Computer and Control Systems

Neuro-adaptive Modeling and Control of a Cement Mill Using a Sliding Mode Learning Mechanism

Andon V. Topalov[1*], Member, IEEE, and Okyay Kaynak[2], Fellow, IEEE

[1,2] Andon V. Topalov and Okyay Kaynak, Department of Electrical and Electronic Engineering , Mechatronics Research and Application Center, Bogazici University, Bebek, 34342 Istanbul, Turkey, e-mail : (topalov, kaynak)@boun.edu.tr
[*] On leave from Technical University of Sofia, Plovdiv branch, 4000 Plovdiv, Bulgaria

Abstract—**A novel neural network adaptive control scheme for cement milling circuits is presented. Estimates of the one-step-ahead errors in control signals are calculated through a neural predictive model of the plant and used for controller tuning. A robust on-line learning algorithm, based on the direct use of 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 plant model. The results from simulations show that both the neural model and the controller inherit some of the advantages of SMC, such as high speed of learning and robustness. Fast convergence ability and good performance on reducing mapping error are observed, leading to an improvement of the transient response of the closed-loop system.**

Index Terms—**Intelligent control; Variable structure systems; Neural networks; Adaptive control**

I. Introduction

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 milling circuit. The cement mill is fed with raw material (feed), consisting of clinker, slag, gypsum and other raw material components. After grinding, the material is introduced in a high-efficiency classifier and separated into two classes. The tailings (refused part) are returned to the mill inlet while the final product (accepted part) exits the milling circuit. The classification of the material is driven by the adjustable rotational speed and air flow rate of the classifier.

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 (kWh/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 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

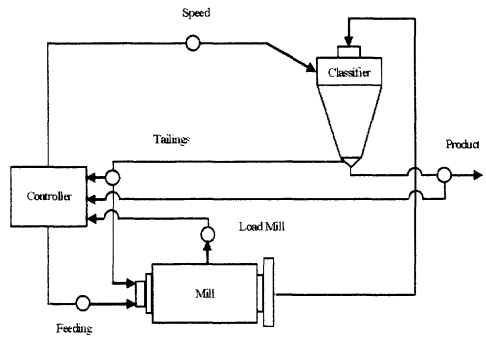


Fig. 1 Cement milling circuit

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.

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 on 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, F. 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 ce-

ment 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 multilayer feed-forward neural network (FNN).

II. Intelligent Control Approach

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.

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.

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

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, while $z^{ref}(k+1)$, and $z^{pr}(k+1)$ 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:

$$e_u(k+1) = \frac{\partial J}{\partial u} = -\frac{\partial y_f^{pr}(k+1)}{\partial u(k)} e_3(k+1) - \frac{\partial z^{pr}(k+1)}{\partial u(k)} e_4(k+1) \tag{2}$$

$$e_v(k+1) = \frac{\partial J}{\partial v} = -\frac{\partial y_f^{pr}(k+1)}{\partial v(k)} e_3(k+1) - \frac{\partial z^{pr}(k+1)}{\partial v(k)} e_4(k+1) \tag{3}$$

A FNN, shown in Fig. 3 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 outputs of the neural network plant model can be determined as follows.

$$y_f^{pr}(k+1) = f\left\{\sum_{i=1}^{q} W2_i^{y_f}(k) f_H\left[\sum_{j=1}^{n} W1_{ij}(k) I_j(k)\right]\right\}, \tag{4}$$

$$z^{pr}(k+1) = f\left\{\sum_{i=1}^{q} W2_i^{z}(k) f_H\left[\sum_{j=1}^{n} W1_{ij}(k) I_j(k)\right]\right\} \tag{5}$$

In above q is the number of neurons in the hidden layer (32 in this work), n is the number of FNN 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 $z(k-1)$. $f_{H\,i}(\cdot)$ is the activation function for the hidden node i, while $f(\cdot)$ is the linear activation function of the output nodes.

