Neural Network Based Control of a Cement Mill by means of a VSS Based Training Algorithm

Uğur YILDIRAN[1], STUDENT MEMBER, IEEE, and M. Okyay KAYNAK[2], SENIOR MEMBER, IEEE

[1,2]UNESCO Chair on Mechatronics, Bogaziçi University, 80815, Bebek, Istanbul, Turkey,
email: (yildirau, kaynak)@boun.edu.tr

Abstract – **In this study, we investigate a neuro-control scheme proposed in the literature, which uses techniques from variable structure systems (VSS) theory in order to robustify learning dynamics, for control of nonlinear systems. Gaussian Radial Basis Function Neural Network (GRBFNN) is chosen as the neural network architecture because of its strong adaptation capabilities. By means of an instability analysis, it is shown that this scheme leads to unbounded evolution of the controller parameters in steady state due to presence of noise and uncertainties. A modification on the original adaptation algorithm is proposed in order to alleviate this problem. The simulation studies on a nonlinear cement mill circuit model show that the modified update rule stabilizes the learning dynamics and closed loop system becomes insensitive to parametric changes.**

I. INTRODUCTION

In an industrial process, the performance of the system is determined by some key parameters such as quality of the product, power consumption and efficiency, which must be maintained appropriately. From control systems theoretic point of view, these factors highly depend on the control strategy applied and can be substantially improved by a proper choice of the architecture and the design technique. One can see many different control techniques employed that have the goal of attaining maximum system performance. However, they are generally based on classical control methods and thus may lead to unsatisfactory results in practice. This stems from the contradicting nature of classical strategies and industrial process systems. On the one hand, the classical methods generally require a precise description of the system, rely on some restrictive assumptions such as linearity of the plant and may lead to inefficiencies in presence of ambiguities. On the other hand, the process systems are very complex, therefore difficult to obtain an accurate model of, inherently nonlinear and involve ambiguities. An alternative approach that can account for these problems is offered by neuro-control schemes that use artificial neural networks (ANN) to mimic the complex problem solving capabilities of intelligent organisms. ANN with their ability to learn from samples can utilize the available data about past history of a process, allow the application of nonlinear controller design techniques to a broad range of systems and adapt to changes in the environment which frequently occurs in practice.

There exist two different learning approaches that can be used for a neuro-control systems, namely offline and online learning. In the former, the weights of the neural network based controller are updated prior to the control application, using some available past data, while, in the latter the learning algorithm utilizes the information acquired on the fly. This enables such controllers to adapt to the changes in the system and makes them more attractive for control of industrial processes that generally involve nonlinearities and time variations. However, in online training, due to complexity of the dynamics imposed by the parameter adjustment mechanism, system response may not converge to its desired value and become sensitive to uncertainties. Therefore, one can encounter with safety problems in practical applications. Incorporating techniques from variable structure systems theory into training procedure can in some ways alleviate these problems. In [1], Sira-Ramirez et al. propose a sliding mode control (SMC) based training algorithm in discrete-time framework that robustify learning dynamics. This algorithm is utilized in the subsequent research [2]-[3] for inverse identification and adaptive control purposes. Later on, a similar technique is proposed for continuous-time adaptation of linear parameters in flexible structures by enforcing learning error dynamics to a sliding regime [4]. Parma et al. [5] introduce another method for the training of feedforward neural networks having nonlinearities at the output layer neurons.

The robustness property of a SMC based training algorithm makes it a good candidate for use in neuro-control systems in order to decrease the sensitivity of system response to uncertainties. However, this strategy requires a measure of the learning error, which is not directly available in control applications. In conventional neuro-control schemes, the prevailing approach to eliminate the unavailability of learning error has been to utilize an intelligent architecture as the identification model and to use the gradient information obtained through the model in order to determine the search direction in the parameter space. However, this approach has some serious drawbacks such that the identification procedure results in a great amount of computational burden, which restrict practical applicability of the strategy, and more importantly it may not be employed in a continuous time framework. A remedy to these problems is first proposed by Efe [6]. In his work, it is assumed that there exist an error relation between the sliding line at the output of the plant and the learning error at the output of the neural network based controller. Then, this relation is utilized to attain the necessary error measure for the learning algorithm instead of a complex identification model.

