ROBUST NEURAL IDENTIFICATION OF ROBOTIC MANIPULATORS USING DISCRETE TIME ADAPTIVE SLIDING MODE LEARNING

Andon V. Topalov* **and Okyay Kaynak**

Department of Electrical and Electronic Engineering, Mechatronics Research and Application Center, Bogazici University, Bebek, 34342 Istanbul, Turkey
**On leave from Control Systems Dept., TU Sofia, br. Plovdiv, 4000 Plovdiv, Bulgaria*
{topalov, kaynak}@boun.edu.tr

Abstract: The problem of identification of uncertain nonlinear systems using feedforward neural networks is investigated. The weights of the neural identifier are updated on-line by a discrete-time learning algorithm based on the sliding mode control technique, which is well known with its robustness to uncertainties. The learning parameters are adjusted to force the error between the actual and desired neural network outputs to satisfy a stable difference error equation and a quasi-sliding mode on the zero learning error is established. The behaviour of the proposed discrete-time algorithm is illustrated by using it for the neural identification of an experimental robotic manipulator. The results show that the neural model inherits some of the advantages of the sliding mode control approach, such as high speed of learning and robustness, and is able to follow the actual robot joint trajectories with a high accuracy.

Keywords: Variable-structure systems, Neural-network models, Learning algorithms, Robotic manipulators, Identification algorithms.

1. INTRODUCTION

Identification of systems having uncertainties and impreciseness constitutes a central part in the practice of systems engineering. The possibility to reach high performance goals in control tasks is usually directly related to the degree of the model accuracy that can be achieved.

Robotic manipulators are hard to control nonlinear systems where perfect knowledge of their parameters is unattainable by conventional techniques because of the time-varying inertia and gravitational loads, and the uncertainties in the model used for joint friction. To guarantee a good tracking performance, robust adaptive control approaches combining conventional methods with new learning techniques are required. In such cases, the adoption of an accurate, robust and fast on-line system identification procedure is frequently a key requirement.

In the literature, most widely used approaches for system identification are based on Least Mean Square, Recursive Least Square, Gradient Descent, Levenberg-Marquardt method or their variants (Ljung, 1999; Haykin, 1991; Krishnapura, and Jutan, 1997; Efe, and Kaynak, 2000). The necessity of costly hardware for data storage, high sensitivity to changes in the input signal, the possibility of getting stuck to local minima or the need for matrix inversions at some intermediate stages are the prime difficulties in their implementation.

When one has to deal with systems with uncertain dynamics a possible way to design a robust identification scheme is to utilize the Variable Structure Systems (VSS) theory in constructing the parameter adaptation mechanism of the identifier (Efe *et al.*, 2002). VSS-based control, implemented in the way of sliding mode control (SMC), is well-known with its robustness to uncertainties. The use of this technique introduces certain invariance properties in a predefined subspace of the state space, defined by the error and its several time derivatives. The basic idea behind is to restrict the state space of the given plant through a hypersurface passing though the origin, called sliding surface, whose dynamics is simpler than the original plant dynamics. The method enforces a state-space trajectory driving, the plant from the initial conditions to the designed sliding surface in finite time. Once there, the plant remains on the surface and its dynamics is substituted by the surface dynamics. In the literature the mode of motion lasting until the hitting of the hypersurface is called the reaching mode. The mode on the surface is named sliding mode and the control theory uses the term SMC due to the latter dynamic behaviour. For adequately designed surfaces, the invariance property exists, guaranteeing an inherent robustness, because the new dynamics do not depend on the plant parameters.

Thanks to their universal approximation capabilities, neural networks can be used as an efficient implementation tool for modelling the complex input/output relations of robot dynamics, being able to solve problems like variable-coupling complexity and state dependency. During the last decade several neural network models and learning schemes have been suggested in the literature for on-line learning of robot dynamics (Karakasoglu *et al.*, 1993; Katic and Vukobratovic, 1995; Lewis *et al.*, 1999; Topalov *et al.*, 2002). Among them are some SMC-based schemes, motivated by the studies that demonstrate the effectiveness of variable structure control in handling imprecision and uncertainties.

One of the earliest studies that suggest the establishment of sliding modes for adaptive learning in Adaline networks is due to Sira-Ramirez and Colina-Morles (1995). Later, Yu *et al* (1998) have extended this approach by introducing adaptive uncertainty bound dynamics.

A sliding mode learning approach for analog multilayer feedforward neural networks (FNNs) has been presented in (Parma *et al.*, 1998), by defining separate sliding surfaces for each network layer. A further contribution to the subject can be found in (Shakev *et al.*, 2003) in which the approach of Sira-Ramirez and Colina-Morles (1995) is extended to allow on-line learning in FNNs with a scalar output. Its main difference from the algorithm presented in (Parma *et al.*, 1998) is that only one sliding surface is defined for the entire network, making it computationally simpler and more suitable for real time applications.