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

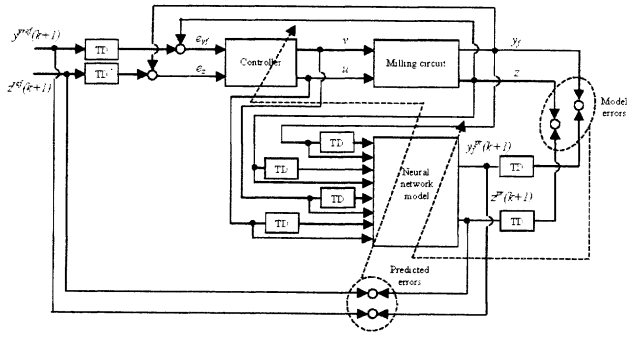


Fig. 2 The neuro-adaptive control scheme of the milling process

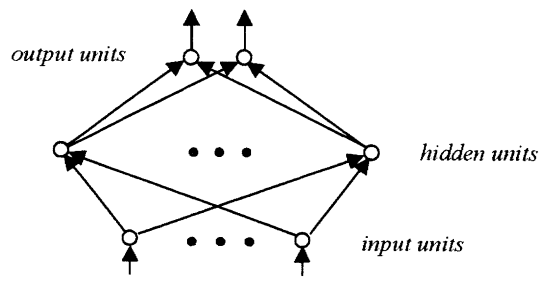


Fig. 3 Structure of the neural network predictor

$$\frac{\partial y_f^{pr}(k+1)}{\partial u(k)} = \sum_{i=1}^{q} \frac{\partial \hat{y}_f(k+1)}{\partial f_H(x_i)} \cdot \frac{\partial f_H(x_i)}{\partial x_i} \cdot \frac{\partial x_i}{\partial u(k)}$$

$$= \sum_{i=1}^{q} W2_i^{y_f}(k)\left[1 - f_H(x_i)\right] f_H(x_i) W1_{i1}(k) \quad (6)$$

where $f_H(x_i) = \frac{1}{1+e^{-x_i}}$ is a log-sigmoid function and $x_i = \sum_{j=1}^{n} W1_{ij}(k) I_j(k)$

Similarly it can be derived that:

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

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

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

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 neural 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:

$$v(k) = v(k-1) + K_{v1}(k) h_{v1}(k) + K_{v2}(k) h_{v2}(k) + K_{v3}(k) h_{v3}(k) \quad (10)$$

$$u(k) = u(k-1) + K_{u1}(k) h_{u1}(k) + K_{u2}(k) h_{u2}(k) + K_{u3}(k) h_{u3}(k) \quad (11)$$

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

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

III. The Learning Algorithm

If the tuning of the parameters $K_{lm}(k)(l = v,u;\ m = 1, 2, 3)$ is never turned off, the controllers (10) and (11) 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-specialised 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 [7].

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 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. Recently, a new on-line learning algorithm for training FNN, based on direct use of SMC strategy, was proposed by Parma et al. [8]. 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 [9]. It is currently implemented 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, ($\rho \geq 0$), respectively, for penalizing the controls v and u a simple type of optimal control criterion is achieved.

$$\varepsilon_v = e_v + \rho v; \quad \varepsilon_u = e_u + \rho u \quad (12)$$

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:

$$S_v(k) = \Delta\varepsilon_v(k) + \lambda_1 \varepsilon_v(k);$$

$$S_u(k) = \Delta\varepsilon_u(k) + \lambda_2\varepsilon_u(k) \quad (13)$$

where $\lambda_1, \lambda_2 > 0$ and

$$\Delta\varepsilon_v(k) = \varepsilon_v(k) - \varepsilon_v(k-1);$$
$$\Delta\varepsilon_u(k) = \varepsilon_u(k) - \varepsilon_u(k-1) \quad (14)$$

For the output layer of the predictive model the sliding surfaces are defined as in Eq. (15) ($\lambda_3, \lambda_4 > 0$):

$$S_{y_f}(k-1) = \Delta e_1(k) + \lambda_3 e_1(k);$$
$$S_z(k-1) = \Delta e_2(k) + \lambda_4 e_2(k) \quad (15)$$

where

$$\Delta e_1(k) = e_1(k) - e_1(k-1);$$
$$\Delta e_2(k) = e_2(k) - e_2(k-1) \quad (16)$$

The following sliding surface is defined for the hidden layer [8]:

$$S_{iH}(k-1) = \Delta e_{iH}(k) + \lambda_H e_{iH}(k) \quad (17)$$

where $\lambda_H > 0$ and

$$\Delta e_{iH}(k) = e_{iH}(k) - e_{iH}(k-1) \quad (18)$$

$$e_{iH}(k) = f'_H(x_i)\left[e_1(k)W2_i^{y_f}(k) + e_2(k)W2_i^{z}(k)\right] \quad \text{and}$$
$$x_i = \sum_{j=1}^{n} W1_{ij}(k)I_j(k) \quad (19)$$