0-7803-7369-3/02/$17.00 © 2002 IEEE

In this study, we utilize the aforementioned neuro-control strategy, which relies on the existence of an error relation, as the core approach for control of a cement mill circuit. The strategy requires the mathematical model of the system to be in controllable canonical form. However this is not the case for the process model in hand. Therefore, the system is separated into two first order subsystems for each of which a different controller is used. In each controller, a Gaussian Radial Basis Function Neural Network (GRBFNN) is utilized as the neural network architecture, due to its strong adaptation capabilities.

Another issue related with the neuro-control strategy introduced in [6] is that the parameters of the controller may drift in steady state due to the excitation of the update algorithm by noise and uncertainties. One natural way to prevent this problem, which is utilized in the original work, is to turn the adaptation algorithm off when the error is in a close neighborhood of the zero level. However, this method diminishes the adaptation capabilities of the controller. In this work, we introduce another approach that ensures the bounded evolution of the controller parameters in the steady state while not stopping the adaptation process. For this purpose, a parametric instability analysis is given. Then, based on this analysis, a modification to the original algorithm is proposed and it is shown that the new method ensures the parametric stability of the controller in steady state.

The paper is organized as follows: The nonlinear dynamical model describing the cement mill process is introduced in the second section. The third section is devoted to the control strategy employed, where the system configuration, the network structure and the training algorithm are described. Moreover, a parametric instability analysis is also given in this section and based on this, a modification to the original update algorithms is proposed. Simulation studies and their analysis are given in the fourth section. The fifth section is the conclusions, where the results are discussed.

II. CEMENT MILL PROCESS

The schematic representation of a cement mill circuit is shown in Fig. 1. As depicted in the figure, the raw material, clinker, is fed into the mill, where it is ground into fine powder with the rotational movements. Then, the resulting material is passed through the classifier and separated into two classes, product and tailing. While the product leaves the system, the rejected material, tailings, is fed back into the mill for further grinding.

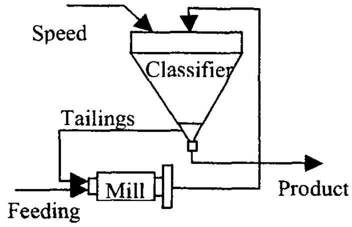

Fig. 1 Cement mill circuit

An experimentally verified nonlinear model for this process is proposed by Magni et al. [7]. The differential equations describing the model can be given as follows

$$T_f \dot{y}_f = -y_f + \frac{K_\alpha}{K_\alpha + \varphi^m(z,d)v^n}\varphi(z,d) \qquad (1)$$

$$\dot{z} = -\varphi(z,d) + u + y_r \qquad (2)$$

$$T_r \dot{y}_r = -y_r + \frac{\varphi^m(z,d)v^n}{K_\alpha + \varphi^m(z,d)v^n}\varphi(z,d) \qquad (3)$$

$$\varphi(z,d) = \max\{0; (-dK_{\varphi 1}z^2 + K_{\varphi 2}z)\} \qquad (4)$$

where y_f is the product flow rate (Tons/h), z is the load in the mill (Tons), y_r is the tailings flow rate (Tons/h), u is the feed flow rate (Tons/h), v is the classifier speed (r/min), and d represents the hardness of the material inside the mill with respect to its nominal value. The parameters of the plant used in this study are shown in Tab. 1, which are the same values with the ones used in [7].

