Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Process Control 14 (2004) 581–589
JOURNAL OF PROCESS CONTROL
www.elsevier.com/locate/jprocont

Neural network modeling and control of cement mills using a variable structure systems theory based on-line learning mechanism

Andon Venelinov Topalov [1], Okyay Kaynak *

Department of Electrical and Electronic Engineering, Mechatronics Research and Application Center, Bogazici University, Bebek, 34342 Istanbul, Turkey

Received 3 March 2003; received in revised form 1 October 2003; accepted 14 October 2003

Abstract

It is well known that the major cause of instability in industrial cement ball mills is the so-called plugging phenomenon. A novel neural network adaptive control scheme for cement milling circuits that is able to fully prevent the mill from plugging is presented. Estimates of the one-step-ahead errors in control signals are calculated through a neural predictive model and used for controller tuning. A robust on-line learning algorithm, based on sliding mode control (SMC) theory is applied to both: to the controller and to the model as well. The proposed approach allows handling of mismatches, uncertainties and parameter changes in the model of the mill. The simulation results from indicate that both the neural model and the controller inherit the major advantages of SMC, i.e. robustness. Furthermore, learning is achieved in a rapid manner.

Keywords: Intelligent control; Neurocontrollers; Variable structure systems; Adaptive control

1. Introduction and process description

In cement industry, the control of the milling process has remained a challenging problem for years because of the existing model uncertainties, nonlinearities, changes in the parameters and their interdependence.

Fig. 1 illustrates a cement ball mill in a closed loop with a separator. Inside of the rotating mill the feed flow, consisting of clinker, slag, gypsum and other raw material components is ground by steel balls. An elevator is used to transport the mill product into the separator where, depending on adjustable air flow rate and speed, it is divided into a flow of rejected oversized particles (tailings), which are returned to the mill inlet, and a flow of fine particles, which forms the final product.

Cement milling is a highly energy-consuming operation in the cement industry, so efficient control is required in order to reduce the specific production costs (kW h/ton of cement produced) while maintaining the product quality, i.e., cement fineness at an acceptable level.

There exists a nonlinear dependence of the mill product on the mill load (amount of material inside the mill) and the hardness of grinded material which can sometimes cause the instability of the system, obstruction of the mill and interruption of the grinding process. The phenomenon is known as ''plugging''. The specific production costs also depend on the load in the mill.

In current practice, the grinding processes are still often operated in manual mode or, to a limited extend, using simple monovariable controllers [1]. During the last 10 years, several control approaches have been proposed including linear multivariable control techniques [2–4]. Recently the research efforts have been concentrated on controllers that are able to prevent the mill from plugging. The linear controllers based on a linear approximation of the grinding process, are stable and effective only around the specified nominal operating point. The existing disturbances (like changes in the grindability of the raw material) may drive the mill to a new operating point where the controller cannot stabilize the plant.

* Corresponding author. Tel.: +90-212-287-2475; fax: +90-212-287-2465.

E-mail addresses: topalov@boun.edu.tr (A.V. Topalov), kaynak@boun.edu.tr (O. Kaynak).

[1] On leave from the Department of Control Systems, Technical University of Sofia—Plovdiv branch, 61, Sankt Petersburg Blvd., 4000 Plovdiv, Bulgaria.

0959-1524/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2003.10.005

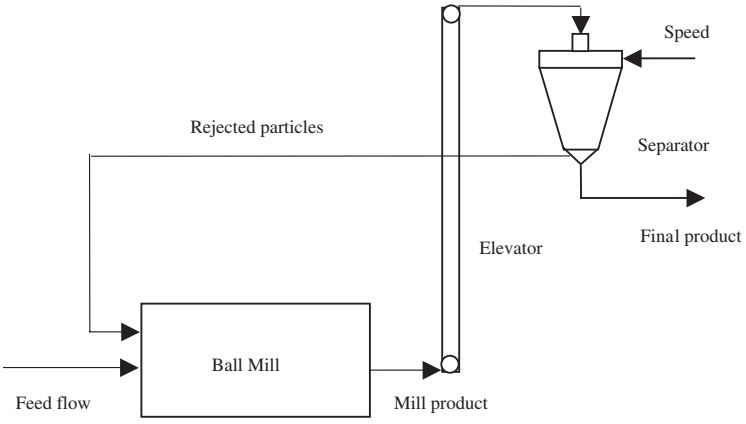

Fig. 1. Milling circuit.

In an attempt to solve the problem, a simplified nonlinear model of the milling circuit, that is able to reproduce the plugging phenomenon in a realistic way, was developed in a recent study [5] and a state feedback controller based a nonlinear predictive control strategy was designed. Although the proposed system has larger stability region compared to the linear controllers, the risk of plugging was not completely avoided. In a later work, Grognard et al. [6], using the model of the grinding circuit proposed in [5], suggested a robust state feedback nonlinear controller that is able to prevent the mill from plugging.

Control techniques using classical controllers proposed in [1–6] can be efficient only if the interrelationships between the variables can be correctly determined and modeled. It is well known that material grinding depends on many factors including mill geometry, speed, ball size distribution, material grindability and granulometry. Due to the inherent process complexity, the development of an accurate model of the cement milling circuit is not a simple task and, as stated in [4], the dynamic modeling of grinding processes is still an open area.

Taking into account the above characteristics of the cement milling circuit, a neuro-adaptive control is proposed as an appropriate control method in this work. The major advantage of the proposed method is that there is no need to develop an accurate model of the milling circuit in advance. It is learned instead on-line, by a neural network. This allows dealing more successfully with the existing model uncertainties, parameter changes and interdependences and in particular can completely prevent mill from plugging.