In case $f_H(x_i) = \dfrac{1}{1+e^{-x_i}}$ is a log-sigmoid function

$$e_{iH}(k) = [1 - f(x_i)]f(x_i)\left[e_1(k)W2_i^{y_f}(k) + e_2(k)W2_i^{z}(k)\right] \quad (20)$$

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 [8], as in Eqs. (21)-(24) (for $\alpha_1 - \alpha_4 > 0$):

$$\Delta W2_i^{y_f}(k) = \alpha_1\left|e_1(k+1)\right| f_H(x_i)\,sign\left(S_{y_f}(k)\right) \quad (21)$$
$$\Delta W2_i^{z}(k) = \alpha_2\left|e_2(k+1)\right| f_H(x_i)\,sign\left(S_z(k)\right) \quad (22)$$
$$\Delta K_{vm}(k) = \alpha_3\left|\varepsilon_v(k)\right| h_{vm}(k)\,sign\left(S_v(k)\right) \quad (23)$$
$$\Delta K_{um}(k) = \alpha_4\left|\varepsilon_u(k)\right| h_{um}(k)\,sign\left(S_u(k)\right) \quad (24)$$

The weight update rule for the hidden layer of the plant model is obtained as in Eq. (25) (for $\beta > 0$):

$$\Delta W1_{ij}(k) = \beta\left|e_{iH}(k+1)\right| I_j(k)\,sign\left(S_{iH}(k)\right) \quad (25)$$

For the system to be controlled, the bounds for α_1, α_2, α_3, α_4 and β (and λ_1, λ_2, λ_3, λ_4 and λ_H) should be derived in advance in order to predict convergence and stability properties. The way how to derive them is presented in the extended version of this paper that is to appear in the Elsevier Journal of Process Control latter this year.

IV. 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 results are presented in Fig. 4(a)-(e).

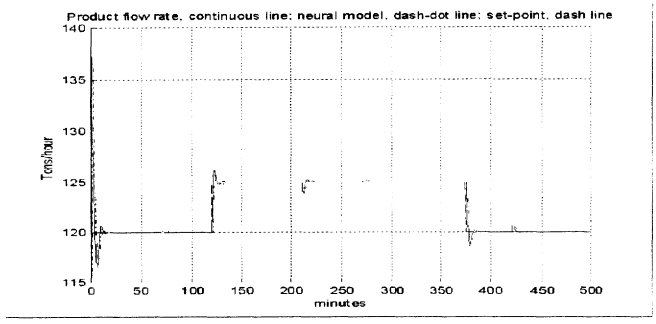
The nonlinear model of the milling circuit introduced in [5] is used in simulations. This model describes the evolution of the load in the mill and it is able to reproduce some (unstable) behavior that has been observed with the LQ controller in the feedback loop. It consists of three nonlinear differential equations explaining the evolution of three states

$$\begin{cases} T_f\dot{y}_f = -y_f + (1-\alpha(z,v,d))\varphi(z,d) \\ \dot{z} = -\varphi(z,d) + u + y_r \\ T_r\dot{y}_r = -y_r + \alpha(z,v,d)\varphi(z,d) \end{cases} \quad (26)$$

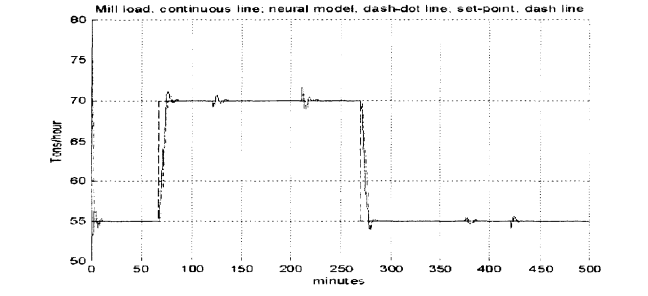
with $\alpha(z,v,d) = \dfrac{\varphi^m v^n}{L_\alpha + \varphi^m v^n}$; $L_\alpha = 570^m 170^n\left(\dfrac{570}{450} - 1\right)$ (tons/h)^m(r/min)^n; $m = 0.8$; $n = 4$; $\varphi(z,d) = \max\{0; (-dL_{\varphi 1}z^2 + L_{\varphi 2}z)\}$; $L_{\varphi 2} = 16.50$ (h)$^{-1}$; $L_{\varphi 1} = 0.1116$ (Tons 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].