Although from a theoretical point of view the development of VSS-based learning algorithms for analogue (i.e. continuous time) neurons seems easier and straightforward, the discrete-time algorithms are more convenient for practical implementation. The discrete-time sliding mode control (DTSMC) design issues have been addressed in (Gao *et al.*, 1995; Sira-Ramirez, 1991). The stability issues in DTSMC have been presented in (Sarpturk *et al.*, 1987) and the sufficient conditions for convergence have been determined. The first results on adaptive learning in discrete-time neural networks, for both single and multilayer perceptrons, based on the theory of the quasi-sliding modes in discrete time dynamical systems are presented by Sira-Ramirez and Zak (1991). These algorithms constitute the basis of the later proposed identification and control schemes in (Kuschewski *et al.*, 1993). Another learning algorithm for FNNs is recently developed in (Parma *et al.*, 1999). It may be considered as the discrete time counterpart of the continuous time algorithm earlier proposed in (Parma *et al.*, 1998).

In this paper, we first present a new discrete-time sliding mode technique to the adaptive learning of FNNs and then apply it to minimize the error between the system to be identified and an implemented neural identifier. In the well known back-propagation algorithm the learning procedure aims at minimizing the error function by suitable adjustment of the learning parameters. In particular, one calculates the gradient of the error function with respect to the learning parameters starting at the output nodes and works back towards the input nodes through the hidden layers. Once the gradient is calculated, the learning parameters are adjusted using the gradient descent method. In contrast to the backpropagation procedure the weight adaptation algorithm proposed in this paper controls the error dynamics of FNN's. It is described by a difference equation for the error, which is the difference between the desired and the actual FNN outputs at discrete instants of time. The learning parameters are adjusted to force the error to satisfy this stable difference error equation, rather than to minimize an error function. In other words, the proposed adaptation rule can be viewed as a discrete-time controlled dynamical system where the network weights can be considered as representing the system state at time k, the correction terms of the present values of the weights can be viewed as controllers, and the error signal $e(k)$ is interpreted as a single output signal. It is shown in the paper that the algorithm presented in (Sira-Ramirez and Zak, 1991) can be considered as a variant of the proposed learning algorithm.

The remainder of the paper is organized as follows. Section 2 describes concisely the applied sliding mode learning algorithm. Section 3 presents the results from the dynamics identification of the second and third joint of an experimental robot manipulator. Finally, in Section 4 a conclusion remark is made for the achievements in this paper.

2. ON-LINE LEARNING IN MULTILAYER PERCEPTRON NETWORKS BASED ON SLIDING MODE CONCEPT

2.1 Initial assumptions and definitions.

Consider a feedforward neural network with one hidden layer and a scalar output. We will use the following definitions:

$X = \left[x_1, x_2, \ldots x_p\right]^T \in \mathbb{R}^p$ is the augmented by a bias term input vector (input pattern) which is assumed fixed during the learning iterations.

$Y_H(k) = \left[y_{H_1}(k), y_{H_2}(k), ..., y_{H_n}(k) \right]^T \in \mathbb{R}^n$ is the vector of the output signals of the neurons in the hidden layer, where k is the time index or iteration.

$netY_H(k) = \left[net\, y_{H_1}(k), net\, y_{H_2}(k), ..., net\, y_{H_n}(k) \right]^T$ is the vector of the net input signals of the hidden neurons. It is computed as

$$netY_H(k) = W1(k)X \quad (1)$$

where $W1(k) \in \mathbb{R}^{n \times p}$ is the matrix of the time-varying weights of the connections between the neurons of the input and the hidden layer. Each element $w1_{i,j}(k)$ of this matrix represents the weight of the connection of the corresponding hidden neuron i from its input j.

$y(k) \in \mathbb{R}$ is the time-varying network output. It can be calculated as follows:

$$y(k) = W2(k)\Phi\left[netY_H(k)\right] = W2(k)Y_H(k) \quad (2)$$

where $W2(k) \in \mathbb{R}^{1 \times n}$ is the vector of the weights of the connections between the neurons in the hidden layer and the output node. Both $W1(k)$ and $W2(k)$ are considered augmented by including the bias weight components for the corresponding neurons.

$\Phi\left[netY_H(k)\right] = \left[f_1\left(net\, y_{H_1}(k)\right), ..., f_n\left(net\, y_{H_n}(k)\right) \right]^T$,

$\Phi : \mathbb{R}^n \to \mathbb{R}^n$ is an operator which elements f_i are the activation functions of the neurons in the hidden layer. It will be assumed here that $f_i : \mathbb{R} \to \mathbb{R}$ is such that $f_i\left(-net\, y_{H_i}\right) = -f_i\left(net\, y_{H_i}\right)$ for $i = 1, ..., n$.