Tab. 1 Parameters of the cement mill process

Parameter	Nominal Value
K_α	$(570)^m(170)^n 0.2667$ (Tons/h)^m r/min^n
$K_{\varphi 1}$	0.1116 (Tons*h)$^{-1}$
$K_{\varphi 2}$	16.50 (h)$^{-1}$
M	0.8
N	4
T_f	0.3 h
T_r	0.01 h
D	1

III. CONTROL STRATEGY

In the following, the neuro-control strategy employed for control of the cement mill process is investigated. First, the configuration of the closed loop system is explained. Then, the neural network architecture utilized as controller is presented and the update equations used to adapt network weights are derived for first order systems based on the techniques proposed in the literature. Finally, a parametric instability analysis for the update equations derived is given and a modification is proposed on them in order to preserve the stability of the overall system in steady state.

A. System Configuration

As can be seen from the nonlinear dynamical equations describing the cement mill process, (1)-(4), the system has three states namely, product flow rate, mill load, tailings flow rate and two inputs, feed flow rate and classifier speed. However, the control structure that we utilize is applicable to single input single output systems as will be explained later. This necessitates the use of a separate controller for each input of the cement mill circuit which should utilize one of the three state measurements as the feedback. Therefore, one can come up with a number of alternative configurations as the control structure. The configuration shown in Fig. 2 is chosen in this study. As depicted in the figure, product flow rate and mill load are chosen as free states and controlled through classifier speed and feed flow rate inputs respectively, while tailing flow rate becomes the dependent state.

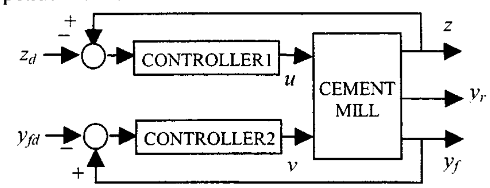

Fig. 2 Control configuration

B. Neuro-control Scheme

In this study, we employ Gaussian radial bases function neural network as the network architecture. This structure has several attractive properties such that it allows the partitioning of the input space, eases the mathematical analysis and its output is a linear function of the output layer weights. The last property is especially important because the update algorithm used in this work requires the adjustable parameter set to be linear with respect to network output.

The input-output relation of a single input single output GRBFNN architecture, the schematic representation of which is depicted in Fig. 3, can be described by the following set of equations.

$$\tau = \underline{w}^T \underline{\varphi}(e) \quad (5)$$

$$\underline{w} = [w_1 \ w_2 \ \dots \ w_r]^T \quad (6)$$

$$\underline{\varphi} = [\varphi_1(e) \ \varphi_2(e) \ \dots \ \varphi_r(e)]^T \quad (7)$$

$$\varphi_i(e) = \exp\left\{ -\frac{(e-\mu_i)^2}{\sigma_i} \right\} \quad (8)$$

In the control loop, the weight vector ($\underline{w}$) of the neural network based controller is updated by means of a SMC based parameter adaptation algorithm introduced by Sira-Ramirez et al. [4]. This adaptation algorithm makes the learning dynamics insensitive against noise and uncertainties, and thus, increases the robustness of the close loop system. The update rule for the training mechanism can be described as follows.

Consider the neural network structure depicted in Fig. 3. The learning error level at the output of the flexible structure is defined as $s_c=\tau-\tau_d$ where τ_d is the desired output sequence. Let us assume that the parameter vector ($\underline{w}$), the output vector of the hidden layer ($\underline{\varphi}$), the time derivative of it ($\underline{\dot{\varphi}}$) and the time derivative of the desired output ($\dot{\tau}_d$) satisfies the following inequalities.

