Adaptive Control of Antilock Braking System Using Grey Multilayer Feedforward Neural Networks

Erdal Kayacan, Yesim Oniz, Okyay Kaynak and Andon V. Topalov*
Bogazici University, Department of Electrical and Electronics Engineering
34342, Bebek, Istanbul, Turkey
{erdal.kayacan,yesim.oniz, okyay.kaynak, topalov}@ieee.org
* On leave from Technical University of Sofia, campus Plovdiv, Bulgaria

Abstract

In this paper, a grey neuro-adaptive control algorithm is suggested for Antilock Braking Systems (ABS). The concept of grey system theory, which has a certain prediction capability, offers an alternative approach to conventional control methods. A multilayer neural network and a grey predictor, GM(1,1) model, are combined in the approach proposed in the paper. The grey neural network controller is examined under several different operating conditions and it is shown that the proposed control algorithm anticipates the upcoming values of wheel slip and optimal wheel slip, and takes the necessary action to keep the wheel slip at the desired value. The simulation results indicate that the proposed controller has the ability to control the nonlinear system accurately with little oscillations and with no steady-state error.

1. Introduction

In most control engineering applications, the performance of a controller is directly related to the accuracy of the mathematical model employed for the controlled system. However, there are many situations in industrial control systems that the control engineer faces the difficulty of incomplete or insufficient information. The reasons for this may be the lack of modeling information, the inaccuracies in measurement and the inappropriateness of the employed control variables [1]. In order to overcome these difficulties, model free control methodologies based on computational intelligence techniques are commonly used throughout the literature.

Artificial Neural Networks (ANNs)-based controllers learn from input-output data and efficiently determine the most appropriate control action to apply [2]. However, in order to draw a conclusion about the performances of the overall system, application of ANNs in feedback control systems requires the investigation of their properties such as stability and robustness to uncertainties and noises coming from both inside and outside of the system [3, 4]. In order to improve performance, ANNs-based controllers should have the capability to adapt their parameters (training operation) in such a way that the robustness, stability and fast convergence would be guaranteed. Two most common ways of training ANNs are the gradient based [5]-[6] and Variable Structure Systems (VSSs)-based methods [7]-[9]. In the literature, when one aim is to build a neuro-adaptive feedback system, the latter one is more preferred as it is faster and more robust.

By using a sliding mode control (SMC)-based learning algorithm for a controller built on multilayer feedforward neural networks (MFNNs), the robustness of the overall system can be guaranteed. A separate sliding surface using the learning error and its time derivative is defined for each layer of the network in [8] for training purpose. In [9], online learning in MFNNs is considered with one sliding surface defined using only the learning error. In [7], the proposed method in [9] has been proved through real-time experiments. The results presented in [7] are quite encouraging for real-time applications. However, the control method essentially based on conventional ANNs theory, has no predictive capabilities. This paper proposes a grey prediction based neural controller that can overcome the stated shortcoming.

Grey system theory was first introduced in early eighties by Professor Deng Ju-long [11]. Since then, the theory has become quite popular with its ability to deal with the systems that have partially unknown parameters. It is therefore a good candidate to real-time control systems that have model uncertainties. In this paper, a grey predictor is used to forecast the wheel slip to be used by the ANNs-based controller of the vehicle's ABS system.

978-0-7695-3495-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ICMLA.2008.15

2. MFNN controller structure and initial assumptions

Consider a full-connected, three layer, feedforward, perceptron neural network with p input and one output neurons. The number of neurons in the hidden layer, n, can be set arbitrary. Let us assume that $X(t) = [x_1(t), \ldots, x_p(t)]^T$ is the vector of the time-varying input signals augmented by the bias term fed to the network. $U_H(t) = [u_{H_1}, \ldots, u_{H_n}]^T$ is the vector representing the output signals of neurons in the hidden layer. The activation function $f(.)$ is assumed to be continuous, nonlinear and differentiable, and the maximum value of its time derivative is denoted by B_A. The neuron in the output layer has a linear activation function and $u(t)$ is the scalar output of the network [7].