2. The neuro-adaptive control scheme

Based on the principles of the one-step-ahead control, a new neurocontrol scheme is constructed as depicted in Fig. 2. The control objective is to regulate the final product rate y_f and the mill load z at the desired set points y_f^{ref} and z^{ref} by acting on the feed flow rate u and the separator speed v. The controller must prevent the mill from plugging and be robust against modeling uncertainties. It is to be noted that the errors used for the adaptation of the controller should be the command-errors (i.e., the plant input errors) [7,8]. However, this is normally not available and what is generally done is the use of the difference between the desired response and the actual response (i.e., the plant performance errors, or the tracking errors). In the approach adopted in this paper, an estimate of the command-errors is obtained, named "virtual" command-errors, by using a forward dynamics plant predictive model. A neural network model of the cement milling circuit is used for one-step-ahead prediction of the product flow rate and the load in the mill. The controller monitors the current outputs of the plant y_f and z, and the corresponding reference signals, and based on errors e_{y_f} and e_z, and the implemented adaptive control law, manipulates the control inputs to make the predicted outputs, based on the internal model of the system discovered and stored in the neural-net model (predictive model) close to the desired outputs.

The model errors $e_1(k) = y_f(k) - \hat{y}_f(k)$ and $e_2(k) = z(k) - \hat{z}(k)$ are used for the training of the plant model, where $y_f(k)$ and $\hat{y}_f(k)$ are the real and the estimated value of the product flow rate (tons/h), while $z(k)$ and $\hat{z}(k)$ are the real and the estimated value of the load in the mill (tons), respectively.

The cost function to be minimized, during the controller adaptation process, is the one-step-ahead predicted sum-squared error of the closed-loop system.

$$J = \tfrac{1}{2}\left[e_3^2(k+1) + e_4^2(k+1)\right] \tag{1}$$

In above $e_3(k+1) = y_f^{ref}(k+1) - y_f^{pr}(k+1)$ and $e_4(k+1) = z^{ref}(k+1) - z^{pr}(k+1)$ are the predicted errors, and $y_f^{ref}(k+1)$, and $y_f^{pr}(k+1)$ are the one-step-ahead reference and the predicted value of the product flow rate,

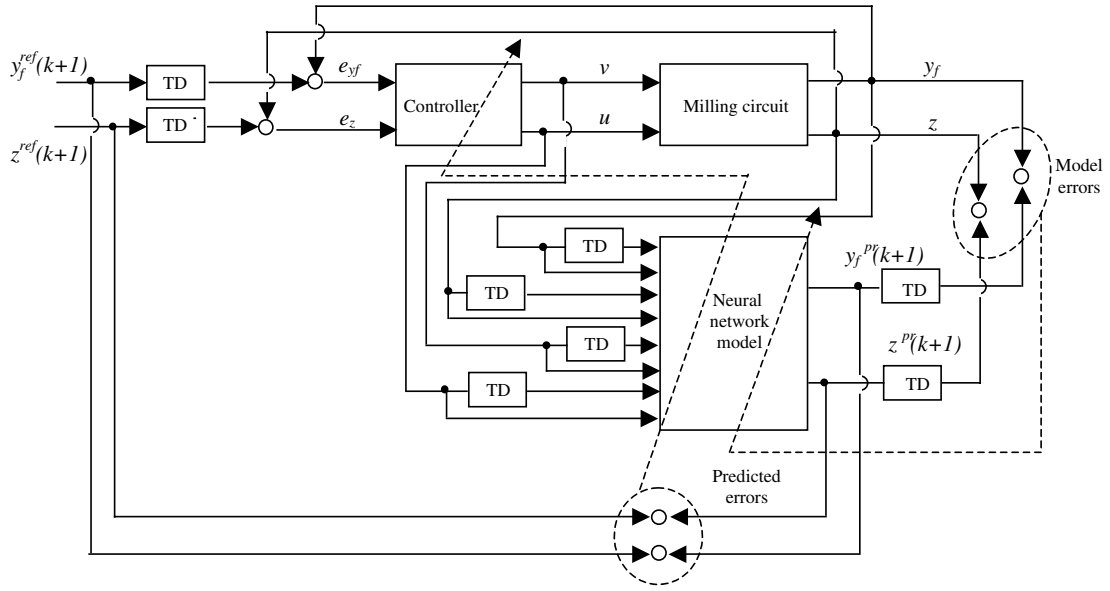

Fig. 2. The neuro-adaptive control scheme of the milling circuit.

while $z^{\mathrm{ref}}(k+1)$, and $z^{\mathrm{pr}}(k+1)$ are the one-step-ahead reference and the predicted value of the load in the mill, respectively.

The predicted command errors (i.e., the plant input errors), named as the "virtual" errors of the control inputs u (feed flow rate (tons/h)) and v (the separator speed (r/min)) in this work, can be determined as follows:

$$\begin{aligned} e_u(k+1) &= \frac{\partial J}{\partial u} \\ &= -\frac{\partial y_{\mathrm{f}}^{\mathrm{pr}}(k+1)}{\partial u(k)} e_3(k+1) \\ &\quad -\frac{\partial z^{\mathrm{pr}}(k+1)}{\partial u(k)} e_4(k+1) \end{aligned} \tag{2}$$

$$\begin{aligned} e_v(k+1) &= \frac{\partial J}{\partial v} \\ &= -\frac{\partial y_{\mathrm{f}}^{\mathrm{pr}}(k+1)}{\partial v(k)} e_3(k+1) \\ &\quad -\frac{\partial z^{\mathrm{pr}}(k+1)}{\partial v(k)} e_4(k+1) \end{aligned} \tag{3}$$

This approach is similar in spirit to the one adopted in [7] under the name "distal supervised learning" and in [9]. The novel aspect of the approach presented in this paper is that the one-step-ahead predicted virtual errors $e_u(k+1)$ and $e_v(k+1)$ are used instead of the present virtual errors that may be noted as $e_u(k)$ and $e_v(k)$.

3. The plant predictive model

A feedforward neural network is used to obtain the plant model and to predict outputs of the plant one-step-ahead. The network has a layered structure that consists of eight input units, two output units and a hidden layer with 32 units. The input structure of the neural network plant model was adopted in accordance with the NNARX model structure in predictor form [9] by extending it for the MIMO case. A neural network with one hidden layer has been adopted here since, as it has been shown by, e.g. Hornik et al. [10], using the Stone–Weierstrass theorem, all continuous functions can be approximated to any desired accuracy, in terms of the uniform norm, with a network of one hidden layer of sigmoidal (or hyperbolic tangent) hidden units and a layer of linear output units. The problem related to how many neurons to include in the hidden layer is addressed by Barron [11] and a significant result is derived about the approximation capabilities of two-layer perceptron networks when the function to be approximated exhibits a certain smoothness. Unfortunately, the result is difficult to apply in practice for selecting the number of hidden units and in this work the number of hidden neurons was established experimentally by using trials and errors.

