Adaptive Recurrent Neural Network Control of Biological Wastewater Treatment

Ieroham S. Baruch,[1,*] Petia Georgieva,[2,†] Josefina Barrera-Cortes,[3,‡]
Sebastiao Feyo de Azevedo[4,§]
[1]*Department of Automatic Control, CINVESTAV-IPN, Ave. IPN No. 2508,
A.P. 14-470, 07360 Mexico D.F., Mexico*
[2]*Department of Electronics and Telecommunications, University of Aveiro,
3810-193 Aveiro, Portugal*
[3]*Department of Biotechnology and Bioengineering, CINVESTAV-IPN, Ave.
IPN No. 2508, A.P. 14-470, 07360 Mexico D.F., Mexico*
[4]*Department of Chemical Engineering, Faculty of Engineering, University of
Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal*

Three adaptive neural network control structures to regulate a biological wastewater treatment
process are introduced: indirect, inverse model, and direct adaptive neural control. The objec-
tive is to keep the concentration of the recycled biomass proportional to the influent flow rate in
the presence of periodically acting disturbances, process parameter variations, and measure-
ment noise. This is achieved by the so-called Jordan Canonical Recurrent Trainable Neural Net-
work, which is a completely parallel and parametric neural structure, permitting the use of the
obtained parameters, during the learning phase, directly for control system design. Comparative
simulation results confirmed the applicability of the proposed control schemes. © 2005 Wiley
Periodicals, Inc.

1. INTRODUCTION

Because of public health, economic, and social issues, the biological waste-
water treatment process is a subject area of world-wide interest. Therefore, in the
relevant literature, several control structures have been recently considered. In

*Author to whom all correspondence should be addressed: e-mail: baruch@ctrl.
cinvestav.mx.
†e-mail: petia@det.ua.pt.
‡e-mail: jbarrera@mail.cinvestav.mx.
§e-mail: sfeyo@fe.up.pt.

Ref. 1, a frequency domain H_∞ controller is designed for the same class of activated sludge process. The robust controller performance is evaluated relative to the observed behavior of a conventional PI controller.[2] In terms of tracking ability, both controllers perform well. However, the PI control action was seen not to be physically feasible. A main obstacle toward practical implementation of the H_∞ robust control is the relatively high order of the final controller. This problem is avoided in Ref. 3, where adaptive Lambda-tracking control is implemented for activated sludge processes. The Lambda tracker is simple in its design, relies only on structural properties of the process and weak feasibility conditions, and acts as a proportional controller with adaptive error-dependent gain.[4] The price being paid is that no zero tracking error is guaranteed. The closed-loop system tolerates a small prespecified tracking error. Therefore, in this article we investigate a new estimation and control paradigm, namely different neural network (NN) control structures, when applied to the same biological wastewater treatment process. The intuition behind this is to benefit from the main network property to capture (and generalize) the process behavior after a certain learning phase relying only on input–output data and finally to overcome the problems experienced by alternative control structures.

The rapid growth of available computational resources led to the development of a large number of neural network-based modeling, identification, prediction, and control applications.[4–6] The ability of NNs to approximate complex nonlinear relationships without prior knowledge of the model structure makes them a very attractive alternative to classical modeling and control techniques.[7–9] Though NN-based techniques have been successfully applied in several engineering areas, a common drawback is that the designed identification NN models are nonparametric, which prevents them from using the obtained information directly for a state-space control system design.[10–12] In a previous article, Baruch et al.[13] applied the state-space approach to design a recurrent NN (RNN) in an universal way, defining a Jordan canonical two- or three-layer RNN model, called a *Recurrent Trainable Neural Network* (RTNN). The NN model has a parametric structure that allows us to estimate the system states and use them for state-space control.[14] In this article this general RTNN approach is implemented to a biological wastewater treatment process.

The article is organized as follows. In Section 2 we introduce a mathematical model of the wastewater treatment process that captures the characteristics of the real process and formulate the control objectives. In Section 3 the architecture and the learning mechanism of a discrete-time Jordan Canonical RTNN that is implemented for process parameter and state estimation are discussed. The indirect, inverse model, and direct adaptive RTNN control structures are introduced in Sections 4, 5, and 6, respectively. Finally, in Section 7 the practical relevance of the proposed control schemes is illustrated by simulations and qualitatively compared.

2. PROCESS DESCRIPTION AND MODELING

Wastewater treatment is performed in an aeration tank, in which the contaminated water is mixed with biomass in suspension (activated sludge), and the
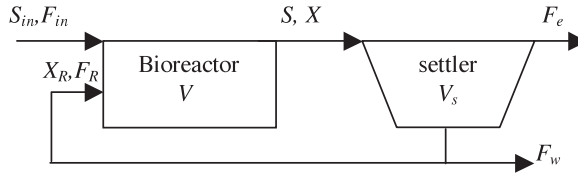
Figure 1.   Biological wastewater treatment with settler.

biodegradation process is then triggered in the presence of oxygen. The tank is equipped with a surface aeration turbine, which supplies oxygen to the biomass and, additionally, changes its suspension into a homogeneous mass. After some period, the biomass mixture and the remaining substrate go to a separating chamber where the biologic flocks (biologic sludge) are separated from the treated effluent. The treated effluent is then led to a host environment. The maintenance of adequate concentration of active biomass in the aeration tank, which allows the aerobic degradation of the incoming wastewater, is achieved by the recirculation of the sludge accumulated in the decanter. The aim is good settling of the biomass in the settler and high conversion of the entering organic material in the bioreactor (Figure 1). The concentration of the biomass in the recycle stream serves as an indicator of both the sludge activity and the sludge settling characteristics, and is therefore considered as the controlled variable.

The main objective of the control system is to keep the recycle biomass concentration close to the reference signal, and this should be achieved in the presence of disturbances and measurement noise acting on the recycle flow rate. The control task is hampered by the strong nonlinearity of the process dynamics, the variations in the reaction kinetics, and by unknown and possibly time-varying process parameters. These considerations are well known and valid for all biochemical processes, but a typical peculiarity of the wastewater treatment system is the (proportional) dependence of the recycle biomass concentration on the influent flow rate, acting as measured disturbance. Because the influent flow rate has generally periodic behavior, the goal is not to keep the recycle biomass concentration constant, but to follow a desired time trajectory, a proportion of the influent flow rate.

A detailed picture of the biological wastewater treatment includes the following phases: (i) biodegradation of organic matter, (ii) nitrification (if applicable), (iii) dissolved oxygen utilization, (iv) sludge sedimentation. A detailed description of all reactions arising in the bioreactor would lead to a high-order model.[15] For the control strategy developed in this work, a simplified reduced order model is sufficient, as far as it preserves the structural properties of the process:

- Considering the availability of the appropriate kinetic model for waste degradation and assuming that oxygen availability and transfer is not a limiting step, only the first (i) and the last (iv) process phases are explicitly considered at the modeling stage.
- Taking into account the objectives underlying the control design, it is not relevant to consider separately all different biomass and substrate concentrations in the recycle bioreactor. Consequently, the activated sludge process can be treated as a single-substrate, single-biomass system.