The matrix of the time varying connections' weights between the neurons in the input and the hidden layer is denoted as $W1(t)_{(nxp)}$, where each element $w1_{i,j}(t)$ means the weight connection of the neuron i from its input j. In a similar manner, $W2(t)_{(1xn)}$ is the vector of connections' weights between the neurons in the hidden layer and the output node. Both $W1(t)_{(nxp)}$ and $W2(t)_{(1xn)}$ are augmented by including bias weight components. The output signal u_{H_i} of the i^{th} neuron from the hidden layer and the output signal of the network $u(t)$ are as follows:

$$u_{H_i} = f\left(\sum_{j=1}^{p} w1_{i,j} x_j\right) \tag{1}$$

$$u(t) = \sum_{i=1}^{n} w2_i u_{H_i} \tag{2}$$

It is assumed that the input vector fed to the network and its time derivative are bounded:

$$\begin{aligned} \| X(t) \| &\leq B_X \quad \forall t \\ \| \dot{X}(t) \| &\leq B_{\dot{X}} \quad \forall t \end{aligned} \tag{3}$$

It is also assumed that the magnitude of all vectors row $W1_i(t)$ constituting the matrix $W1(t)$ and the elements of the vector $W2(t)$ are bounded at each instant of time t by means of:

$$\begin{aligned} \| W1_i(t) \| &\leq B_{W1} \quad \forall t \\ \| W2_i(t) \| &\leq B_{W2} \quad \forall t \end{aligned} \tag{4}$$

The desired control input signal $u_d(t)$ and its time derivative $\dot{u}_d(t)$ are supposed to be also bounded signals:

$$\begin{aligned} |u_d(t)| &\leq B_{u_d} \quad \forall t \\ |\dot{u}_d(t)| &\leq B_{\dot{u}_d} \quad \forall t \end{aligned} \tag{5}$$

In the inequalities (3), (4) and (5), B_X, $B_{\dot{X}}$, B_{W1}, B_{W2},B_{u_d} and $B_{\dot{u}_d}$ respectively are some known positive constants, and $i = 1, 2, \ldots, n$.

3. Sliding mode neuro-adaptive control approach

Sliding mode control provides an effective approach to control nonlinear plants. In this study, a VSSs-based algorithm is employed to train the MFNN controller. The proposed learning algorithm establishes an inner sliding motion in terms of the controller parameters leading the command error towards zero. The outer sliding motion concerns the controlled system the state tracking error vector of which is simultaneously forced towards the origin of the phase space. The switching function for the system under control, s_p, is defined as:

$$s_p = \dot{e} + \lambda_s e \tag{6}$$

where e, $\dot{e}$ and λ_s are the error of the overall system, its time derivative, and a positive constant determining the slope of the sliding surface, respectively. Furthermore, a zero adaptive learning error level for the MFNN controller, s_c, is defined as:

$$s_c = u - u_d \tag{7}$$

where u_d is the desired control input signal.

The learning algorithm to update the neural networks' weights should be derived in such a way that the system will be enforced to move along the sliding surface, i.e. $s_p = \dot{e} + \lambda_s e = 0$. For an inner sliding motion of the controller parameters to occur, the corresponding update laws can be defined as follows [7]:

$$\dot{w}1_{i,j} = -\left(\frac{w2_i x_j}{X^T X}\right) \alpha sign(s_c) \tag{8}$$

$$\dot{w}2_i = -\left(\frac{u_{H_i}}{{U_H}^T U_H}\right) \alpha sign(s_c) \tag{9}$$

where α is a sufficiently large positive constant satisfying:

$$\alpha > n B_A B_{W1} B_{\dot{X}} B_{W2} + B_{\dot{u}_d} \tag{10}$$

For any initial condition $s_c(0)$, the adaptive learning error, $u - u_d$, will converge to zero within a finite time t_h [7].