The outputs of the neural network plant model can be determined as follows:

$$\begin{aligned} y_{\mathrm{f}}^{\mathrm{pr}}(k+1) &= \sum_{i=1}^{q} W2_i^{y_{\mathrm{f}}}(k) f_{\mathrm{H}}\left[\sum_{j=1}^{n} W1_{ij}(k) I_j(k)\right] \\ &= \sum_{i=1}^{q} W2_i^{y_{\mathrm{f}}}(k) f_{\mathrm{H}}(x_i) = W2^{y_{\mathrm{f}}}(k) Y_{\mathrm{H}}(k) \end{aligned} \tag{4}$$

$$\begin{aligned} z^{\mathrm{pr}}(k+1) &= \sum_{i=1}^{q} W2_i^{z}(k) f_{\mathrm{H}}\left[\sum_{j=1}^{n} W1_{ij}(k) I_j(k)\right] \\ &= \sum_{i=1}^{q} W2_i^{z}(k) f_{\mathrm{H}}(x_i) = W2^{z}(k) Y_{\mathrm{H}}(k) \end{aligned} \tag{5}$$

In above q is the number of neurons in the hidden layer (32 in this work), n is the number of network inputs (8 in this work), $W1_{ij}$ is the weight of the ith neuron in the first (hidden) layer from its jth input, $W2_i^{y_f}$ is the weight of the output neuron for y_f^{pr} from its ith input (from ith neuron in the hidden layer), $W2_i^z$ is the weight of the output neuron for z^{pr} from its ith input, $I(k)$ is a set of input data presented to the network at the current time step k and consisting of $u(k)$, $u(k-1)$, $v(k)$, $v(k-1)$, $y_f(k)$, $y_f(k-1)$, $z(k)$ and $z(k-1)$. $f_H(\cdot)$ is the activation function for the hidden node i and $x_i = \sum_{j=1}^{n} W1_{ij}(k)I_j(k)$.

In order to calculate the "virtual" errors in the control inputs v and u, the partial derivatives in Eqs. (2) and (3) must be determined first. The partial derivative of y_f^{pr} with respect to u can be calculated as in Eq. (6):

$$\frac{\partial y_f^{pr}(k+1)}{\partial u(k)} = \sum_{i=1}^{q} \frac{\partial \hat{y}_f(k+1)}{\partial f_H(x_i)} \cdot \frac{\partial f_H(x_i)}{\partial x_i} \cdot \frac{\partial x_i}{\partial u(k)} = \sum_{i=1}^{q} W2_i^{y_f}(k)[1 - f_H(x_i)]f_H(x_i)W1_{i1}(k) \quad (6)$$

where $f_H(x_i) = \frac{1}{1+e^{-x_i}}$ is a log-sigmoid function.

Similarly it can be derived that:

$$\frac{\partial y_f^{pr}(k+1)}{\partial v(k)} = \sum_{i=1}^{q} W2_i^{y_f}(k)[1 - f_H(x_i)]f_H(x_i)W1_{i3}(k) \quad (7)$$

$$\frac{\partial z^{pr}(k+1)}{\partial u(k)} = \sum_{i=1}^{q} W2_i^{z}(k)[1 - f_H(x_i)]f_H(x_i)W1_{i1}(k) \quad (8)$$

$$\frac{\partial z^{pr}(k+1)}{\partial v(k)} = \sum_{i=1}^{q} W2_i^{z}(k)[1 - f_H(x_i)]f_H(x_i)W1_{i3}(k) \quad (9)$$

If a tan-sigmoid activation function $f_H(x_i) = \frac{1-e^{-x_i}}{1+e^{-x}}$ is applied, the Eq. (6) needs to be modified as follows:

$$\frac{\partial y_f^{pr}(k+1)}{\partial u(k)} = \sum_{i=1}^{q} W2_i^{y_f}(k)[1 - f_H^2(x_i)](1/2)W1_{i1}(k) \quad (10)$$

Similar modifications should be done to Eqs. (7)–(9).

4. The controller

Adaptive PID-like controller structures are considered in this work for the two simultaneously controlled parameters (the finished product and the load in the mill). It should be noticed that the range of possible controller implementations can be much wider and structures having one or more hidden layers can be used too. The required controls $v(k)$ and $u(k)$ at time instant k are computed as follows:

$$v(k) = v(k-1) + K_{v1}(k)h_{v1}(k) + K_{v2}(k)h_{v2}(k) + K_{v3}(k)h_{v3}(k) \quad (11)$$

$$u(k) = u(k-1) + K_{u1}(k)h_{u1}(k) + K_{u2}(k)h_{u2}(k) + K_{u3}(k)h_{u3}(k) \quad (12)$$

where $K_{lm}(k)$ $(l = v, u;\ m = 1, 2, 3)$ are the controller parameters at the time instant k and

$$e_{y_f}(k) = y_f^{ref}(k) - y_f(k)$$
$$e_z = z^{ref}(k) - z(k)$$
$$h_{v1}(k) = e_{y_f}(k) - e_{y_f}(k-1)$$
$$h_{v2}(k) = e_{y_f}(k)$$
$$h_{v3}(k) = e_{y_f}(k) - 2e_{y_f}(k-1) + e_{y_f}(k-2)$$
$$h_{u1}(k) = e_z(k) - e_z(k-1)$$
$$h_{u2}(k) = e_z(k)$$
$$h_{u3}(k) = e_z(k) - 2e_z(k-1) + e_z(k-2)$$

5. The sliding mode learning algorithm

If the tuning of the parameters $K_{lm}(k)$ $(l = v, u;\ m = 1, 2, 3)$ is never turned off, the controllers (11) and (12) should in principle work as adaptive controllers for time-varying systems as well. Naturally the "forward" neural model providing the virtual errors e_v and e_u must be updated on-line, simultaneously with the update of the controller parameters. When the above-specialized training principle was originally introduced, a recursive gradient method was considered. However, it is well known that the convergence speed of gradient based learning (e.g., dynamic back propagation method) is very slow, especially when the search space is complex. Additionally, the tuning process can easily be trapped into a local minimum. Furthermore, a number of numerical robustness issues need to be taken also into account when recursive estimation algorithms are applied over long periods of time [12].