Table I. Notations.

c_d	Decay rate parameter
$F_{in}(t)$	Influent flow rate (1/h)
$F_R(t)$	Recycle flow rate (1/h)
k_{ref}	Reference proportionality factor
$n(t)$	Measurement noise
$q(t)$	Output–input ratio of the settler
S_{in}	Influent substrate concentration (COD mg/l)
$S(t)$	Substrate concentration (COD mg/l)
T_m	Sensor time constant
$Y(t)$	Yield coefficient
V	Bioreactor volume (l)
$X(t)$	Biomass concentration (mg/l)
X_m	Measured biomass concentration (mg/l)
$X_R(t)$	Concentration of the biomass in the recycle stream (mg/l)
X_{Rref}	Reference signal for X_R (mg/l)
$\mu(t)$	Specific growth rate (1/h)

- The complete continuous-flow model can be obtained combining both the kinetics of the main process streams and the appropriate input–output transport terms. Hereafter, the bioreactor is considered to be perfectly mixed so that the concentration of each component is spatially homogeneous.
- A usual stage of simplification involves accepting quasi-steady-state assumptions applied to substrates and products, which are characterized by fast dynamics.[16]

The notations used in this part are given in Table I.

The overall set of equations describing the process is established in the following subsections.

2.1.   Mass Balance to the Bioreactor

$$\dot{X}(t) = \left(\mu(S) - \frac{F_{in}(t) + F_R(t)}{V} - c_d(t)\right)X(t) + \frac{F_R(t)}{V}X_R(t) \qquad (1.1)$$

$$\dot{S}(t) = -\frac{1}{Y}\mu(S)X(t) + \frac{F_{in}(t)}{V}S_{in} - \frac{F_{in}(t) + F_R(t)}{V}S(t) \qquad (1.2)$$

The specific growth rate, $\mu(\cdot)$, which is the key parameter for description of biomass growth and substrate consumption of the reaction, is modeled by a Monod-type equation:

$$\mu(S(t)) = \frac{\mu_m(t)S(t)}{K_m(t) + S(t)} \qquad (2)$$

where $\mu_m(\cdot)$ is the maximum growth rate and $K_m(\cdot)$ is the half-saturation constant of biodegradable organic matter. It is the concentration of the substrate for the specific growth rate equal of $\mu = \mu_m/2$. Both parameters are subject to variations.

2.2.   Mass Balance to the Settler

It is supposed that none of the biomass is left in the effluent F_e of the settler (see Figure 1), so that the whole biomass in the clarifier is settled. The concentration of the biomass in the recycle stream depends on the settler used. The dynamics of the concentration of the biomass in the settler $X_R(t)$ can be described by the following mass balance equation:

$$\dot{X}_R(t) = \left(\frac{F_{in}(t) + F_R(t)}{V_S} \right) X(t) + \left(\frac{F_W(t) + F_R(t)}{V_S} \right) X_R(t) \tag{3}$$

where F_W denotes the waste flow rate and V_s the volume of the settler.

Because the settler has first order dynamics, which are much faster than the bioreactor dynamics, and because we assume that a constant ratio of output to input solids concentration is maintained, we may approximate the settler behavior by

$$X_R(t) = q(t)X(t) \tag{4}$$

where the parameter $q(t)$ is considered as a continuously differentiable and bounded function with bounded inverse, bounded derivative and $q(t) > 1$ for all $t \geq 0$. The biomass concentration in the settler is higher than the biomass concentration in the reactor because it accumulates at the bottom of the vessel, and good settling is only possible if the settler is designed such that $X_R(t) > X(t)$.

2.3.   Process Measurements

It is assumed that measurements are available for the biomass concentration in the recycle stream, either inferred from measurements of the concentration of carbon dioxide and oxygen in the bioreactor or taken from on-line turbid metric measurements. The sensor dynamics is modeled by:

$$T_m \dot{X}_m(t) = -X_m(t) + X_R(t) + n(t) \tag{5}$$

Additionally the sensor and plant dynamics is corrupted by some white Gaussian noise $n(t)$. The specific model we consider for the simulations is obtained after substitution of $X(t)$ from Equation 4 into Equation 1.1 and has the following form:

$$\dot{X}_R(t) = \left(\frac{\dot{q}(t)}{q(t)} + \mu(t, S(t)) - \frac{F_{in}(t)}{V} - c_d + \frac{q(t) - 1}{V} \right) X_R(t) \tag{6.1}$$

$$\dot{S}(t) = -\frac{1}{Y(t)} \mu(t, S(t)) \frac{1}{q(t)} X_R(t) + \frac{F_{in}(t)}{V} S_{in} - \frac{F_{in}(t) + F_R(t)}{V} S(t) \tag{6.2}$$

$$T_m \dot{X}_m(t) = -X_m(t) + X_R(t) + n(t) \tag{6.3}$$

2.4.   Time-Varying Control Reference

A basic assumption is that the process input has a diurnal periodicity. This approach is well suited for plants dealing with domestic wastes whose inputs are known to be consistently periodic. Therefore, the control objective is to ensure that the biomass concentration in the recycle flow tracks asymptotically a time-varying reference signal, which is proportional to the influent flow rate:

$$X_{Rref}(t) = k_{ref} F_{in}(t) \qquad (7)$$

This reference signal is assumed to be measurable.

3.   RTNN FOR PROCESS STATE ESTIMATION

In this section the process states $(X_R(t), S(t), X_m(t))$ are estimated applying a discrete time model of a Jordan canonical RTNN. This network model, introduced in Refs. 13, 14, and 17, has a state-space parametric structure, which permits us to use the estimated states directly for (state-space) control design.[14,17–19]

The general RTNN two-layer architecture has the following mathematical description:

$$X(k + 1) = AX(k) + BU(k) \qquad (8)$$

$$Z(k) = \varphi(X(k)) \qquad (9)$$

$$y(k) = \varphi(CZ(k)) \qquad (10)$$

$$A = block\text{-}diag(A_i); |A_i| < 1 \qquad (11)$$

where $X(\cdot)$ is an n-state vector, $U(\cdot)$ is an m-input vector, $y(k)$ is an l-output vector, and $Z(\cdot)$ is an auxiliar vector variable with dimension n. Equations 8 and 9 define the hidden layer of the RTNN, and by Equation 10 the feedforward output layer is represented. The matrix A is the feedback weight matrix of the hidden layer, which has an $(n \times n)$ block-diagonal structure with (1×1) blocks; the matrices B and C are $(n \times m)$ and $(l \times n)$ weight input and output matrices, respectively, with structure, corresponding to the structure of A. The inequality 11 is the additional model stability condition imposed on the elements of A, which is in fact stability constrains on discrete-time system eigenvalues. The function $\varphi(\cdot)$ is a vector-valued activation function with hyperbolic tangent elements, given by

$$\varphi[Z] = \tanh(Z) = \frac{1 - e^{-2(Z)}}{1 + e^{-2(Z)}} \qquad (12)$$

The main advantage of the proposed two-layer Jordan canonical RTNN architecture is that it is a universal hybrid neural model that contains one feedforward output layer and one recurrent hidden layer with completely decomposed dynamics, as the matrix A is block-diagonal. Hence, it has a minimum number of parameters and is a completely parallel structure as the Jordan canonical form is parallel with respect to a regressive sequential model. The RTNN architecture is described

in a state-space form (SISO or MIMO) and serves as a one-step-ahead state predictor/estimator; therefore it is suitable for identification and control purposes.