Because of not having the necessary information about the desired control input signal u_d, s_c cannot be easily determined. As a solution to this problem, a mathematical relation between the sliding surface s_p and the zero adaptive learning error s_c can be specified as follows [7]:

$$s_c^{(n)} = \Phi(s_p^{(m)}) \tag{11}$$

Three conditions for the function Φ are defined in [12]. In this study, the following selection is made:

$$\Phi(x) = x \tag{12}$$

If it is assumed that $m = 0$ and $n = 0$, then s_c goes to zero as s_p goes to zero. Hence, an outer sliding motion of the controlled system will occur simultaneously, and a perfect tracking can be achieved. Then, an equivalence between the sliding mode control of the plant and the sliding mode adaptive learning inside of the MFNN-based controller will have place.

To deal with the chattering problem, which usually occurs because of the discontinuous control input and high speed switching in sliding mode control, the discontinuous switching function is replaced by the following continuous one where $\delta \geq 0$ [13]:

$$sgn(s_c) \approx \frac{s_c}{|s_c| + \delta} \tag{13}$$

4. Grey neural net control

Grey models can predict the future outputs of systems with high accuracy without a mathematical model of the actual system. In grey systems theory, GM(n,m) denotes a grey model, where n is the order of the difference equation and m is the number of variables. Although various types of grey models can be mentioned, most of the previous researchers have focused their attention on GM(1,1) model in their predictions because of its computational efficiency. The differential equations of the GM(1,1) model have time-varying coefficients. In other words, the model is renewed as the new data become available to the prediction model.

In order to smooth the randomness, the primitive data obtained from the system to form the GM(1,1) is subjected to an operator, named Accumulating Generation Operator (AGO)[14]. The differential equation (i.e. GM(1,1)) thus evolved is solved to obtain the $n - step$ ahead predicted value of the system. Finally, using the predicted value, the Inverse Accumulating Generation Operator (IAGO) is applied to find the predicted values of the primitive data.

Consider a single input and single output system. Assume that the time sequence $X^{(0)}$ represents the outputs of the system:

$$X^{(0)} = \left(x^{(0)}(1), x^{(0)}(2), ..., x^{(0)}(n)\right), n \geq 4 \tag{14}$$

where $X^{(0)}$ is a non-negative sequence and n is the sample size of the data. When this sequence is subjected to AGO, the following sequence $X^{(1)}$ is obtained. It is obvious that $X^{(1)}$ is monotone increasing.

$$X^{(1)} = \left(x^{(1)}(1), x^{(1)}(2), ..., x^{(1)}(n)\right), n \geq 4 \tag{15}$$

where

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), k = 1, 2, 3....., n \tag{16}$$

The generated mean sequence $Z^{(1)}$ of $X^{(1)}$ is defined as:

$$Z^{(1)} = \left(z^{(1)}(1), z^{(1)}(2), ..., z^{(1)}(n)\right) \tag{17}$$

where $z^{(1)}(k)$ is the mean value of adjacent data, i.e.

$$z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1), k = 2, 3, ..., n \tag{18}$$

The least square estimate sequence of the grey difference equation of GM(1,1) is defined as follows [14]:

$$x^{(0)}(k) + az^{(1)}(k) = b \tag{19}$$

The whitening equation is therefore as follows:

$$\frac{dx^1(t)}{dt} + ax^{(1)}(t) = b \tag{20}$$

In above, $[a, b]^T$ is a sequence of parameters that can be found as follows:

$$[a, b]^T = (B^T B)^{-1} B^T Y \tag{21}$$

where

$$Y = \left[x^{(0)}(2), x^{(0)}(3), ..., x^{(0)}(n)\right]^T \tag{22}$$

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ . & . \\ . & . \\ . & . \\ -z^{(1)}(n) & 1 \end{bmatrix} \tag{23}$$