In the theory of control engineering, one way of designing a robust and stable control system is to use the variable structure systems (VSS) approach, which enables the designer to come up with rigorous stability analysis. It is a well-known fact that a variable structure controller with a switching output will (under certain circumstances) result in a sliding mode on a predefined subspace of the state space. This mode has useful invariance properties in the face of uncertainties in the plant model and therefore is a good candidate for control of uncertain nonlinear systems.

The studies demonstrating the high performance of the variable structure control in handling the uncertainties and imprecision have motivated the use of sliding mode control (SMC) scheme in training of computationally intelligent systems. The results presented in [13,14] have shown that the convergence properties of the gradient-based training strategies can

be improved by utilizing the SMC scheme. The method adopted in these works can be considered as an indirect use of VSS theory. Some studies on the direct use of SMC strategy are also reported in the literature [15,16]. In [15] the zero level set of the learning error variable in Adaline neural networks is regarded as a sliding surface in the space of learning parameters. A sliding mode trajectory can then be induced, in finite time, on such a desired sliding manifold. Sliding mode invariance conditions determine a least-squares characterization of the adaptive weights average dynamics whose stability features can be studied using standard time-varying linear systems results. Robustness of the learning algorithm, with respect to bounded external perturbation signals, and measurement noises, is also demonstrated. Yu et al. [16] extend further the results of [15] by introducing adaptive uncertainty bound dynamics of the signals.

Recently, a new learning algorithm for training multilayer artificial neural networks, based on direct use of SMC strategy, was proposed by Parma et al. [17,18]. In addition to its applicability for updating the weights in the hidden layers of multilayer network structures it also differs from the algorithms in [15,16] due to the definition of the sliding surface, which is now determined by taking not only the learning error variable, but also its time derivative. The on-line learning capability of the algorithm in applications demanding adaptation to constantly changing environmental parameters, such as adaptive real-time control, was first investigated in [19]. It is currently implemented as an on-line learning mechanism for the proposed neurocontrol system of the cement milling circuit. Some improvements are suggested in order to achieve learning of the controller under an optimal control criterion.

The SMC design is divided into two phases. In the first, sliding surfaces to produce the input/output behavior are defined. In the second, the parameters/weights are updated in order to satisfy the conditions for tracking and sliding on the surfaces.

Consider the virtual errors e_v and e_u, by adding to these terms the terms ρv and ρu, ($\rho \geqslant 0$), respectively, for penalizing the controls v and u a simple type of optimal control criterion is achieved.

$$\varepsilon_v = e_v + \rho v; \quad \varepsilon_u = e_u + \rho u \tag{13}$$

As ρ is increased the control signal will become smoother and will attain smaller values. The sliding surfaces for the controller could then be defined as follows:

$$\begin{aligned} S_v(k) &= \Delta\varepsilon_v(k) + \lambda_1\varepsilon_v(k) \\ S_u(k) &= \Delta\varepsilon_u(k) + \lambda_2\varepsilon_u(k) \end{aligned} \tag{14}$$

where λ_1, $\lambda_2 > 0$ and

$$\begin{aligned} \Delta\varepsilon_v(k) &= \varepsilon_v(k) - \varepsilon_v(k-1) \\ \Delta\varepsilon_u(k) &= \varepsilon_u(k) - \varepsilon_u(k-1) \end{aligned} \tag{15}$$

For the output layer of the predictive model the sliding surfaces are defined as in Eq. (16) ($\lambda_3, \lambda_4 > 0$):

$$\begin{aligned} S_{y_{\mathrm{T}}}(k-1) &= \Delta e_1(k) + \lambda_3 e_1(k) \\ S_z(k-1) &= \Delta e_2(k) + \lambda_4 e_2(k) \end{aligned} \tag{16}$$

where

$$\begin{aligned} \Delta e_1(k) &= e_1(k) - e_1(k-1) \\ \Delta e_2(k) &= e_2(k) - e_2(k-1) \end{aligned} \tag{17}$$

The following sliding surface is defined for the hidden layer [18]:

$$S_{i\mathrm{H}}(k-1) = \Delta e_{i\mathrm{H}}(k) + \lambda_{\mathrm{H}} e_{i\mathrm{H}}(k) \tag{18}$$

where $\lambda_{\mathrm{H}} > 0$ and

$$\Delta e_{i\mathrm{H}}(k) = e_{i\mathrm{H}}(k) - e_{i\mathrm{H}}(k-1) \tag{19}$$

$$\begin{aligned} &e_{i\mathrm{H}}(k) = f'_{\mathrm{H}}(x_i)[e_1(k)W2_i^{y_{\mathrm{T}}}(k) + e_2(k)W2_i^z(k)] \quad \text{and} \\ &x_i = \sum_{j=1}^{n} W1_{ij}(k)I_j(k) \end{aligned} \tag{20}$$

In case $f_{\mathrm{H}}(x_i) = \frac{1}{1+e^{-x_i}}$ is a log-sigmoid function

$$e_{i\mathrm{H}}(k) = [1 - f(x_i)]f(x_i)[e_1(k)W2_i^{y_{\mathrm{T}}}(k) + e_2(k)W2_i^z(k)] \tag{21}$$

If a tan-sigmoid activation function $f_{\mathrm{H}}(x_i) = \frac{1-\mathrm{e}^{-x_i}}{1+\mathrm{e}^{-x}}$ is applied, $e_{i\mathrm{H}}(k)$ is defined as follows:

$$e_{i\mathrm{H}}(k) = \tfrac{1}{2}[1 - f_{\mathrm{H}}^2(x_i)][e_1(k)W2_i^{y_{\mathrm{T}}}(k) + e_2(k)W2_i^z(k)] \tag{22}$$