The tuning of the network weights is based on the dynamic Back Propagation (BP) algorithm for RTNN, which is derived using the sensitivity model.[20] The most general BP updating rule is described by

$$W_{ij}(k+1) = W_{ij}(k) + \eta\Delta W_{ij}(k) + \alpha\Delta W_{ij}(k-1) \tag{13}$$

where W_{ij} is the (i,j)th element of each weight matrix A, B, C of the RTNN model; ΔW_{ij} is the (i,j)th weight correction of W_{ij}, and it corresponds to an element in ΔC_{ij}, ΔA_{ij}, or ΔB_{ij}; η is the learning rate parameters $(0 < \eta < 1)$; and k is the iteration number. The last term in Equation 13, also denoted as the momentum term, is added to prevent big oscillations in the error during the learning. Thus, α is the momentum term learning rate parameter $(0 < \alpha < 1)$. Practical considerations for the choice of η and α are the expressions $\eta = ab$ and $\alpha = a(1 - b)$ for a, $b \in (0,1)$, which establish an inverse relation between the two learning rate parameters. The weight matrix updates ΔC_{ij}, ΔA_{ij}, and ΔB_{ij} are given by

$$\Delta C_{ij}(k) = [y_j(k) - \tilde{y}_j(k)]\varphi'(y_j(k))Z(k) \tag{14}$$

$$R = C_i(k)[y_j(k) - \tilde{y}_j(k)]\varphi'(Z_i(k)) \tag{15}$$

$$\Delta A_{ij}(k) = Rx_i(k-1) \tag{16}$$

$$\Delta B_{ij}(k) = Ru_i(k) \tag{17}$$

where y_j is the jth element of the target vector (the actual process output) and $\tilde{y}_j$ is a jth element of the RTNN output. Note that the elements of A_{ij} are restricted, as is indicated in inequality 11, in order to ensure the RTNN stability during the learning, taking into account that $\varphi'(y(k)) = 1 - y^2(k)$ for the hyperbolic tangent activation function. The stability, controllability, and observability of the RTNN and the BP of its learning are proved in Ref. 17. Using this algorithm, the training of the neuro-identifier (see Figure 2) is accomplished via an on-line learning scheme in order to minimize the norm of the instantaneous identification error. The identification error e_{ident} and the performance index E_{ident} to be minimized are

$$e_{ident}(k) = y_p(k) - \tilde{y}_n(k) \tag{18}$$

$$E_{ident}(k) = \frac{1}{2}e_{ident}^2(k) \tag{19}$$

The obtained RTNN model has inherent robustness properties mainly due to the BP learning method adopted and guaranteed stability because the stability condition is checked at each step of the network learning process. When the convergence is reached and the identification error decreased to small values, it is admitted that the RTNN model could substitute the plant model and it could be used for neural control system design.
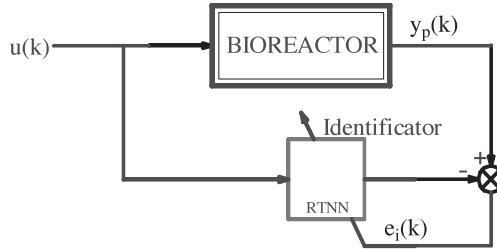
Figure 2.   Block diagram of the real-time process identification by RTNN.

4.   INDIRECT ADAPTIVE NEURAL CONTROL

In this section the indirect adaptive neural control[14] is discussed when the objective is for the output to follow a time-varying reference. The tracking control problem is to design a controller that asymptotically reduces the error between the process output and the reference signal.

As the hyperbolic tangent activation function is odd, smooth, and bounded (see Equation 12), the RTNN model[8–11] is linear in small deviations and nonlinear in large ones from the equilibrium point zero.[13,14] Hence, this model can be linearized as:

$$X(k+1) = AX(k) + BU(k) \qquad (20)$$

$$Y(k) = CX(K) \qquad (21)$$

The model 20, 21 gives a locally stable controllable linear state-space model of the process, which is improved iteratively during the network learning stage. The linearized RTNN is given in state-space Jordan canonical form; therefore, it is a parametric one. The optimized parameters A, B, C, obtained during the learning phase, are implemented in the indirect adaptive tracking control system design. Once the output model 21 is rewritten for the $(k+1)$th step and the state variable $X(k+1)$ is substituted by model 20, it yields

$$Y(k+1) = CX(k+1) \qquad (22)$$

$$Y(k+1) = CAX(k) + CBU(k) \qquad (23)$$

Let the reference setpoint signal be denoted by $r(k)$, then according to the design method in Ref. 14, the control law is defined as:

$$U(k) = (CB)^{-1}\{-CAX(k) + r(k+1) + \gamma[r(k) - y_p(k)]\} \qquad (24)$$

where A, B, C, and $X(k)$ are generated from the RTNN during the learning stage. The matrix product (CB) is supposed to be nonsingular due to the controllability
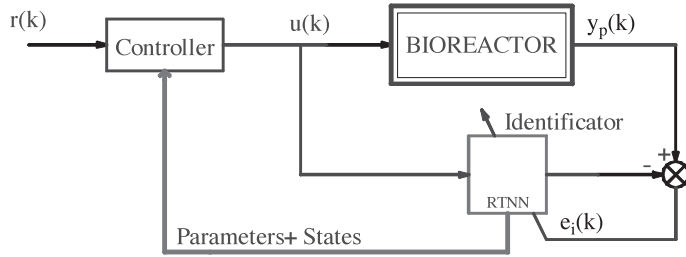
Figure 3.   Block diagram of the indirect adaptive RTNN control system.

and the observability of the RTNN model.[17] The substitution of 24 in 23 gives the following difference equation for the system error:

$$E(k + 1) + \gamma E(k) = 0 \tag{25}$$

Hence, the stability of the Equation 25 depends on the value of the control gain parameter γ. The absolute value of this parameter has to be less than 1. The block diagram of the indirect adaptive control system is depicted in Figure 3.

The control structure contains one identification and state estimation RTNN, which generates the system parameters and states necessary for the controller to compute the adequate control in real time (for more details, see Ref. 14). In fact once the process states are reliably provided by the network model, the next step is to implement the classical full state feedback control, details for which can be found elsewhere. In Section 7 this control scheme is tested with the wastewater treatment case study.[4]

5.   INVERSE PLANT MODEL ADAPTIVE NEURAL CONTROL

The inverse plant model is used as a dynamic system compensator to obtain a unity transfer function of the closed-loop system.[18] Thus, the system is able to follow the reference signal. The block diagram of this adaptive neural control is depicted in Figure 4. It contains one inverse plant model feedforward controller RTNN, which is trained by the error between the reference signal and the output of
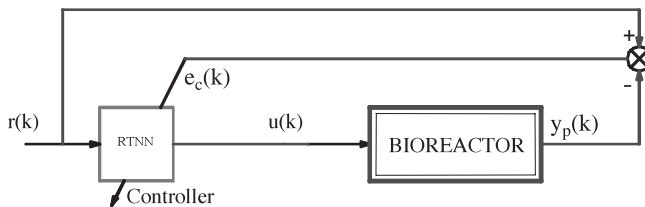

Figure 4.   Block diagram of the inverse plant model adaptive RTNN control system.

the plant. So the control system could ensure the reference signal tracking, and the control signal is given by the following equation:

$$U(k) = N(r(k)) \tag{26}$$

where $N(r(k))$, depending on the reference signal, is the inverse model feedforward control, generated by the RTNN.