According to (20), the solution of $x^{(1)}(t)$ at time k:

$$x_p^{(1)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a}\right] e^{-ak} + \frac{b}{a} \tag{24}$$

To obtain the predicted value of the primitive data at time (k+1), IAGO is used to establish the following grey model.

$$x_p^{(0)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a}\right] e^{-ak}(1 - e^a) \tag{25}$$

and the predicted value of the primitive data at time $(k+H)$:

$$x_p^{(0)}(k+H) = \left[x^{(0)}(1) - \frac{b}{a}\right] e^{-a(k+H-1)}(1 - e^a) \tag{26}$$

The structure of the grey multilayer feedforward neural network (GMFNN) controller is presented in Fig. 1. In this scheme, upcoming value of wheel slip is considered instead of its present value, which enables the GMFNN controller to have prediction capability.

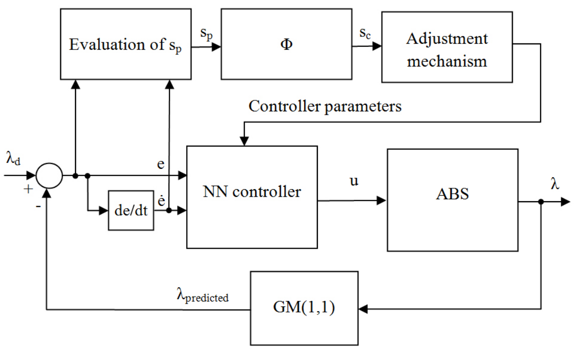

Figure 1. Grey multilayer feedforward neural network (GMFNN) controller structure

5. Description of controlled object

The free body diagram of the quarter vehicle model describing longitudinal motion of the vehicle and angular motion of the wheel under braking is presented in Fig. 2 [15]. Although the model is quite simple, it preserves the fundamental characteristics of an actual system. In deriving the dynamic equations of the system, several assumptions are made. First, only longitudinal dynamics of the vehicle is considered. The lateral and vertical motions are neglected. Furthermore, it is assumed that there is no interaction between the four wheels of the vehicle [16].

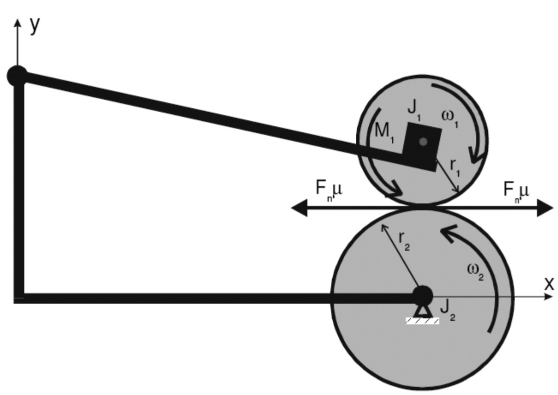

Figure 2. Schematic view of quarter vehicle model

Regarding the model, there are three torques acting on the upper wheel: Braking torque, friction torque in the upper bearing and the friction torque among the wheels. Similarly, two torques are acting on the lower wheel: The friction torque in the lower bearing and the friction torque between these wheels.

During deceleration, a braking torque is applied to the upper wheel, which causes wheel speed to decrease. According to Newton's second law, the equation of the motion of the system can be written as:

Table 1. System Parameters

ω_1	Angular velocity of the upper wheel
ω_2	Angular velocity of the lower wheel
T_B	Braking torque
r_1	Radius of the upper wheel
r_2	Radius of the lower wheel
J_1	Moment of inertia of the upper wheel
J_2	Moment of inertia of the lower wheel
d_1	Viscous friction coefficient of the upper wheel
d_2	Viscous friction coefficient of the lower wheel
F_n	Total normal load
μ	Road adhesion coefficient
λ	Wheel slip
λ_R	Reference slip
F_t	Road friction force
M_{10}	Static friction of the upper wheel
M_{20}	Static friction of the lower wheel
M_g	Moment of gravity acting on balance lever