Therefore, considering the fact that the output units of the both learning structures have linear activation functions, the weight update rules for the output layer of the plant model and for the controller can be obtained, by following the method presented in [18], as in Eqs. (23)–(26) (for $\alpha_i > 0$, $i = 1, \ldots, 4$):

$$\Delta W2_i^{y_{\mathrm{T}}}(k) = \alpha_1|e_1(k+1)|f_{\mathrm{H}}(x_i)\mathrm{sign}\,(S_{y_{\mathrm{T}}}(k)) \tag{23}$$

$$\Delta W2_i^z(k) = \alpha_2|e_2(k+1)|f_{\mathrm{H}}(x_i)\mathrm{sign}\,(S_z(k)) \tag{24}$$

$$\Delta K_{vm}(k) = \alpha_3|\varepsilon_v(k)|h_{vm}(k)\mathrm{sign}\,(S_v(k)) \tag{25}$$

$$\Delta K_{um}(k) = \alpha_4|\varepsilon_u(k)|h_{um}(k)\mathrm{sign}\,(S_u(k)) \tag{26}$$

The weight update rule for the hidden layer of the plant model is obtained as in Eq. (27) (for $\beta > 0$):

$$\Delta W1_{ij}(k) = \beta|e_{i\mathrm{H}}(k+1)|I_j(k)\mathrm{sign}\,(S_{i\mathrm{H}}(k)) \tag{27}$$

The use of SMC, when compared to standard back-propagation, provides a faster and more efficient way for updating the weights.

It can be easily seen that higher values of λ_i ($i = 1, \ldots, 4$) and λ_{H} will lead to a faster convergence, but there is in fact a trade-off between the values of λ_i and λ_{H}, and the relevant values of α_i ($i = 1, \ldots 4$) and β, which by their turn, depend also on the amplitude variations of the system parameters. Therefore, for every system to be controlled, the bounds for α_i and β (and λ_i

and λ_H) should be derived in advance in order to predict convergence and stability properties.

The upper limits of α_i and β can be obtained from the sliding surface expression. For a given S, delimited by the training data and network topology, the upper limits for the gains α_i and β can be easily obtained. According to Utkin [20], the condition for existence of sliding mode and system stability is defined by Eq. (28).

$$S\frac{dS}{dt} < 0 \tag{28}$$

For discrete time, Sarpturk et al. [21] defined the equation $|S(k+1)| < |S(k)|$, instead of Eq. (28), as a condition to guarantee the sliding manifold. It will be true also in case the above condition is written as in Eq. (29), because the convergence of the sequence of $S(k)$ is quaranteed if the sequence $|S(k)|$ is a decreasing sequence.

$$|S(k)| < |S(k-1)| \tag{29}$$

Since the adaptation of the weights of the learning structures, which could be interpreted as control signals with regard to these structures, is a function of α_i $(i = 1,\ldots,4)$ and β, the analysis of Eq. (29) results in their limit values. Convergence is guaranteed for any values of α_i $(i = 1,\ldots,4)$ and β, within the limits established by S, since it implies on the existence of sliding mode.

For the output layer of the predictive model, taking into account that

$$\begin{aligned}\Delta e_1(k+1) &= (y_f(k+1) - y_f^{pr}(k+1)) - (y_f(k) - y_f^{pr}(k)) \\ &= \Delta y_f(k+1) - \Delta y_f^{pr}(k+1)\end{aligned} \tag{30}$$

and

$$\Delta e_2(k+1) = \Delta z(k+1) - \Delta z^{pr}(k+1) \tag{31}$$

the substitution of Eqs. (30) and (31) into Eq. (16), results in Eqs. (32) and (33)

$$\begin{aligned}S_{y_f}(k) &= \Delta y_f(k+1) - \Delta y_f^{pr}(k+1) + \lambda_3(y_f(k+1) \\ &\quad - y_f^{pr}(k+1))\end{aligned} \tag{32}$$

$$\begin{aligned}S_z(k) &= \Delta z(k+1) - \Delta z^{pr}(k+1) + \lambda_4(z(k+1) \\ &\quad - z^{pr}(k+1))\end{aligned} \tag{33}$$

Considering Eq. (23) and (24) and the network definitions described previously (Eq. (4) and (5)), Eqs. (34) and (35) could be now obtained

$$\begin{aligned}\Delta y_f^{pr}(k+1) &= \Delta W 2^{y_f}(k) Y_H(k) \\ &= |e_1(k+1)| Y_H^2(k) \alpha_1 \operatorname{sign}(S_{y_f}(k))\end{aligned} \tag{34}$$

$$\Delta z^{pr}(k+1) = |e_2(k+1)| Y_H^2(k) \alpha_2 \operatorname{sign}(S_z(k)) \tag{35}$$

Substituting Eqs. (34) and (35) into Eqs. (32) and (33), leads to Eqs. (36) and (37) respectively

$$S_{y_f}(k) = [-B_{y_f}(k)\alpha_1 + |A_{y_f}(k)|]\operatorname{sign}(S_{y_f}(k)) \tag{36}$$

$$S_z(k) = [-B_z(k)\alpha_2 + |A_z(k)|]\operatorname{sign}(S_z(k)) \tag{37}$$

where $B(k)$ and $A(k)$ are defined as:

$$\begin{aligned}B_{y_f}(k) &= |e_1(k+1)| Y_H^2(k) \\ B_z(k) &= |e_2(k+1)| Y_H^2(k)\end{aligned} \tag{38}$$

$$\begin{aligned}A_{y_f}(k) &= \lambda_3 e_1(k+1) + \Delta y_f(k+1) \\ A_z(k) &= \lambda_4 e_2(k+1) + \Delta z(k+1)\end{aligned} \tag{39}$$

Considering Eqs. (29) and (36), (37) the upper limits for α_1 and α_2 could be now obtained:

$$\alpha_1 < \left|\frac{A_{y_f}(k)}{B_{y_f}(k)}\right|, \quad \alpha_1 < \frac{|A_{y_f}(k-1)| - |A_{y_f}(k)|}{B_{y_f}(k-1) - B_{y_f}(k)} \tag{40}$$

$$\alpha_2 < \left|\frac{A_z(k)}{B_z(k)}\right|, \quad \alpha_2 < \frac{|A_z(k-1)| - |A_z(k)|}{B_z(k-1) - B_z(k)} \tag{41}$$

