu \frac{\partial J}{\partial w} \tag{11}$$

$$J = \frac{1}{2} E\left(e^2(k)\right) \tag{12}$$

$$e(k) = y_d(k) - u(k) \tag{13}$$

More compactly, the parameter update rule is given by Eqns (14) through (16) where i = 0,1,2 and j = 1,2 ;

$$\Delta a_i(k+1) = \mu_{a_i} E\left(e(k)\, g_s^2(k)\, \mathrm{sech}^2[v_1(k)]\, x(k-i)\right) \tag{14}$$

$$\Delta b_j(k+1) = -\mu_{b_j} E\left(e(k)\, g_s^2(k)\, \mathrm{sech}^2[v_1(k)]\, v_1(k-j)\right) \tag{15}$$

$$\Delta g_s(k+1) = g_s(k) \mu_{g_s} E\left(e(k)\, \mathrm{sech}^2[v_1(k)]\, v_1(k)\right) \tag{16}$$

In the parameter update equations, the coefficients μ_{ai}, μ_{bj} and μ_{gs}, denote the step size for the corresponding parameter and are chosen to be constant throughout a simulation. In Fig. 6.2, DNU layer includes the desired number of individual DNU blocks whose inputs are connected together and whose outputs are added to form the control u(k). Depending on the magnitude of the output

error, the algorithm updates the feedforward and feedback weights and the gain of the nonlinear activation functions of each dynamical neural unit in the DNU layer. The derivation of the algorithm is given in [4] in detail. In our simulations, we applied the input r(t) = 0.1 + 0.05sin(2πt/50) as the reference input. The performance of the DNU based control strategy is given in Fig. 6.3.

7. CONCLUSIONS

Based on the extensive simulation results carried out within the course of this study, the performance of each approach with respect to several comparison metrics can be tabulated as given in Table 1. It is seen that DNU has the most desirable characteristics.

TABLE 1
COMPARISON OF CONTROL STRATEGIES

	ICS	SLC	DNU
Tracking performance	LOW	HIGH	HIGH
Applicability to different plants	LOW	HIGH	HIGH
Robustness under perturbations	LOW	HIGH	HIGH
Mathematical tractability for stability analysis	NO	NO	YES
Noise reduction	-	-	HIGH
Capability of fault tolerance	-	MID	HIGH

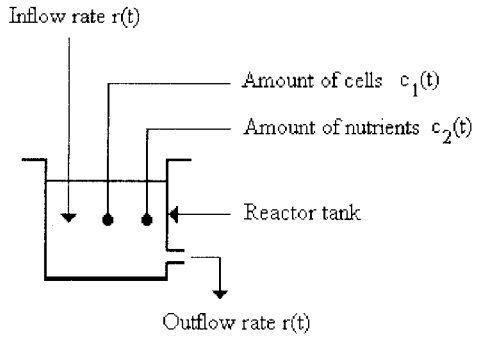

Fig. 2.1 The bioreactor is a tank of a liquid mixture of cells and nutrients.

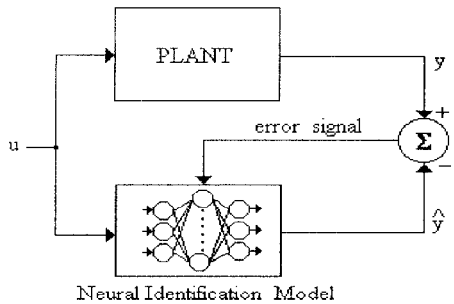

Fig. 3.1 System Identification Structure

8.REFERENCES

[1]. Ungar, L. H., "A Bioreactor Benchmark for Adaptive-Network Based Process Control," in W. T. Miller, R. S. Sutton, P. J. Werbos, *Neural Networks for Control*, MIT Press, 1991

[2]. Narendra, K. S., Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4-27, March 1990.

[3]. Nguyen, D. H., Widrow, B., "Neural Networks for Self-Learning Control Systems," *IEEE Control Systems Magazine*, pp. 18-23, April 1990.

[4]. Gupta, M. M., Rao, D. H., "Dynamical Neural Units with Applications to the Control of Unknown Nonlinear Systems," *Journal of Intelligent and Fuzzy Systems*, vol.1, no.1, pp. 73-92, 1993.

[5]. Hagan, M. T., Menhaj, M. B., "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989-993, November 1994.