$$J_1\dot{\omega}_1 = F_t r_1 - (d_1\omega_1 + M_{10} + T_B) \quad (27)$$

$$J_2\dot{\omega}_2 = -(F_t r_2 + d_2\omega_2 + M_{20}) \quad (28)$$

F_t in (27) and (28) stands for the road friction force which is given by Coulomb Law:

$$F_t = \mu(\lambda)F_n \quad (29)$$

F_n is calculated by the following equation:

$$F_n = \frac{d_1\omega_1 + M_{10} + T_B + M_g}{L(\sin\phi - \mu(\lambda)\cos\phi)} \quad (30)$$

where L is the distance between the contact point of the wheels and the rotational axis of the balance lever and ϕ is the angle between the normal in the contact point and the line L.

Under normal operating conditions, the rotational velocity of the wheel would match the forward velocity of the car. When the brakes are applied, braking forces are generated at the interface between the wheel and road surface, which causes the wheel speed to decrease. As the force at the wheel increases, wheel slip will occur between the tire and the road surface. The wheel speed will tend to be lower than vehicle speed. The parameter used to specify this difference in these velocities is called wheel slip (λ), and it is defined as:

$$\lambda = \frac{r_2\omega_2 - r_1\omega_1}{r_2\omega_2} \quad (31)$$

The road adhesion coefficient is a nonlinear function of some physical variables including wheel slip and it can be approximated by the following formula [15]:

$$\mu(\lambda) = \frac{c_4\lambda^p}{a+\lambda^p} + c_3\lambda^3 + c_2\lambda^2 + c_1\lambda \qquad (32)$$

For the numerical values used in this study, the reader can refer to [16].

The road adhesion coefficient is a nonlinear function of some physical variables including the velocity of the vehicle and wheel slip. The LuGre friction model [17] deals with the dependence of friction on velocity. In [18], a pseudo static expression for friction force is given as:

$$F(\eta, v) = -F_N h\left[1 + 2\gamma\frac{h}{\sigma_0 L|\eta|}\left(e^{-\frac{\sigma_0 L|\eta|}{2h}} - 1\right)\right] - F_N\sigma_2 V_r \qquad (33)$$

where
$\eta = \frac{\lambda}{\lambda-1}$, $h = \mu_c + (\mu_s - \mu_c)e^{-|\frac{V_r}{V_s}|^{1/2}}$, $\gamma = 1 - \frac{\sigma_1|\eta|}{Rwh}$

As can be seen from (33), if the velocity of the vehicle changes, the curve will change as well. However, the change in the curve will happen faster than a change in the vehicle velocity. Hence, it is possible to calculate the approximated peak value of braking force produced by tire/road friction for each time step.

6. Simulation results

In this section, a number of computer simulated dynamic responses are obtained to investigate the performance of the proposed control algorithms. All figures below show simulation results for a car with initial longitudinal velocity of $V = 30m/s$ maneuvering on a straight line. The reference wheel slip is considered as a function of vehicle velocity. A pseudo-static curve is used to calculate the reference wheel slip and corresponding tire friction coefficient. These values are used to construct a table, which relates the vehicle speed to the peak values of tire road friction coefficient and to the reference wheel slip. The sampling time is 0.01ms.

A three layer feedforward neural network with one hidden layer of hyperbolic tangent neurons and a linear scalar output layer is employed. There are 16 neurons in the hidden layer, and the learning rate is specified as $\alpha = 15$. At the beginning of each simulation, the initial weights of the ANN are set randomly to prove that a priori knowledge is not necessary for the controlled system. During the simulation studies, it has been observed that the number of the neurons in the hidden layer does not effect the performance of the ANN controller drastically. The selection of the learning rate is made by trial and error method. In the sliding mode controller part, a sliding surface with a slope of $\lambda_s = 10$ is designed. The step size for the grey predictor GM(1,1), H, is considered as a constant value which equals to 12.