Finally the limits for α_1 and α_2 are:

$$0 < \alpha_1 < \min\left(\left|\frac{A_{y_f}(k)}{B_{y_f}(k)}\right|, \frac{|A_{y_f}(k-1)| - |A_{y_f}(k)|}{B_{y_f}(k-1) - B_{y_f}(k)}\right) \tag{42}$$

$$0 < \alpha_2 < \min\left(\left|\frac{A_z(k)}{B_z(k)}\right|, \frac{|A_z(k-1)| - |A_z(k)|}{B_z(k-1) - B_z(k)}\right) \tag{43}$$

In a similar way the limits for the coefficients α_3, α_4 and β can be obtained.

With the surfaces defined as in Eqs. 14, 16, 18, the weight updating rules defined as in Eqs. (23)–(27), α_i $(i = 1,\ldots,4)$ and β satisfying the boundary conditions, it can be affirmed that the model errors e_1, e_2 and the "virtual" command-errors e_v, e_u tend to zero [18].

6. Simulation results

Extensive simulations are conducted to investigate the ability of the scheme depicted on Fig. 2 to work as an adaptive controller when the sliding mode learning algorithm is applied. The nonlinear model of the milling circuit introduced in [5] is used in simulations. This model describes the evolution of the load in the mill and it is able to reproduce some (unstable) behavior that has been observed with the LQ controller in the feedback loop. It consists of three nonlinear differential equations explaining the evolution of three states

$$\begin{cases} T_f \dot{y}_f = -y_f + (1 - \alpha(z,v,d))\varphi(z,d) \\ \dot{z} = -\varphi(z,d) + u + y_r \\ T_r \dot{y}_r = -y_r + \alpha(z,v,d)\varphi(z,d) \end{cases} \tag{44}$$

with $\alpha(z,v,d) = \frac{\varphi^m v^n}{L_\alpha + \varphi^m v^n}$; $L_\alpha = 570^m 170^n (\frac{570}{450} - 1)$ (tons/h)^m (r/min)^n; $m = 0.8$; $n = 4$; $\varphi(z,d) = \max\{0; (-dL_{\varphi 1} z^2 + L_{\varphi 2} z)\}$; $L_{\varphi 2} = 16.50$ (h)$^{-1}$; $L_{\varphi 1} = 0.1116$ (tons h)$^{-1}$; $T_f = 0.3$ h; $T_r = 0.01$ h; $d = 1$.

In the above y_r is the tailings flow rate (tons/h), $\varphi(z,d)$ is the output flow rate of the mill (tons/h), and d represents the hardness of the material inside the mill

with respect to the nominal one. The values of the various coefficients in this nonlinear model have been tuned in such a way that the model step responses fit with experimental step responses described in [5].

From the second nonlinear differential equation and the definition of φ, it can be noticed that there is a constraint on the maximum value of the circulating load (feed plus tailings flow rates). By an easy manipulation of the three nonlinear differential equations it follows that it is possible to choose independently only two of the three steady-state values, the third one being imposed from the model equations. Choosing set-points for y_f and y_r would impose a given steady-state value for φ which could be unachievable. The above explains the reasons for using the load in the mill z and the value of the product flow rate y_f as set-points in the control strategy suggested in [5].

In the simulations a sampling period $T_e = 0.3$ min is chosen. The results are presented in Fig. 3(a)–(e). Simulations start with randomly chosen weights of the neural model and the controller and with set-points values $y_f^{ref} = 120$ tons/h and $z = 55$ tons. Then it is supposed that the hardness d changes from its nominal value 1 to 1.5 within the time interval 211–420 min, the set-point for the product flow rate y_f^{ref} changes from 120 to 125 tons/h within the time interval 120–375 min, and the set-point for the load in the mill z changes from 55 to 70 tons during the period from 67 to 270 min. The