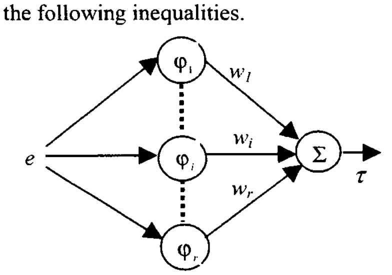

Fig. 3 Gaussian radial basis function neural network

$$\|\underline{w}\| \le N_w, \ \|\underline{\varphi}\| \le N_\varphi, \ \|\underline{\dot{\varphi}}\| \le N_{\dot{\varphi}}, \ |\dot{\tau}_d| \le N_{\dot{\tau}_d} \quad (9)$$

Then, the parameter update mechanism, described in (10) enforces the flexible structure to the sliding surface ($s_c = 0$), and the sliding surface is hit (i.e. the zero learning error level is reached) in a finite time, the upper bound of which is given by (11).

$$\underline{\dot{w}} = -\frac{\underline{\varphi}(e)}{\underline{\varphi}^T(e)\underline{\varphi}(e)} K \operatorname{sgn}(s_c) \quad (10)$$

$$t_h \le \frac{|s_c(0)|}{K-(N_w N_{\dot{\varphi}} + N_{\dot{\tau}_d})} \quad (11)$$

To ensure the occurrence of sliding regime, K must satisfy the following criterion.

$$K > N_w N_{\dot{\varphi}} + N_{\dot{\tau}_d} \quad (12)$$

Unfortunately, because the desired control input, which is the desired output of the flexible structure (τ_d), is not known, the control error (s_c) that is necessary for training of the neural network cannot be evaluated. Therefore, the above mentioned parameter update rule is not directly applicable to control systems. Efe et al. [6] proposes a novel strategy, which is shown in Fig. 4 for a first order system, in order to alleviate unavailability of control error problem. In this work, it is assumed that there exists an relation (Ψ) between the sliding line at the output of the plant (s_p) and the control error and this relation is used to estimate the error at the output of the neural network based controller. Here, we utilize the strategy depicted in Fig. 4 as the core approach and choose the error relation as $s_c = \Psi(s_p) = s_p$. The rationale behind this choice is that it is the simplest function that satisfies the three conditions regarding the error relation, namely region, compability and invertabilitiy conditions, given in [6]. The resulting parameter update rule can be expressed as follows

$$\dot{w}_i = -R_{Gi}(e) K \operatorname{sgn}(e) \quad (13)$$

$$R_{Gi}(e) = \frac{\varphi_i(e)}{\underline{\varphi}^T(e)\underline{\varphi}(e)} \quad (14)$$

A neuro-control scheme with the parameter update algorithm described by (13)-(14) can be used for stabilization of first order nonlinear dynamical systems. However, if the state measurements are corrupted by noise, one can encounter with a serious problem. The presence of small perturbations in steady state error results in continuously excitation of the training algorithm and this can lead to a drift of the network weights. In the sequel, it is shown by means of an instability analysis that the measurement noise leads to unbounded evolution of controller parameters and a modification is proposed on (13)-(14) in order to prevent this problem.

C. Instability Analysis and the Modified Update Rule

In the following, the limiting behaviour of the controller parameters is investigated under the aforementioned training algorithm in the presence of noise and uncertainties. For this purpose we need to make the assumptions given below.

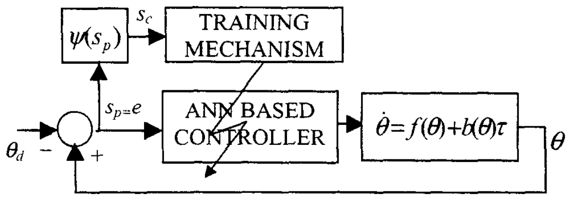

Fig. 4. Intelligent control scheme

$$|e(t)| < \varepsilon, \quad \text{for } \forall t > t_h \tag{15}$$

$$\lim_{t\to\infty} \int_{t_h}^{t} |e(\tau)| d\tau = +\infty \tag{16}$$

Here, (15) states that after a finite hitting time t_h the output of the system gets into a small neighbourhood of the desired output and stays in it thereafter. On the other hand, (16) means that error at the output of the system cannot converge to zero because of noise and uncertainties.