The so called tan-sigmoid activation function $\left(tan\text{-}sig(x) = \dfrac{1-e^{-x}}{1+e^{-x}} \right)$, common to neural networks, has been used in the experiments.

The neuron in the output layer is considered with a linear activation function.

The scalar signal y_d represents the desired output of the neural network and $e(k) = y(k) - y_d \in \mathbb{R}$ is the error at time k.

2.2 The discrete-time SMC-based learning algorithm.

In the proposed here VSS-based learning approach, the zero value of the learning error coordinate $e(k)$ is defined as a time-varying sliding surface, i.e.

$$S\left(e(k)\right) = S(k) = e(k) = y(k) - y_d = 0 \quad (3)$$

In the continuous SMC design the well known stability condition to be satisfied for a sliding mode to occur is (Edwards and Spurgeon, 1998)

$$S(t)\dot{S}(t) < 0 \quad (4)$$

In the discrete-time implementation of the sliding mode methodology a non-ideal sliding (quasi-sliding) regime will inevitably appear, since the control input is computed and applied to the system at discrete instants. It is clear that the condition (4) which assures the sliding motion is no longer applicable in discrete-time systems. Thus, a discrete-time sliding mode condition must be imposed. The simplest approach is to substitute the derivative by the forward difference as in (5)

$$\left[S(k+1) - S(k) \right] S(k) < 0 \quad (5)$$

but this represents the necessary, not sufficient condition for the existence of a quasi-sliding motion (Sarpturk *et al.*, 1987). It does not assure any convergence of the state trajectories onto the sliding manifold and may result in an increasing amplitude chatter of the state trajectories around the hyperplane, which means instability. A necessary and sufficient condition assuring both sliding motion and convergence onto the sliding manifold is given by Sarpturk *et al.* (1987), of the form:

$$\left| S(k+1) \right| < \left| S(k) \right| \quad (6)$$

The above condition can be decomposed into two inequalities as:

$$\left[S(k+1) - S(k) \right] sign\, S(k) < 0 \quad (7)$$

and

$$\left[S(k+1) + S(k) \right] sign\, S(k) > 0 \quad (8)$$

where (7) and (8) are known as sliding condition and convergence condition, respectively.

The network should be continuously trained in such a way that the sliding mode conditions (7) and (8) will be enforced. To enable $S = 0$ is reached, the following theorem is used:

Theorem 1: If the adaptation law for the weights $W1(t)$ and $W2(t)$ is chosen respectively as

$$\Delta W1(k) = -\frac{2\, netY_H(k)\, X^T}{X^T X} \quad (9.a)$$

$$W1(k+1) = W1(k) + \Delta W1(k) \quad (9.b)$$

and

$$\Delta W2(k) = -2W2(k) + \frac{Y_H^T(k)}{Y_H^T Y_H} \alpha\, sign\, e(k) \quad (10.a)$$

$$W2(k+1) = W2(k) + \Delta W2(k) \quad (10.b)$$

with $\alpha \in \mathbb{R}$ being the adaptive reduction factor satisfying $0 < \alpha < 2\left|e(k)\right|$, then, for any arbitrary initial condition $e(0)$, the learning error $e(k)$ will converge asymptotically to zero and a quasi-sliding motion will be maintained on $e = 0$.

Proof: One can check that the following string of equations is satisfied.

$$\begin{aligned}\Delta e(k+1) &= e(k+1) - e(k)\\ &= y(k+1) - y(k)\\ &= W2(k+1)Y_H(k+1) - W2(k)Y_H(k)\\ &= \left[W2(k) + \Delta W2(k)\right]Y_H(k+1) - W2(k)Y_H(k)\\ &= W2(k)\left[Y_H(k+1) - Y_H(k)\right] + \Delta W2(k)Y_H(k+1)\\ &= W2(k)\left\{\Phi\left[netY_H(k+1)\right] - \Phi\left[netY_H(k)\right]\right\}\\ &\quad + \Delta W2(k)\Phi\left[netY_H(k+1)\right] \end{aligned} \tag{11}$$

Note that

$$\begin{aligned} netY_H(k+1) &= W1(k+1)X\\ &= \left[W1(k) + \Delta W1(k)\right]X\\ &= netY_H(k) + \Delta W1(k)X \end{aligned} \tag{12}$$

Substituting (12) and (9.a) into (11) yields

$$\begin{aligned}\Delta e(k+1) &= W2(k)\left\{\Phi\left[netY_H(k) + \Delta W1(k)X\right]\right.\\ &\quad \left. -\Phi\left[netY_H(k)\right]\right\}\\ &\quad +\Delta W2(k)\Phi\left[netY_H(k) + \Delta W1(k)X\right]\\ &= W2(k)\left\{\Phi\left[-netY_H(k)\right] - \Phi\left[netY_H(k)\right]\right\}\\ &\quad +\Delta W2(k)\Phi\left[-netY_H(k)\right] \end{aligned} \tag{13}$$