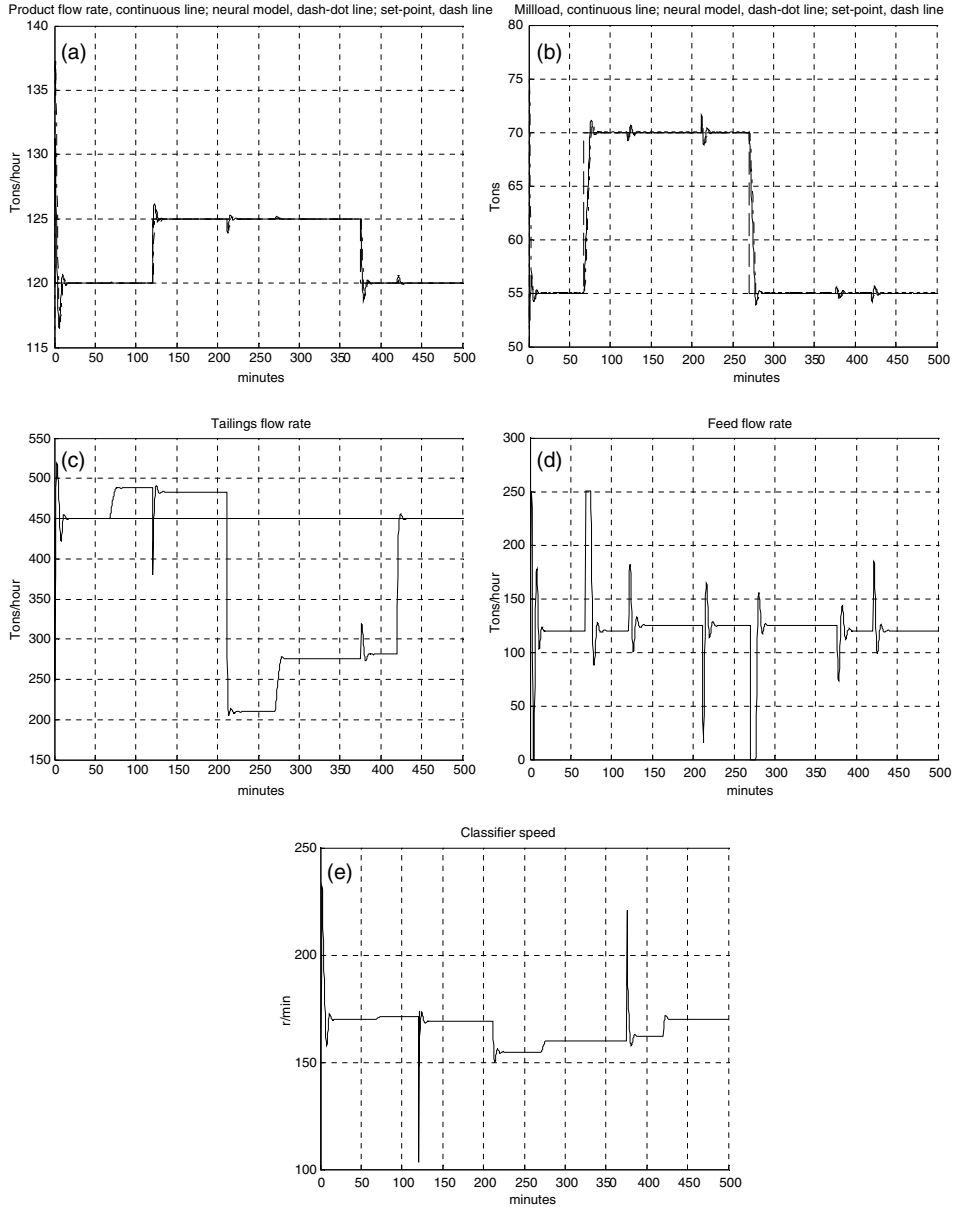

Fig. 3. Simulation results.

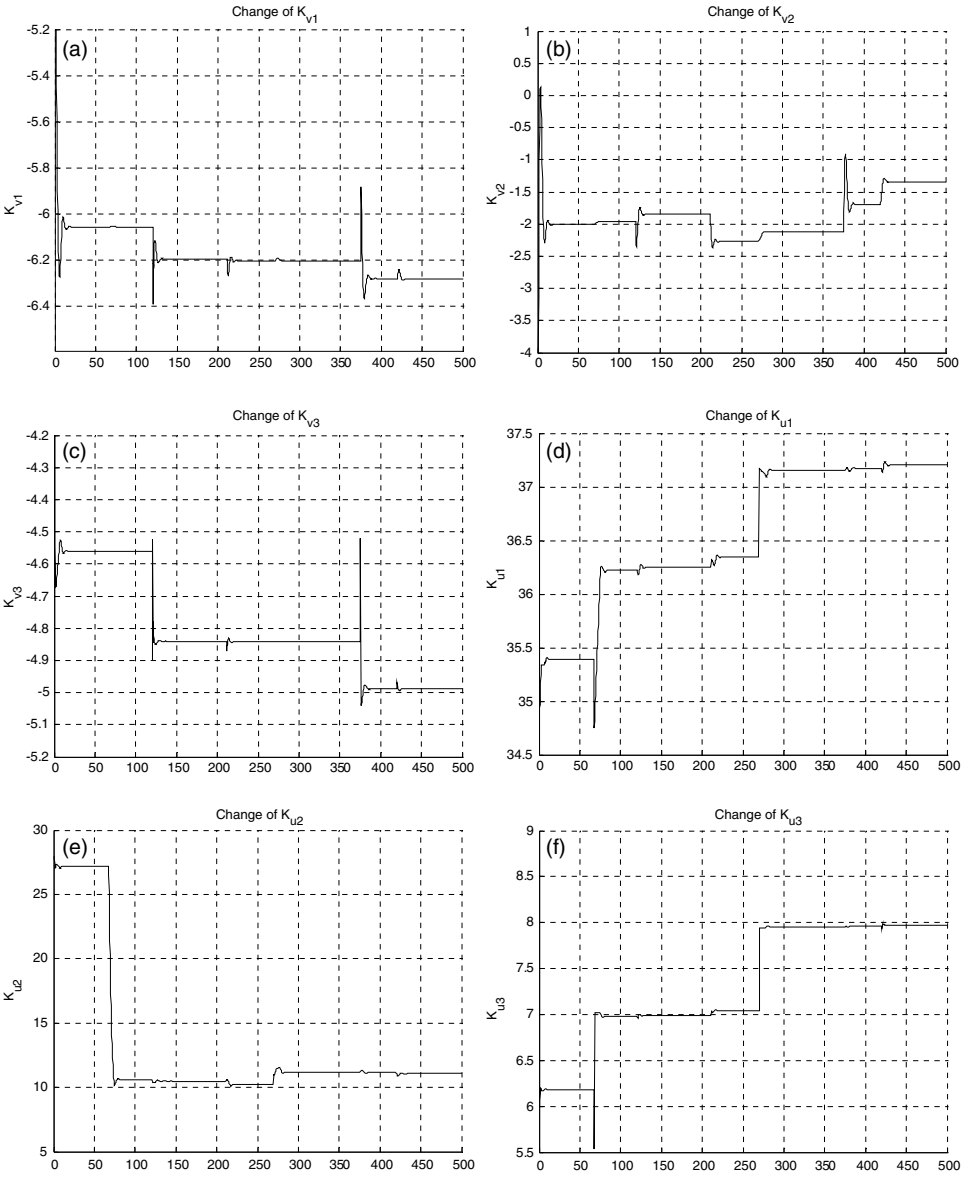

Fig. 4. Changes of the parameters of the adaptive controller during the time of simulation.

accepted values of the different coefficients and constants during simulations are: $\lambda_i = 1 \ \ (i = 1, \ldots 4, H)$; $\alpha_1, \alpha_2, \ \ \beta = 5 \times 10^{-4}$; $\ \alpha_3, \alpha_4 = 8 \times 10^{-3}$; $\ \rho = 5 \times 10^{-2}$; $\delta = 1 \times 10^{-2}$; $\eta = 1 \times 10^{-8}$. The control signals u and v are restricted within the interval [0 250]. The values of the signals presented to the inputs of the neural network model are decreased by factor of 10^{-4}. It is clear that the nominal responses to the set-points changes as well as to the hardness change are correct and these changes do not cause instabilities of the closed loop (plugging). It should be noted also that the transition times are much smaller as compared to those obtained when a LQ controller, or a NRH control law is implemented [5].