In the remaining part, to prove parametric instability of the training algorithm, it is assumed that one of the controller parameters remains bounded, and then, it is shown that the other parameters must evolve unboundedly. In the light of this reasoning, we can proceed as follows. By integrating both sides of (13), one can obtain the time evolution of the controller parameters expressed in (17).

$$w_i(t) = -K \int_{t_h}^{t} R_{Gi}(e)\,\mathrm{sgn}(e)dt + w_i(t_h) \tag{17}$$

If it is assumed that one of the controller parameters, denoted by w_m, remains bounded, that is

$$N_L < w_m(t) < N_U, \ \text{for } \forall t > t_h \tag{18}$$

one can come up with (19). Please note that in below $R_{Gi}(0) > 0$ because $\varphi_i(0) > 0$.

$$\frac{w_m(t_h) - N_U}{KR_{Gm}(0)} < \int_{t_h}^{t} \frac{R_{Gm}(e)}{R_{Gm}(0)}\mathrm{sgn}(e)dt < \frac{w_m(t_h) - N_L}{KR_{Gm}(0)} \tag{19}$$

On the other hand, by rearranging (17), it can be written in a more suitable form as given in (20)-(22).

$$w_i(t) = -KR_{Gi}(0) \int_{t_h}^{t} V(e)\,\mathrm{sgn}(e)dt + F(t) \tag{20}$$

$$V(e) = \frac{R_{Gi}(e)}{R_{Gi}(0)} - \frac{R_{Gm}(e)}{R_{Gm}(0)} \tag{21}$$

$$F(t) = -KR_{Gi}(0) \int_{t_h}^{t} \frac{R_{Gm}(e)}{R_{Gm}(0)}\mathrm{sgn}(e)dt + w_i(t_h) \tag{22}$$

As can be seen from (21) that $V(0) = 0$. Moreover, by keeping in mind that the centers and the standard deviations of Gaussian activation functions corresponding to different neurons are chosen arbitrarily, we can say that the time derivative of $V(e)$ at the origin is not equal to zero in general (i.e. $\left.\frac{dV(e)}{dt}\right|_{e=0} = \dot{V}(0) \neq 0$). Therefore, in ε-neighborhood of zero error level, the following inequalities hold true,

$$\begin{aligned} V(e)\,\mathrm{sgn}(e) &> \alpha|e|, & \text{if } \dot{V}(0) > 0 \\ V(e)\,\mathrm{sgn}(e) &< -\alpha|e|, & \text{if } \dot{V}(0) < 0 \end{aligned} \tag{23}$$

where α is a positive constant. Now, if one considers $\dot{V}(0) > 0$ and evaluates the limit of (20) as time goes to infinity, the inequality given below can be induced because of the assumption (16) and the inequality (19), which states that $F(t)$ is bounded.

$$\lim_{t\to\infty} w_i(t) < -KR_{Gi}(0)\alpha \lim_{t\to\infty} \int_{t_h}^{t} |e| dt + \lim_{t\to\infty} F(t) = -\infty \tag{24}$$

This tells us that if m[th] weight remains bounded, the other weights must evolve unboundedly. In other words, at most one controller parameter can remain bounded in steady state. Similar results can also be obtained for the case $\dot{V}(0) < 0$, which are not given here because of space limitations.

The key factor that helps us to attain the above given results is the fact that $\dot{V}(0) \neq 0$ for arbitrary choice of neuron parameters. If this inequality can somehow be vanished, the parameter drift problem may possibly be alleviated. In the light of this fact, we propose to replace $R_{Gi}(e)$ terms in the update equations with +1 when the error at the output of the plant gets into the ε-neighborhood of zero error level. Indeed, it can be shown that this strategy ensures the bounded evolution of the controller parameters in steady state. If one substitutes +1 for $R_{Gi}(e)$ term, (17) collapses into the following vector form

$$\underline{w}(t) = -K\underline{b} \int_{t_h}^{t} \mathrm{sgn}(e)dt + \underline{w}(t_h) \tag{25}$$

where $\underline{b}$ is the rx1 column vector each element of which is +1. By combining (25) and (5) and rearranging the result one can come up with (26).

$$\frac{\underline{\varphi}^T(e)\underline{w}(t_h) - \tau}{K\underline{\varphi}^T(e)\underline{b}} = \int_{t_h}^{t} \mathrm{sgn}(e)dt \tag{26}$$

Because denominator of the left-hand side of the above given equation is always greater than zero, the integral term remains bounded, which means that for the modified update rule, the time evolution of the parameter vector in steady sate, which is expressed in (25), remains bounded.