[6]. Efe, Mehmet Önder, "Identification and Control of Nonlinear Dynamical Systems Using Neural Networks," M.S. Thesis, Boğaziçi University, 1996

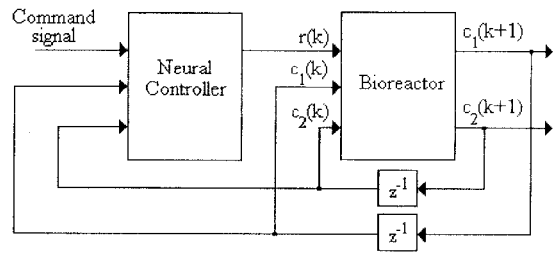

Fig. 4.1 Control System Architecture

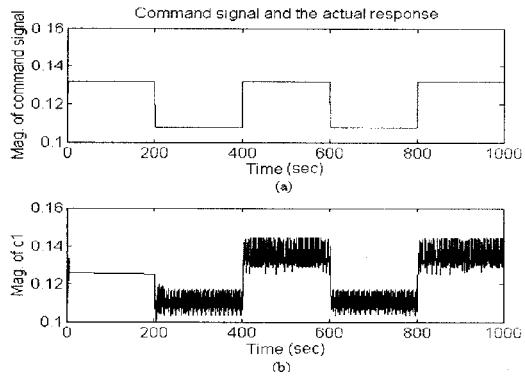

Fig. 4.2 (a) Command signal (b) Actual c_1 trajectory

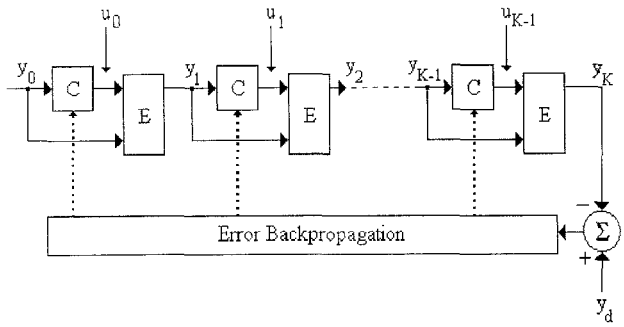

Fig. 5.1 Controller training architecture for the self-learning control scheme

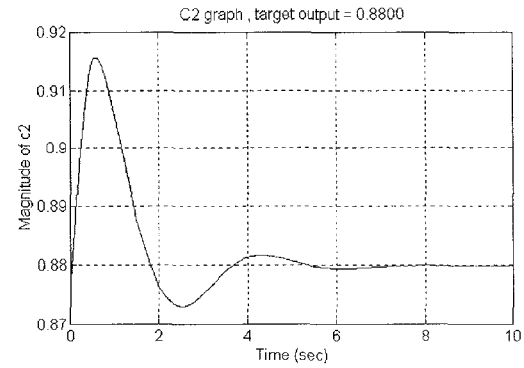

Fig. 5.3(b) The time evolution of nutrient concentration (c_2). The desired level is defined to be 0.8800

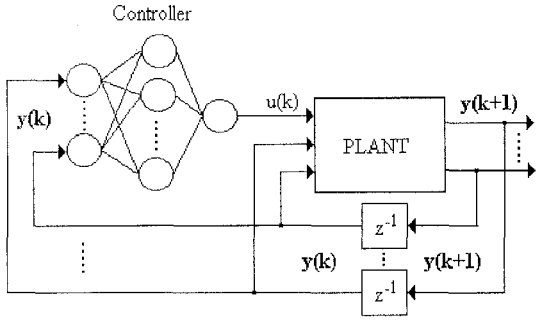

Fig. 5.2 control system structure with neural controller

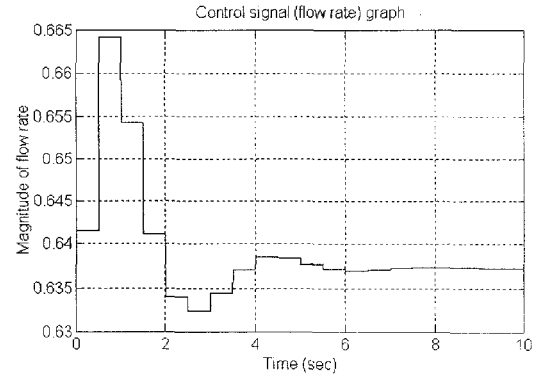

Fig. 5.3(c) The control signal (flow rate) that is applied to the bioreactor

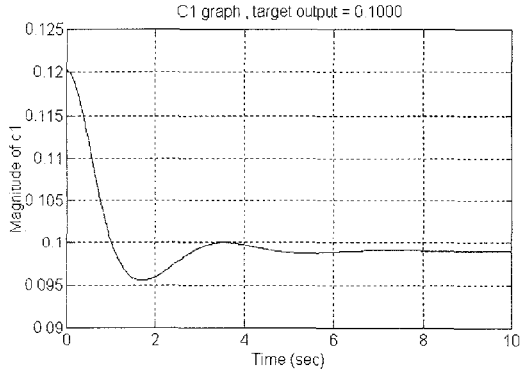

Fig. 5.3(a) The time evolution of cell concentration (c_1). The desired level is defined to be 0.1000