Fig. 3-Fig. 5 illustrate system responses of MFNN and MFNN coupled to a grey predictor (GMFNN) for a reference wheel slip which is function of vehicle velocity. Although both controllers give sufficiently high performances, GMFNN exhibits less oscillatory response when compared to conventional MFNN. It can be observed that the system is getting unstable at the end of the simulation. This stems from the fact that the equilibrium point of ABS is an unstable equilibrium point, and when the velocity of the car is below a threshold value, it is very difficult to force the slip to remain at a constant value. In real life cases, there is a relay type switch for braking system of the vehicle. If the velocity of the vehicle decreases to a minimum value, the ABS controller is switched off, and regular brakes are applied.

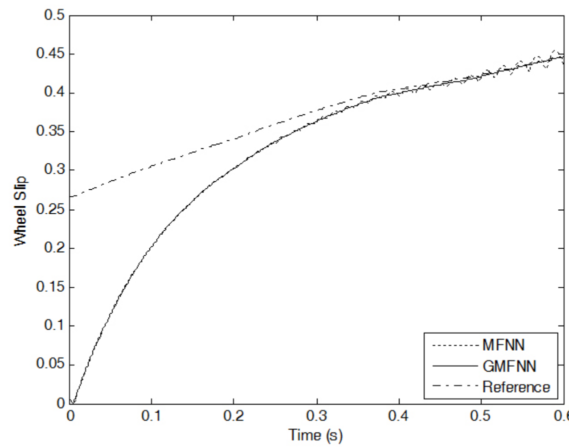

Figure 3. Wheel slip for MFNN and GMFNN

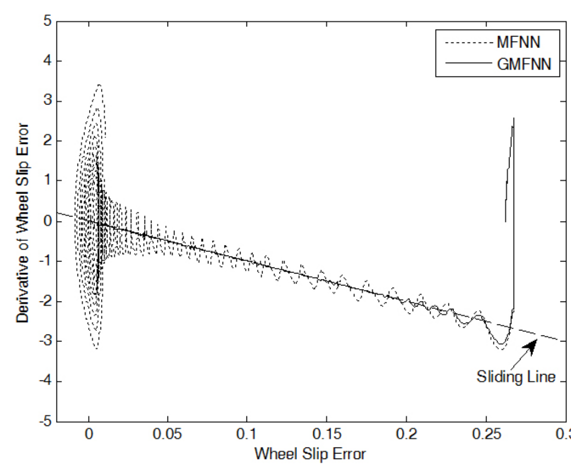

Figure 4. Phase space behavior for MFNN and GMFNN

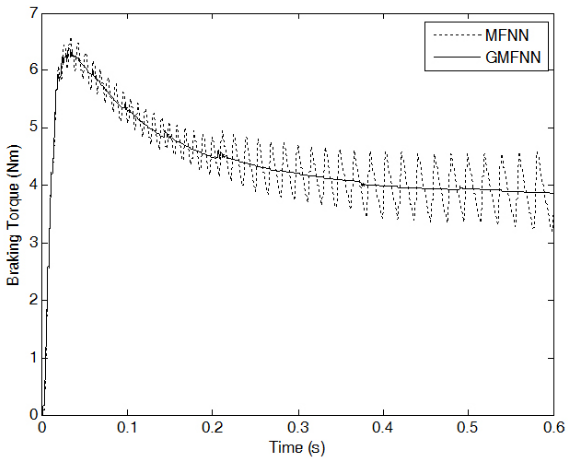

Figure 5. Braking torque for MFNN and GMFNN

7. Conclusion and future works

In this study, a MFNN controller and a MFNN controller coupled to grey predictor for ABS have been proposed. According to various simulation results, the most attractive characteristic of GMFNN controller is the robustness in the presence of the uncertainties in the system. In such a condition, GMFNN gives more accurate and less oscillatory responses. Hence, braking operation will be more stable and the performance of ABS system will be increased. Encouraged by these simulation results, an experimental investigation is about to be launched.