IV. Simulation Results

In the following simulation results for cement mill process are presented. Simulations are performed in MATLAB 5.3 environment and the step size is chosen as 0.001 hours. Initial states of the plant are set to zero and the target set points for the mill load and the product flow rate are chosen as 55 tons and 120 tons/hour respectively. To emphasize the robustness of the control system, the hardness parameter d is changed from 1 to 1.34 at t=10 hours and both state measurements are corrupted with Gaussian noise, having zero mean and variance equal to 10e-5. As can be seen from Fig. 5, Fig. 6 and Fig 7, the state errors converge to zero level in approximately 2.2 hours while tailings flow rate remains bounded. The step change in the hardness parameter does not affect the stability of the system and the free states remains in the close neighborhood of their desired values in steady state despite presence of measurement noise.

As explained before, there are two control loops in the closed loop system, which are highly coupled. This is apparent from the equations describing the mathematical model of the cement mill process. A separate neural network based controller is utilized for the each control loop in order to achieve desired control objectives. One controller produces the feed flow rate input based on the mill load measurements while the other one determines rotational velocity of the classifier utilizing information about the flow rate of the end product. For each controller, the sign functions utilized in the parameter adaptation algorithm is replaced with the approximation function $\mathrm{sgn}(s_c) \approx s_c/(|s_c| + 0.005)$. In this way, any chattering problem that may occur due to discontinuous switching at s_c=0 is eliminated. Furthermore, the uncertainty bounds for the update equations (K_u and K_v) are set to 300 in both controllers. If one investigates (1), it can be seen that the product flow rate increases with decreasing values of the classifier speed which contradicts the assumption given in [6] that if s_p>0, τ>τ_d. In order to get rid of this problem, the output of the controller determining the classifier speed is multiplied with -1. Moreover, to prevent applied control signals taking negative values, the outputs of both controllers are passed through saturation functions. Lastly, for each controller, $R_{Gi}(e)$ term is set to +1 when the absolute value of error is less than 0.2 to eliminate the parameter drift problem discussed.

As depicted in the Fig. 8, the control signals produced by both controllers are satisfactorily smooth and remain within physical limits. This promotes the practical applicability of the proposed scheme. Moreover, the input signals can stabilize the overall system, despite the existence of uncertainties and the strong couplings between the states that occur due to the physical characteristics of the cement mill process. This indicates the robustness of the proposed control algorithm.

Finally, the evolution the controller parameters as functions of time, which, together with the activation functions shown in Fig. 9, determines the control signals, are illustrated in Fig. 10 and Fig. 11 respectively. As these figures suggest, each parameter converges to a steady state value. In other words, the proposed modification on the update equations is successful in preventing the parameter drift problem while allowing the update of the controller parameters at every point in state space.

V. Conclusions

In this paper, neural network based control of a cement mill process is elaborated. A SMC based learning algorithm proposed in the literature, which can robustify learning dynamics, is utilized as the core approach for the construction of the parameter update equations. GRBFNN is chosen as the network architecture and the necessary equations for the parameter adaptation of this structure are derived for first order nonlinear systems. By means of an instability analysis, it is shown that the resulting update rule leads to a drift of the controller parameters in the presence of noise and uncertainties and a modification is proposed in order to alleviate this parametric instability problem. The simulation studies presented indicate that the closed loop system can be stabilized and the states are maintained at the desired levels despite of the measurement noise, the parametric changes and the strong couplings between the system states while allowing the continuous excitation of the update algorithm. This shows that the overall system becomes robust to uncertainties. Another advantage of the control strategy employed is that no explicit knowledge about the plant is required for the design of the controller.