6. DIRECT ADAPTIVE NEURAL CONTROL

In this section a direct adaptive neural control structure[14,19] is presented. The block diagram is shown in Figure 5.

Note that it is the most complex control scheme, which contains tree networks. The RTNN-1 is the plant model identificator and state estimator, which generates the state vector supplied as an input to the second network. The RTNN-2 is a feedback stabilizing system controller. The RTNN-3 is an inverse system neural model (a feedforward precompensator) that ensures the ability of the system output to track the time-varying reference trajectory. The identificator RTNN-1 is trained by the error between the plant output and the computed RTNN-1 output. The two control networks are trained by the control error between the reference signal and the plant output. The control law is defined as:

$$U(k) = -N_1(X(k)) + N_2(r(k)) = U_{ff}(k) + U_{fb}(k) \tag{27}$$

where $N_1(X(k))$, depending on the system states, is the feedback control $U_{fb}(k)$ generated by RTNN-2; $N_2(r(k))$, depending on the reference signal, is the feedforward control $U_{ff}(k)$ generated by RTNN-3.

These control schemes have already been successfully applied for kinematics engineering problem,[14] for DC motor control,[17] for fed-batch fermentation processes,[18,19] and they are also positively experimentally used for the activated sludge process in the next section.
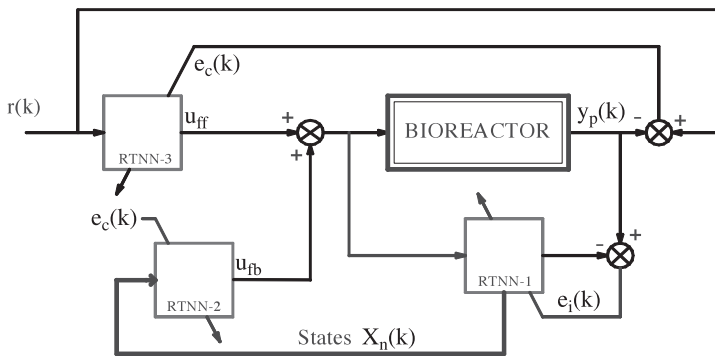

Figure 5. Block diagram of the direct adaptive RTNN control system.

7. SIMULATION RESULTS

The specific model considered for process simulation is the system of nonlinear algebraic differential equations 2 and 6 with constant parameters:

$$V = 1.5 \cdot 10^7 \text{ (l)}, \qquad S_{in} = 300 \text{ (mgCOD/l)}, \qquad T_m = 1/12 \text{ (h)} \qquad (28)$$

The model uncertainties are taken into account by introducing time-varying parameters as:

$$\mu_m(t) = 0.2 + 0.1 \sin(2\pi t/3 + 4\pi/3) \qquad (29)$$

$$K_m(t) = 90 + 30 \sin(\pi t/2) \qquad (30)$$

$$Y(t) = 0.6 + 0.1 \sin(\pi t/3 + \pi/3) \qquad (31)$$

$$q(t) = 4 + \sin(\pi t/6) \qquad (32)$$

$$c_d(t) = 10^{-4}(25 + 5 \sin(\pi t/12)) \qquad (33)$$

The control objective is to track the reference signal, given by Equation 7, where the parameters are as follows:

$$k_{ref} = 3.8 \cdot 10^{-3} \text{ (mgh/l}^2\text{)} \qquad (34)$$

$$F_{in}(t) = 3 \cdot 10^6 \text{ (1 + 0.25 sin } \pi t/12) \qquad (35)$$

The above data coincide with the typical range for domestic wastewater and are chosen identical to those in Ref. 2. We keep these data for all of the following simulations, and the initial conditions are always set to:

$$S(0) = 8 \text{ (mgCOD/l)}, \qquad X_R(0) = 11.4 \cdot 10^3 \text{ (mg/l)}, \qquad X_m(0) = 0 \text{ (mg/l)} \qquad (36)$$

To overcome saturation of the RTNNs, the output and the input of the plant are transformed by the following scale factors:

$$y_p = (X_m - 11400)/5700 \qquad (37)$$

$$F_R = [([(U * 7.5 \times 10^5) + 3 \times 10^6]F_{Conv}) - X_m]K_{stab} \qquad (38)$$

where the scaling parameters are given by:

$$K_{stab} = 3 \cdot 10^{-3}, \qquad F_{conv} = 0.0038 \qquad (39)$$

The variable U is the bioreactor input control signal, issued by one of the proposed control algorithms; F_R is the plant control input variable, specified in Table I and defined above; y_p is the scaled output of the bioreactor; and X_m is the controlled plant measured output variable, specified in Table I and defined above. This scale factor coincides with the typical range of the domestic wastewater and the range of the reference signal. Also, to preserve the stability of the plant, the measured plant output is feedbacked to the plant input by a small gain. The substitution of 34 and 35 in Equation 7 and the application of the transformation on it give the following expression for the scaled reference signal:

$$r(k) = 0.5 \sin\left(\frac{\pi k}{12}\right) \tag{40}$$

The inverse transformation of Equation 37 is given by:

$$X_m = 5700 y_p + 11400 \tag{41}$$

The substitution of Equations 39 and 41 in 38 finally yield:

$$F_R = (0.5U - y_p)1.71 \cdot 10^8 \tag{42}$$

This equation expresses the plant input variable F_R as a function of the control variable U, issued by whichever of the above described tree schemes of neural control and the small gain stabilization feedback, with respect to the scaled plant output variable y_p. Equation 42 coincides also with the range of the reference signal. The simulation is done using the following plant equations: the process plant equations 6.1–6.3; the Monod-type equation 2; the time-variable plant parameters 29–33; the plant output scaling equation 37; the scaled reference signal equation 40; the scaled plant input equation 42, and each of the three control scheme equations 24, 26, and 27, respectively. In all simulations, the severe realistic conditions such as measurement noise are taken into account by generating a stochastic signal added to the process output with variance 1200, which is commonly used to simulate noisy measurement. The variance chosen corresponds to 6% noise on the data. The process is simulated over a period of 48 h, which gives an idea about its periodic behavior (a typical period is about 24 h) and the discretization step is set at $T_0 = 0.01$ h. For sake of comparison the RTNN estimation and control structures are designed with the same learning parameters, which, after a number of trials, were fixed at $\alpha = 0.01$, $\eta = 0.95$. The activation functions of the hidden and output network layers are hyperbolic tangent (Equation 12).

7.1.   RTNN Process State Estimation

In Figure 6 are depicted the results of the RTNN process state estimation. The network topology is (1, 3, 1); that is, it has one hidden layer with three neurones. RTNN is provided on-line by the measured process input, which serves as the network input, and the measured process output, which serves as the network target. The network learns adaptively the process dynamics, and after approximately 2.5 h, the mean squared error (MSE) becomes less than 1%. Note that the estimation converges fast in case of noisy measurements.