8. Acknowledgements

The authors gratefully acknowledge the financial support provided within the TUBITAK, Project Number 107E284 and the Ministry of Education and Science of Bulgaria Research Fund Project BY-TH-108/05.

References

[1] E. Kayacan and O. Kaynak, "An Adaptive Grey Fuzzy PID Controller with Variable Prediction Horizon", *Proceedings of SCIS&ISIS 2006*, Tokyo, Japan, 2006, pp. 760-765.

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

[3] M. O. Efe and O. Kaynak, Stabilizing and Robustifying the Learning Mechanisms of Artificial Neural Networks in Control Engineering Applications, *Int. J. Intell. Syst.*, vol. 15, no. 5, 2000, pp. 365-388.

[4] M. O. Efe, O. Kaynak and B. M. Wilamowski, Stable Training of Computationally Intelligent Systems by Using Variable Structure Systems Technique, *IEEE Trans. Ind. Electron.*, vol. 47, no. 2, 2000, pp. 487-496.

[5] T.-C. Chen and T.-T. Sheu, Model Reference Neural Network Controller for Induction Motor Speed Control, *IEEE Trans. Energy Convers.*, vol. 17, no. 2,2002, pp. 157-163.

[6] C.-H. Tsai, H.-Y. Chung, and F.-M. Yu, Neuro-Sliding Mode Control with its Applications to Seesawsystems, *IEEE Trans. Neural Networks*, vol. 15, no. 1, 2004, pp. 124-134.

[7] A. V. Topalov, G. L. Cascella, V. Giordano, F. Cupertion and O. Kaynak, Sliding Mode Neuro-Adaptive Control of Electric Drives, *IEEE Trans. Industrial Electronics*, vol. 54, no. 1, 2007, pp. 671-679.

[8] G. G. Parma, B. R. Menezes and A. P. Braga, Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks, *Electron. Lett.*, vol. 34, no. 1, 1998, pp. 97-98.

[9] N. G. Shakev, A. V. Topalov and O. Kaynak, *Sliding Mode Algorithm for On-Line Learning in Analog Multilayer Feedforward Neural Networks*, "Artificial Neural Networks and Neural Information Processing", Kaynak, Ed. et al. Berlin, Germany: Springer-Verlag, 2003, Lecture Notes in Computer Science, pp. 1064-1072.

[10] Lin, Y. and Liu, S., "A Historical Introduction to Grey Systems Theory", *Proceedings of the IEEE SMC 2004*, vol. 1, The Netherlands, 2004, pp. 2403-2408.

[11] J. L. Deng, Control Problems of Grey System, *Syst. and Contr. Lett.*, vol. 1, 1982, pp. 288-294.

[12] 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", *ASME J. Dynamic Syst., Measur. Control*, vol. 122, 2000, pp. 632-640.

[13] C. Edwards and S. K. Spurgeon, *Sliding Mode Control Theory and Applications*, Taylor and Francis Ltd., USA, 1998.

[14] J. L. Deng, Introduction to Grey System Theory, *The Journal of Grey System*, vol. 1, 1989, pp. 1-24.

[15] "The Laboratory Antilock Braking System Controlled from PC", *User's Manual, Inteco Ltd., Poland*.

[16] Y. Oniz, E. Kayacan and O. Kaynak, "Simulated and Experimental Study of Antilock Braking System Using Grey Sliding Mode Control", *Proceedings of the IEEE SMC 2007*, Montreal, Canada, 2007, pp. 90-95.

[17] J. Svendenius, "Tire Models for Use in Braking Applications", Department of Automatic Control, Lund Institute of Technology, 2003, Sweden.

[18] J. Yi, L. Alvarez, R. Horowitz and D. W. Canudas, "Adaptive Emergency Braking Control using a Dynamic Tyre/road Friction model", *Proceedings of 39th IEEE Conference on Decision and Control*, 2000, Sydney.