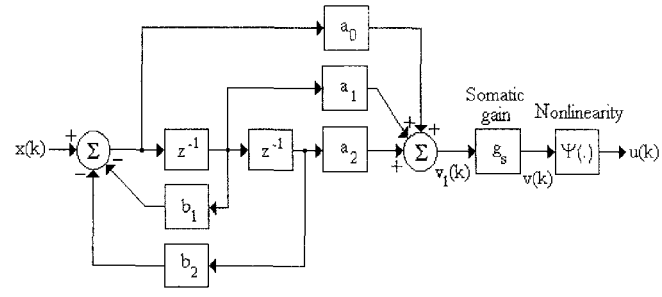

Fig. 6.1 Structure of a single dynamical neural unit

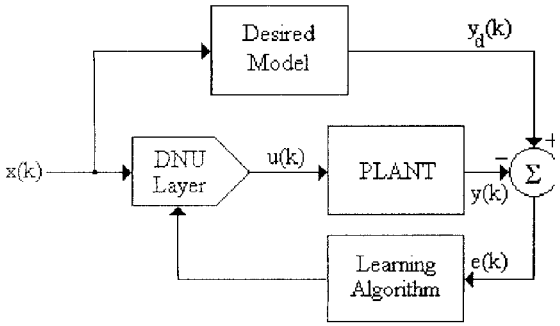

Fig. 6.2 Control system architecture with DNU based controller network

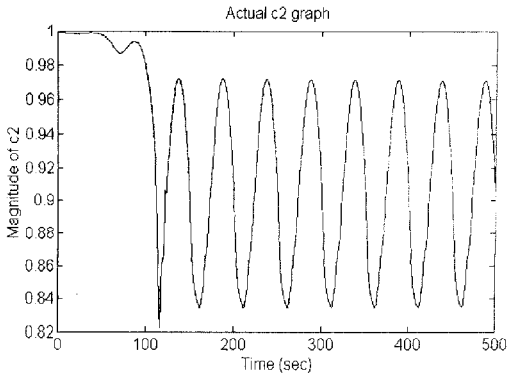

Fig. 6.3(c) $c_2(t)$ Output of the bioreactor plant

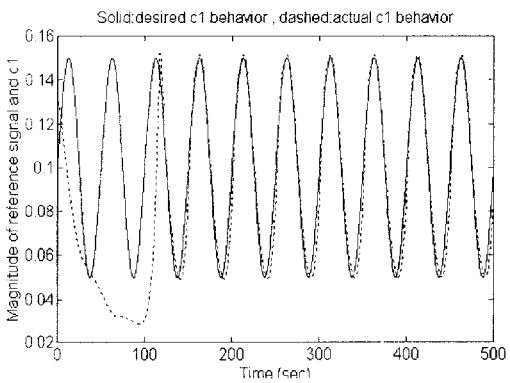

Fig. 6.3(a) $c_1(t)$ output of the bioreactor

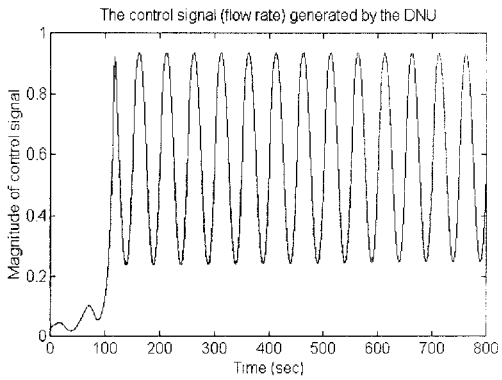

Fig. 6.3(d) Control signal applied to the bioreactor

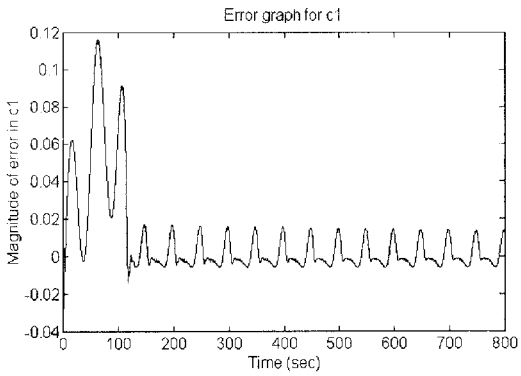

Fig. 6.3(b) Tracking error in $c_1(t)$