7.2.   Indirect Adaptive RTNN Control

The results, obtained from the indirect adaptive neural control scheme, are given in Figure 7. The identification RTNN has a topology (1, 3, 1) and the control parameter is set to $\gamma = 0.1$. The results show a good convergence of the system output to the desired trajectory after approximately 2.1 h. The graphics of the learned
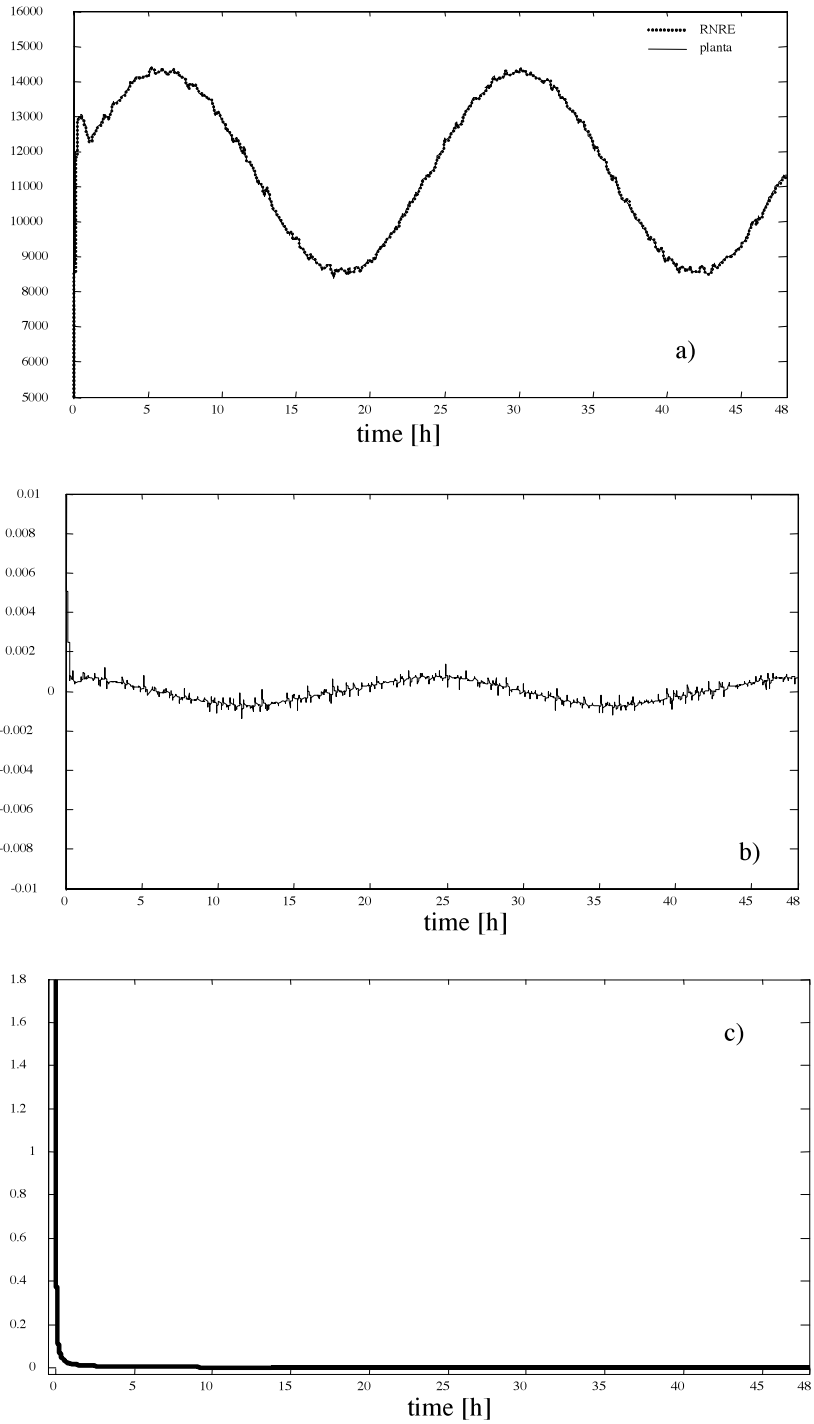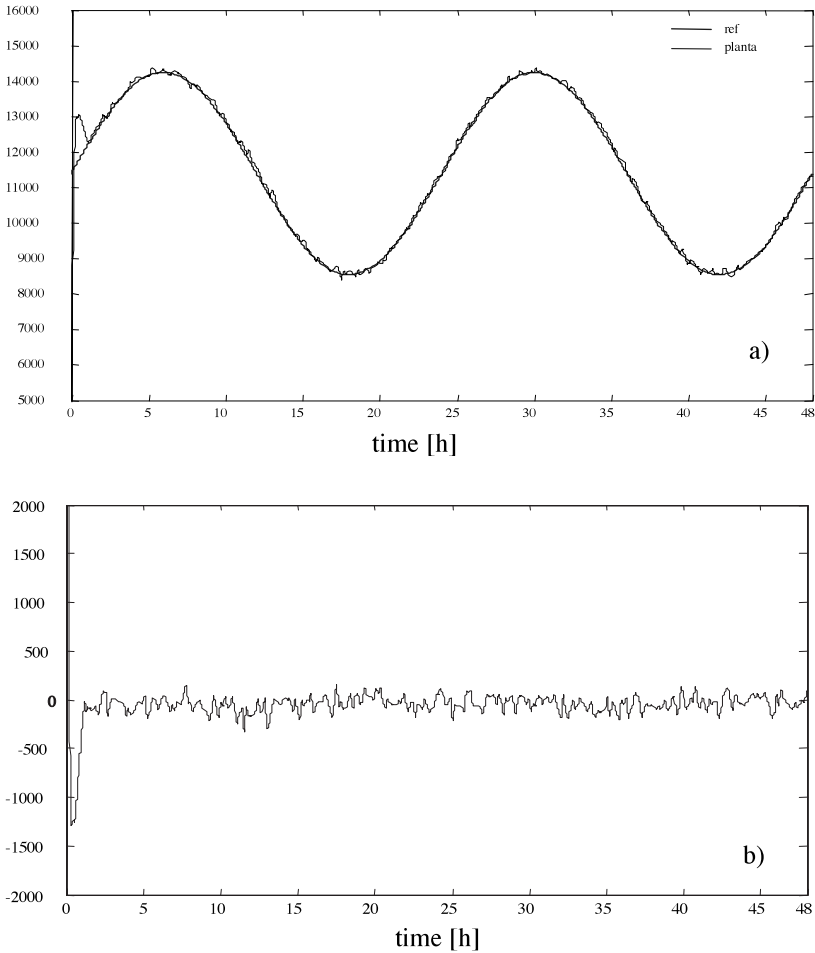
Figure 6. Graphical results for systems identification: (a) Comparison between the plant and RTNN outputs; (b) Instantaneous error of plant identification; (c) MSE of plant identification.
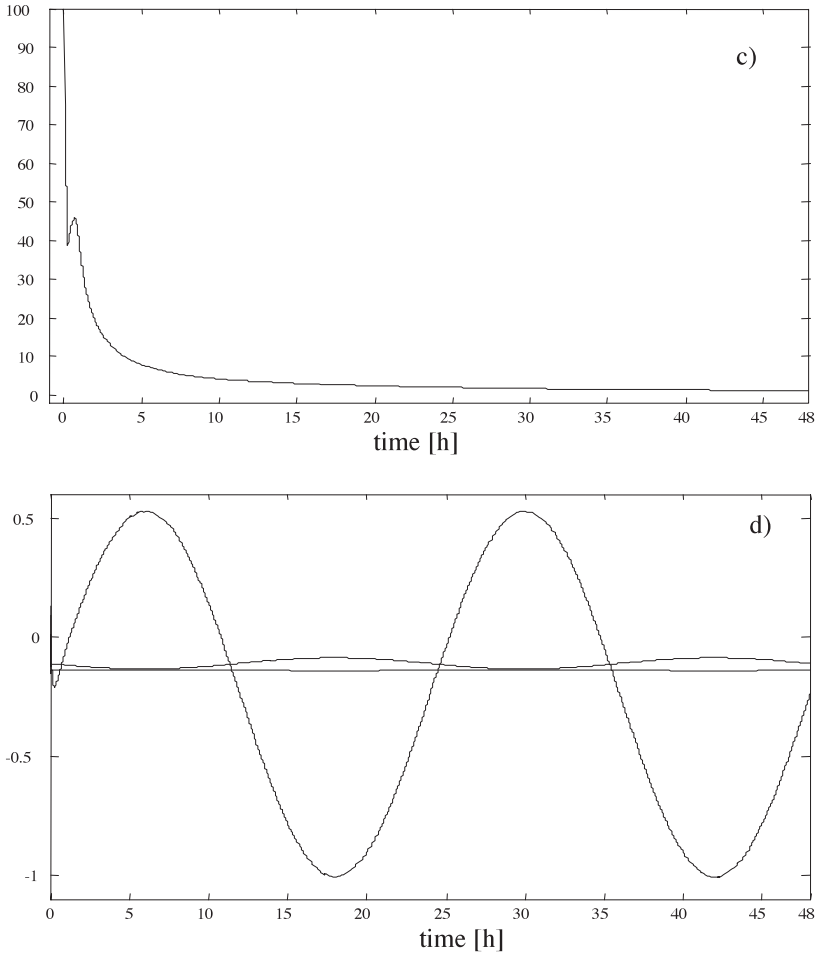
Figures 7a and 7b. Graphical results for indirect adaptive RTNN control: (a) Comparison between the plant output and the reference signal; (b) Instantaneous error of control.