Acknowledgement

This work is supported by TUBITAK (Project no: 100E042).

VI. References

[1] H. Sira-Ramirez and S. H. Zak, "Adaptation of perceptrons with applications to inverse dynamical identification of unknown dynamical systems," IEEE Transactions on System, Man and Cybernetics, vol. 21, pp. 634-643, May/June 1991.

[2] E. Colina-Morles and N. Mort, "Neural network-based adaptive control design," Journal of System Engineering, vol. 1, no. 1, pp. 9-14, 1993.

[3] J. G. Kuschewski, S. Hui and S. H. Zak, "Application of feedforward networks to dynamical system identification and control," IEEE Transactions on Contr. Syst. Technol., vol. 1, pp. 37-49, March 1993.

[4] H. Sira-Ramirez and E. Colina-Morles, "A Sliding Mode Strategy for Adaptive Learning in Adalines," IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications, vol. 42, no. 12, pp. 1001-1012, December 1995.

[5] G. G. Parma, B. R. Menezes and A. P. Braga, "Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks," Electronics Letters, vol. 34, no. 1, pp. 97-98, January 1998.

[6] M. O. Efe, O. Kaynak and X. Yu, "Sliding mode control of a three degrees of freedom anthropoid robot by driving the controller parameters to an equivalent regime," Transactions of the ASME: Journal of Dynamic Systems, Measurement and Control, vol. 122, no. 4, pp. 632-640, December 2000.

[7] L. Magni, G. Bastin and V. Wertz, "Multivariable Nonlinear Predictive Control of Cement Mills," IEEE Transactions on Control Systems Technology, vol.7, no.5, pp. 502-508, July 1999.