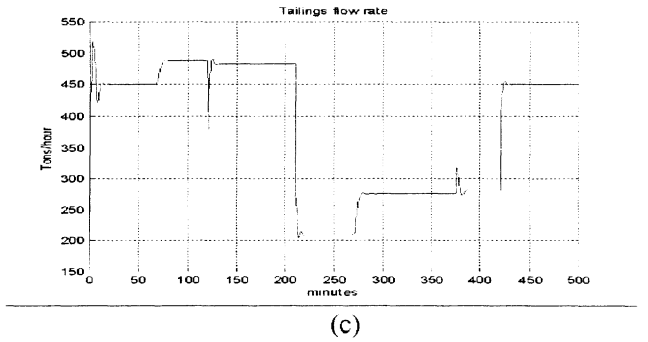
From the second nonlinear differential equation and the definition of φ, it can be noticed that there is a constraint on the maximum value of the circulating load (feed plus tailings (rejected oversized particles) flow rates). By an easy manipulation of the three nonlinear differential equations it follows 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



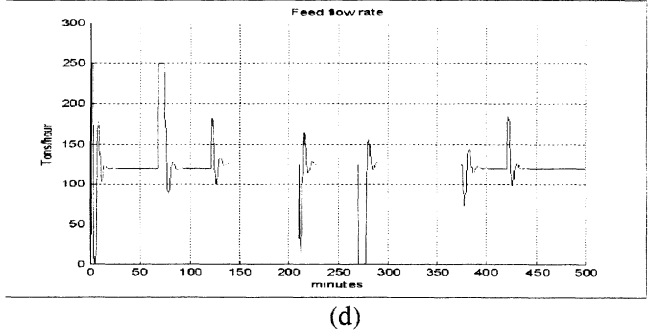
(a)



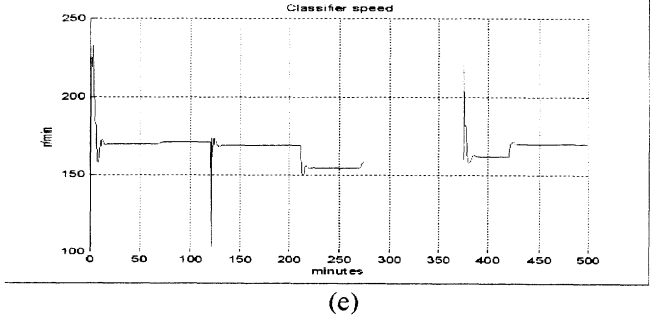
(b)



(c)



(d)



(e)

Fig. 4 Simulation results

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. 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 in, the set-point for the product flow rate y_f^{ref} changes from 120 to 125 tons/h within the time interval 120 – 375 in, and the set-point for the load in the mill z changes from 55 to 70 tons during the period from 67 to 270 in. The accepted values of the different coefficients and constants during simulations are: $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_H = 1$; $\alpha_1, \alpha_2, \beta = 5 \times 10^{-4}$; $\alpha_3, \alpha_4 = 8 \times 10^{-3}$; $\rho = 5 \times 10^{-2}$; $\delta = 1 \times 10^{-2}$; $\eta = 1 \times 10^{-8}$.

The control signals u and v are restricted within the interval $[0 \quad 250]$. 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].

V. Conclusions

A novel neuro-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 [8], as 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. Simulations 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.

VI. Acknowledgements

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

VII. References

[1] C. Ciganek and K. Kreysa, "Two-parameter control system for a cement grinding process," Translation of *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 Transactions on Industrial Applications*, 32, 1996, pp. 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," *Control Engineering Practice*, 2, 1994, pp. 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 the 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 Transactions on Control Systems Technology*, 7, 1999, pp. 502-508.

[6] F. Grognard, F. Jadot, L. Magni, G. Bastin, R Sepulchre, V. Wertz, "Robust stabilization of a nonlinear cement mill model," *IEEE Transactions on Automatic Control*, vol. 46, no. 4, 2001, pp. 618-623.

[7] K. J. Astrom, B. Wittenmark, *Adaptive Control*, second ed., Addison-Wesley, Reading, MA, 1995, 573 p.

[8] G. G. Parma, B. R. Menezes, A. P. Braga, "Sliding mode algorithm for training multilayer artificial neural networks," *Electronics Letters*, vol. 34, no. 1, 1998, pp. 97-98.

[9] A. V. Topalov, O. Kaynak, "On-line learning in adaptive neurocontrol schemes with a sliding mode algorithm," *IEEE Journal on SMC*, part B, vol. 31, no. 3, 2001, pp. 455-450.