matrix parameters *A*, *B*, and *C* are omitted here because they are constant due to the small MSE (in percent), which is below 2%.

7.3. Inverse Model Adaptive RTNN Control

The results, obtained from the inverse model adaptive neural control scheme, are given in Figure 8. The topology of the identification RTNN is (1, 3, 3, 1) and the control RTNN has a topology (1, 3, 3, 1). Note that both networks contain two hidden layers (one recurrent and one feedforward) and one feedforward output layer. The results show a good convergence of the system output to the desired trajectory after approximately 2 h with the MSE of tracking less than 2%.

Figures 7c and 7d. Graphical results for indirect adaptive RTNN control: (c) MSE of control; (d) States X of the RTNN, obtained from the identification.

7.4.   Direct Adaptive RTNN Control

The results obtained by the direct adaptive neural control scheme are depicted in Figure 9. The identification RTNN has a topology (1, 6, 1) and the two control RTNNs have topologies (6, 6, 1) and (1, 6, 1), respectively. The results show a good convergence of the system output to the desired trajectory after approximately 3.2 h and the MSE is also below 2%.

7.5.   Adaptive λ-Tracking Control

For sake of comparison, in Figure 10a, b are given simulation results with the adaptive λ-tracking control.[4] The MSE obtained is below 2%.
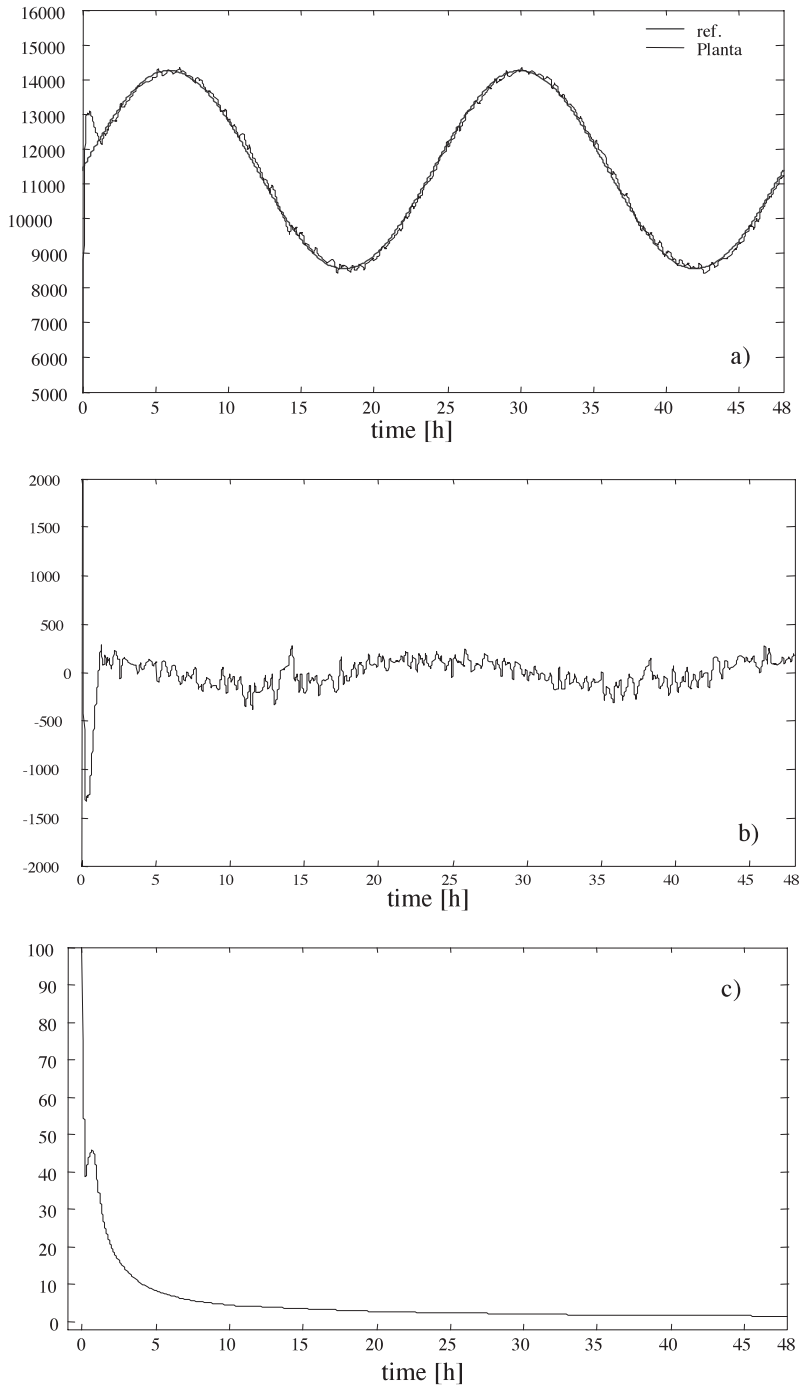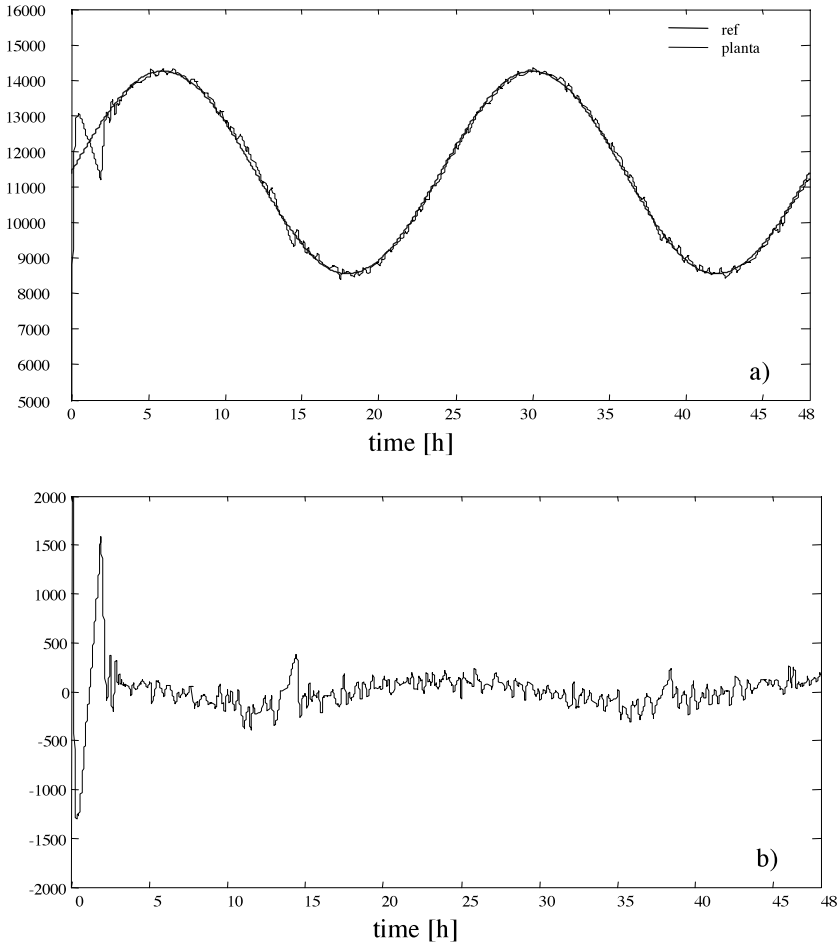
Figure 8. Graphical results for inverse plant model RTNN control: (a) Comparison between the plant output and the reference signal; (b) Instantaneous error of control; (c) MSE of control.