Fig. 4(a)–(f) shows the changes of the controller parameters during the time of simulation.

7. Conclusions

A novel neural network adaptive control scheme for cement milling circuits is proposed. Its major advantage is that there is no need to develop an accurate model of the milling circuit in advance. Instead, it is learned on-line by a neural network. This allows dealing more successfully with the existing model uncertainties, parameter changes and their interrelations. The possibilities for implementation of a sliding mode learning algorithm, proposed in [17,18], as an on-line mechanism for adaptation of neurocontrol systems have been investigated. It is confirmed that the algorithm can be used to train the networks as they interact with the external environment. The trained structures are robust and

learn fast, both of which are features inherited from SMC. Simulation results show that the nominal responses to the set-point changes as well as to the hardness change are correct and these changes do not cause instabilities of the closed loop system (plugging). In can be observed also that the transition times, undershoots and overshoots are much smaller compared to these reported in [5].

The real difficulty related with the practical implementation of the proposed control approach (and also with the previously proposed approaches in [5,6]) is the need for a measurement of the load of the mill. As it has been already mentioned in [5] various techniques can be implemented (some of them are already available in industrial milling circuits):

- The cement mill could be installed on a weighing system and its total mass can be measured. The cement load can be derived from it.
- Some mills are equipped with an "electronic ear". The measured noise is inversely proportional to the mill load.

In case the mill load obtained by the above techniques is not precise enough, then a nonlinear observer of this variable can be developed as an alternative. The authors are planning to study it in their future work.

Acknowledgements

The authors would like to acknowledge Bogazici University Reseach Fund Project No: 03A202 and TUBITAK Project No: 100E042.

References

[1] C. Ciganek, K. Kreysa, Two-parameter control system for a cement grinding process, Translat. Zement-Kalk-Gips (1991) 202–206.
[2] V. Van Breusegem, L. Chen, G. Bastin, V. Wertz, V. Werbrouck, C. de Pierpont, An industrial application of multivariable linear quadratic control to a cement mill circuit, IEEE Trans. Ind. Appl. 32 (1996) 670–677.
[3] V. Van Breusegem, L. Chen, V. Werbrouck, G. Bastin, V. Wertz, Multivariable linear quadratic control of a cement mill: an industrial application, Contr. Eng. Practice 2 (1994) 605–611.
[4] M. Boulvin, C. Renotte, A. Vande Wouwer, M. Remy, S. Tarasiewicz, P. Cesar, Modeling, simulation and evaluation of control loops for a cement grinding process, in: Proceedings of European Control Conference (CD-ROM), Brussels, Belgium, July 1997, Paper TH-E-H4.
[5] L. Magni, G. Bastin, V. Wertz, Multivariable nonlinear predictive control of cement mills, IEEE Trans. Contr. Syst. Technol. 7 (1999) 502–508.
[6] F. Grognard, F. Jadot, L. Magni, G. Bastin, R. Sepulchre, V. Wertz, Robust stabilization of a nonlinear cement mill model, IEEE Trans. Autom. Contr. 46 (4) (2001) 618–623.
[7] M.I. Jordan, Forward models: supervised learning with a distal teacher, Cognitive Sci. 16 (1992) 307–354.
[8] H. Gomi, M. Kawato, Neural network control for a closed-loop system using feedback-error-learning, Neural Networks 6 (1993) 933–946.
[9] M. Norgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, Neural Networks for Modeling and Control of Dynamic Systems, Springer, 2000.
[10] K. Hornik, M. Stinchcombe, H. White, Multilayer feed-forward networks are universal approximators, Discussion paper, Department of Economics, University of California, San Diego, La Jolla, CA, 1988.
[11] A.R. Barron, Universal approximation bounds for superpositions of a sigmoid function, IEEE Trans. Inform. Theory 39 (1993) 930–945.
[12] K.J. Astrom, B. Wittenmark, Adaptive Control, second ed., Addison-Wesley, Reading, MA, 1995.
[13] M.O. Efe, O. Kaynak, Stabilizing and robustifying the learning mechanisms of artificial neural networks in control engineering applications, Int. J. Intell. Syst. 15 (5) (2000) 365–388.
[14] M.O. Efe, O. Kaynak, B.M. Wilamowski, Stable training of computationally intelligent systems by using variable structure technique, IEEE Trans. Ind. Electron. 47 (2) (2000) 487–496.
[15] H. Sira-Ramirez, E. Colina-Morles, A sliding mode strategy for adaptive learning in adalines, IEEE Trans. Circ. Syst.—I: Fundam. Theory Appl. 42 (12) (1995) 1001–1012.
[16] X. Yu, M. Zhihong, S.M.M. Rahman, Adaptive sliding mode approach for learning in a feedforward neural network, Neural Comput. Appl. 7 (1998) 289–294.
[17] G.G. Parma, B.R. Menezes, A.P. Braga, Sliding mode algorithm for training multilayer artificial neural networks, Electron. Lett. 34 (1) (1998) 97–98.
[18] G.G. Parma, B.R. Menezes, A.P. Braga, Neural network learning with sliding mode control: the sliding mode backpropagation algorithm, Int. J. Neural Syst. 9 (3) (1999) 187–193.
[19] A.V. Topalov, O. Kaynak, On-line learning in adaptive neuro-control schemes with a sliding mode algorithm, IEEE J. SMC, Part B 31 (3) (2001) 450–455.
[20] V.I. Utkin, Sliding Modes and Their Application in Variable Structure Systems, MIR, Moscow, 1978.
[21] S.Z. Sarpturk, Y. Istefanopulos, O. Kaynak, On the stability of discrete time sliding mode control systems, IEEE Trans. Autom. Contr. 39 (10) (1987) 930–932.