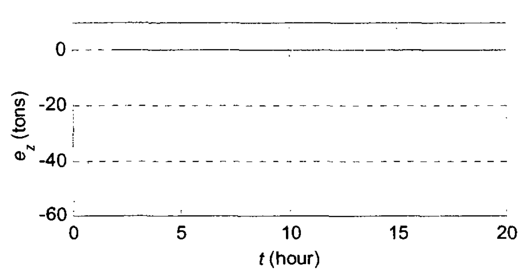

Fig. 5 Time evolution of the mill load error

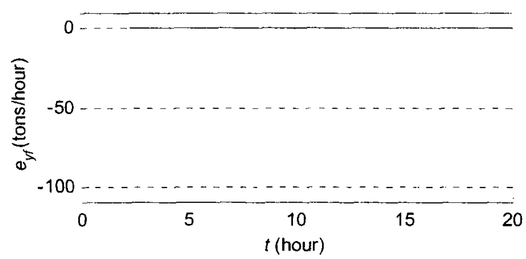

Fig. 6 Time evolution of the product flow rate error

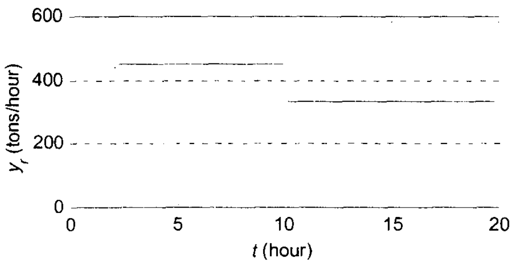

Fig. 7 Time evolution of the tailings flow rate

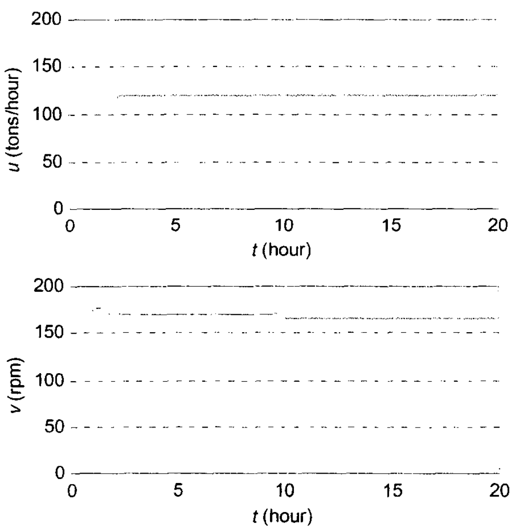

Fig. 8 Applied control signals

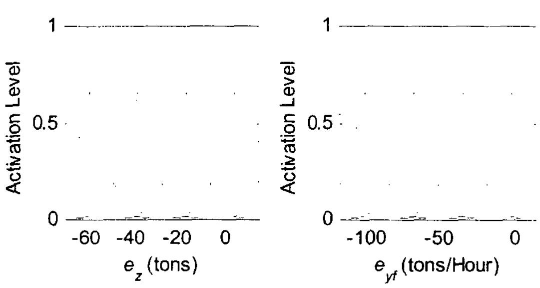

Fig. 9 Activation functions of the GRBFNN based controllers

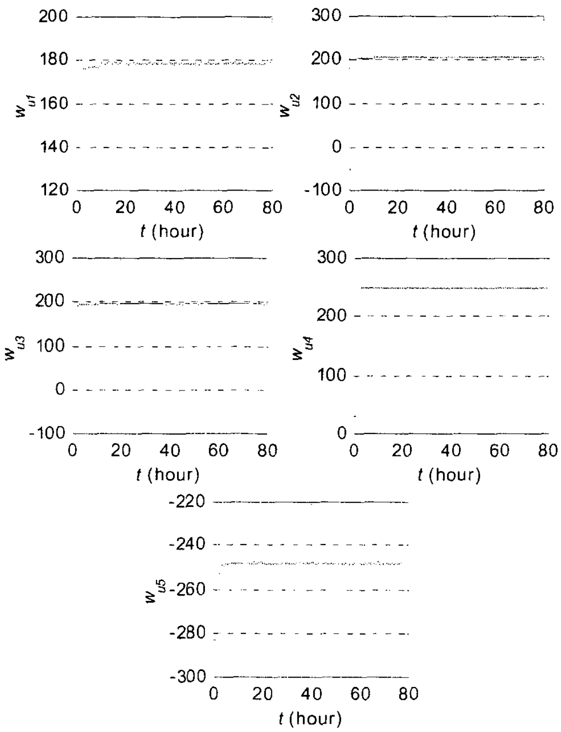

Fig. 10 Time evolutions of parameters for the controller producing feed flow rate

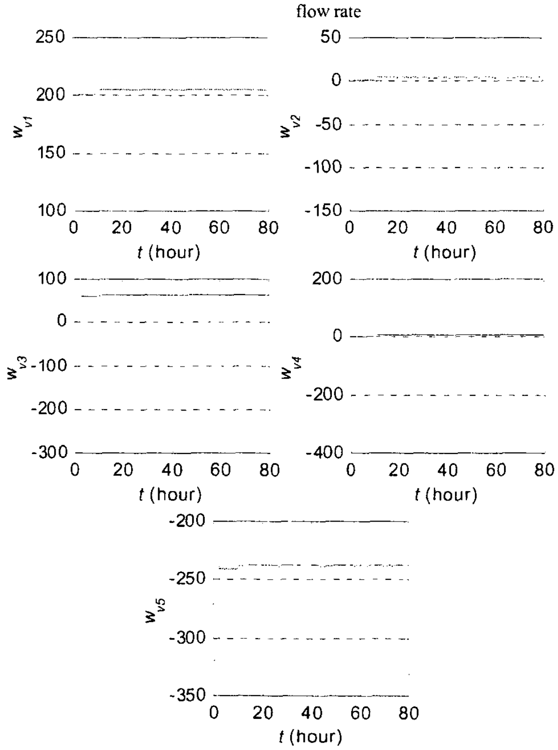

Fig. 11 Time evolutions of parameters for the controller producing classifier speed