Figures 9a and 9b. Graphical results for direct adaptive RTNN control: (a) Comparison between the plant output and the reference signal; (b) Instantaneous error of control.

A comparison between the four control schemes is done using the MSE values at the end of each simulation, which are summarized in Table II.

So, the adaptive neural control systems have more precision with respect to the adaptive λ-tracking control, but this is paid for with a more complex control scheme. The comparison between the adaptive neural control schemes gives some priority to the indirect adaptive RTNN control, although the inverse model adaptive RTNN control performs better, particularly during the initial transient period. It exhibits a lower degree of overshoot than the indirect adaptive scheme. As for the direct adaptive RTNN control, it is subject to a considerably vivid initial transient period. This behavior is due to the noisy process output, which is directly involved in computing the network training error signal $(e_c(k) = y_p(k) -$
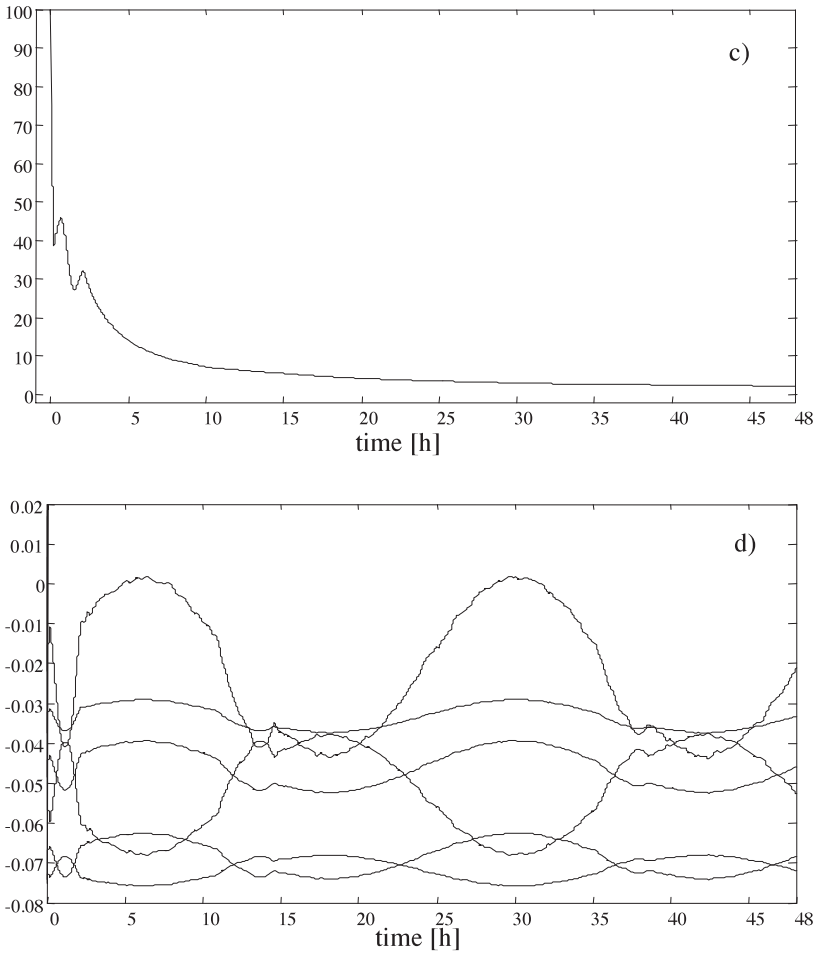
Figures 9c and 9d. Graphical results for direct adaptive RTNN control: (c) MSE of control; (d) States X of the RTNN-1, obtained from the identification.

$r(k)$). A possible way to overcome this problem would be to increase the controller gain, but in this case the physical actuator constraints might become active. So there is a trade-off between good tracking performance and control input limitations.

In fact, apart from the final MSE, the inverse model adaptive RTNN control demonstrates the best dynamical performance for the process considered, which is due to the way the controller network is trained $(e_c(k) = y_{RTNN1}(k) - r(k))$. Assuming the identification network reliably estimates the process states, it also filters the high-frequency measurement noise, and the overall control scheme becomes sufficiently robust against process noise and perturbations.
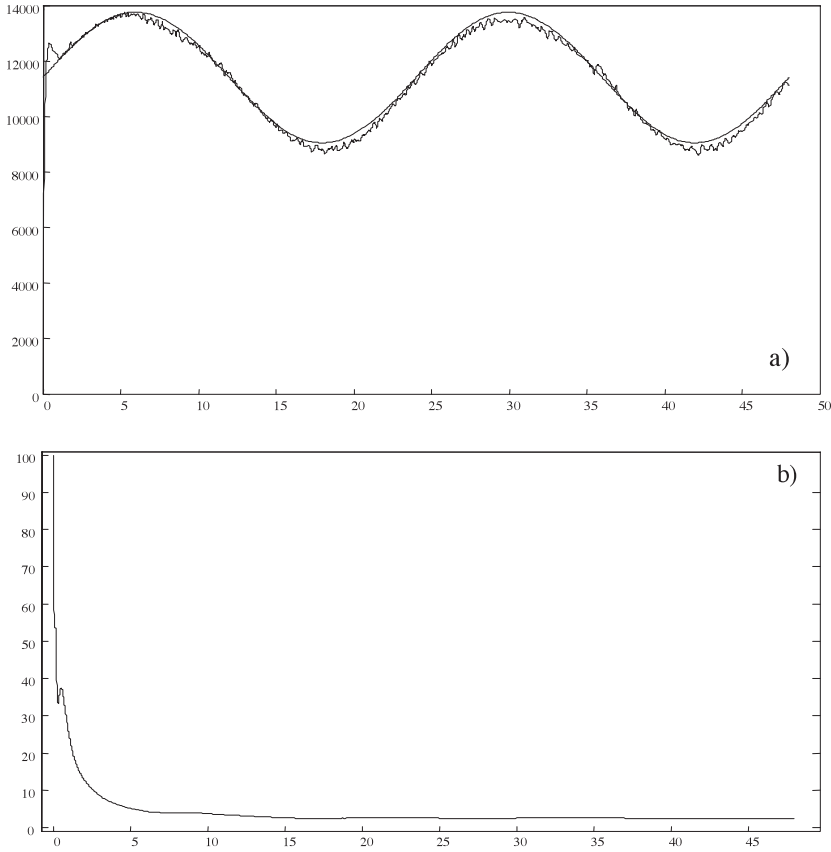
Figure 10.   Graphical results for the adaptive λ-tracking control: (a) Comparison between the plant output and the reference signal; (b) MSE of control.

8.   CONCLUSIONS

In this article we report the application of three recurrent trainable neural network models for real-time identification and state feedback/feedforward control of biological wastewater treatment: an indirect, an inverse model, and a direct

Table II.   Mean squared error (MSE) at the end of each run.

Control structure	MSE (%)
Indirect adaptive RTNN control	1.2
Inverse plant model adaptive neural control	1.4
Direct adaptive neural control	2
Adaptive λ-tracking control	2.4

adaptive neural control. The proposed RTNN models have a Jordan canonical structure, which permits us to use the generated vector of estimated states directly for process control. A back-propagation through time learning algorithm is implemented for RTNN training. The experimental results confirm the applicability of the described identification and control methodology in practice. The neural controllers are able to force the system in tracking a time-varying process-dependent reference signal for the concentration of biomass in the recycle stream. They perform well under restrictive conditions of periodically acting disturbances, parameter uncertainties, and inevitable sensor dynamics.

The adaptive neural control has the following advantages with respect to existing classical control solutions: (i) It is based on more complete process information, obtained from the identification and state estimation neural network, therefore, it is expected to give more precise and tolerant tracking; (ii) the application of RTNN control does not require a linearization of the plant model, it exploits directly the nonlinear process dynamics; (iii) the output of the RTNN process model can substitute direct measurements of the plant output, which are often very expensive or even not possible.

The comparison between the results obtained with the adaptive neural control and the method of λ-tacking shows that the neural control gives better results. A comparative study among the three neural control structures led to the conclusion that the indirect and the inverse model adaptive RTNN schemes give qualitatively similar results. Nevertheless, the latter structure is more favorable with respect to rejecting sensor noise and output oscillations due to unknown perturbations.

Acknowledgments

References

1. Georgieva P, Feyo de Azevedo S. Robust control design of an activated sludge process. Int J Robust Nonlin Control 1999;9:949–967.
2. Schaper C, Mellichamp D, Seborg D. Robust control of a wastewater treatment system. In: Proc 29th Conf on Decision and Control, Honolulu, Hawai'i, December 1990. pp 2035–2040.
3. Georgieva P, Ilchmann A, Weirig MF. Modelling and adaptive control of aerobic continuous stirred tank reactors. Eur J Control 2001;7:1–16.
4. Georgieva P, Ilchmann A. Adaptive λ-tracking control of activated sludge processes. Int J Control 2001;74:1247–1259.
5. Gorinevsky D. Sampled-data indirect adaptive control of bioreactor using affine radial basis function network architecture. Trans ASME J Dyn Syst Measure Control 1997; 119:94–97.
6. Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ. Neural network for control systems—A survey. Automatica 1992;28:1083–1112.

7. Zhao H, Hao OJ, McAvoy TJ. Approaches to modelling nutrient dynamics: ASM2, simplified model and neural nets. Water Sci Technol 1999;39:227–234.
8. Norgaard M, Ravn O, Poulsen NK, Hansen LK. Neural networks for modelling and control of dynamics systems. London: Springer-Verlag; 2000.
9. Yu DL, Gomm JB. Implementation of neural network predictive control to a multivariable chemical reactor. Control Eng Pract 2003;11:1315–1323.
10. Nikravesh M, Farell AE, Stanford TG. Control of nonisothermal CSTR with time varying parameters via dynamic neural network control (DNNC). Chem Eng J 2000;76:1–16.
11. Nascimento CAO, Giudici R, Guardani R. Neural network based approach for optimisation of industrial chemical processes. Comput Chem Eng 2000;24:2303–2314.
12. Chen J, Huang TCh. Applied neural network to on-line updated PID controllers for nonlinear process control. J Process Control 2004;14:211–230.
13. Baruch I, Stoyanov I, Gortcheva E. Topology and learning of a class RNN. Elektrik 1996;4(Suppl.):35–42.
14. Baruch I, Flores JM, Thomas F, Garrido R. Adaptive neural control of nonlinear systems. In: Dorffner G, Bischof H, Hornik K, editors. Artificial neural networks—ICANN 2001. Lecture Notes in Computer Science 2130. Berlin: Springer; 2001. pp 930–936.
15. Sundstrom DW, Klei HE. Wastewater treatment. Englewood Cliffs, NJ: Prentice-Hall; 1979.
16. Kokotovic P, Khalil HK, O'Reilly J. Singular perturbations methods in control: Analysis and design. London: Academic Press; 1986.
17. Baruch I, Flores JM, Nava F, Ramirez IR, Nenkova B. An advanced neural network topology and learning, applied for identification and control of a D.C. motor. In: Proc 1st Int IEEE Symp on Intelligent Systems, Varna, Bulgaria, September 2002. pp 289–295.
18. Baruch I, Barrera J, Perez M J, Hernandez P LA. An adaptive neural control of a fed-batch fermentation processes. In: Proc IEEE Int Conf Control Applications, Istanbul, Turkey, June 23–25, 2003. pp 808–812.
19. Baruch I, Barrera-Cortes J, Herhandez LA, Nenkova B. A direct adaptive neural control scheme of fed-batch fermentation process. In: Proc Int Conf on Automatics and Informatics, Sofia, Bulgaria, October 6–8, 2003. pp 219–223.
20. Narendra KS, Parthasarathy K. Gradient methods for the optimization of dynamical systems containing neural network. IEEE Trans Neural Netw 1991;2:252–262.