Since Φ is odd by assumption, the previous error equation becomes

$$\begin{aligned}\Delta e(k+1) &= -2W2(k)\Phi\left[netY_H(k)\right]\\ &\quad -\Delta W2(k)\Phi\left[netY_H(k)\right]\\ &= -\left[2W2(k) + \Delta W2(k)\right]Y_H(k) \end{aligned} \tag{14}$$

Substituting (10.a) into the above equation gives

$$\begin{aligned}\Delta e(k+1) &= \left[-2W2(k) + 2W2(k)\right.\\ &\quad \left. - \frac{Y_H^T(k)\alpha\, sign\, e(k)}{Y_H^T(k)Y_H(k)}\right]Y_H(k)\\ &= -\alpha\, sign\, e(k) \end{aligned} \tag{15}$$

By multiplying both sides of eq. (15) by $e(k)$ it follows that

$$\Delta e(k+1)e(k) = -\alpha\left|e(k)\right| < 0 \tag{16}$$

which means that the sliding condition (5) or (7) is satisfied.

Eq. (15) can be also rewritten as follows

$$e(k+1) = e(k) - \alpha\, sign\, e(k) \tag{17}$$

By adding to the both sides of eq. (17) $e(k)$ and subsequent multiplication with $sign\, e(k)$ the following equation can be obtained

$$\left[e(k+1) + e(k)\right]sign\, e(k) = 2\left|e(k)\right| - \alpha \tag{18}$$

It follows from eq. (18) that the convergence condition (8) will be satisfied for all $0 < \alpha < 2\left|e(k)\right|$ and $e(k) \neq 0$. This proof is a sufficient condition for the quasi-sliding mode to occur.

Remark 1: Note that eq. (17) describes the FNN error dynamics.

In particular if $\alpha = \beta\left|e(k)\right|$ with $0 < \beta < 2$ is used it follows that

$$e(k+1) = (1-\beta)e(k) \tag{19}$$

which coincides with the result obtained by Sira-Ramirez and Zak (1991), and shows that the error will converge asymptotically to 0 at a rate of $1-\beta$.

In order to reduce the "chattering" phenomenon, when a small sampling period is adopted, the following approximation for the signum function has been adopted:

$$sign\, e(k) \approx \frac{e(k)}{\left|e(k)\right| + \delta} \tag{20}$$

with δ being a small constant.

3. ON-LINE IDENTIFICATION OF ROBOTIC MANIPULATOR DYNAMICS

In this section, the effectiveness of the proposed discrete-time sliding mode learning approach is evaluated by simulation studies carried out on a realistic computer model of the dynamics of five degrees of freedom experimental robotic manipulator (CRS CataLyst-5, produced by Quanser). The identification task has been restricted to the design of neural network identifiers for two consecutive robot joints (the shoulder and elbow joint) by using one neural network per joint. The manipulator dynamics has been carefully simulated using Matlab SimMechanics toolbox by taking into account the data about the frame assignments for each link, distances between each two joint axes, default orientation, mass and inertia tensors of each link with respect to the center of gravity, information about friction dynamics, gear mechanisms and motor transfer functions of each joint (see Fig. 1). The developed model has motor input voltages as input signals and joint angle and angular speed of each joint as output signals.

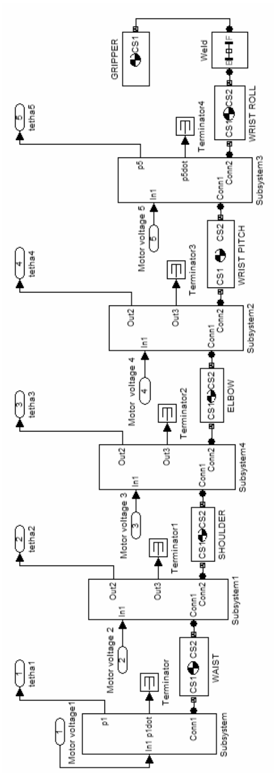


Fig. 1. Model of manipulator dynamics within SimMechanics Toolbox

Feedforward neural network topology with one hidden layer consisting of 13 neurons with tan-sigmoid activation functions, twelve inputs, and output layer with one linear neuron has been considered as an appropriate neural identifier (NI) structure. Two identical FNNs (one per joint) where used as NIs.
The input vector of each of the two NIs has been constructed as follows:

$$X(k)=[q_2(k),q_2(k-1),q_3(k),q_3(k-1),\dot{q}_2(k),\dot{q}_2(k-1),\\ \dot{q}_3(k),\dot{q}_3(k-1),v_2(k),v_2(k-1),v_3(k),v_3(k-1)]^T \quad (21)$$

where $q_i(k)$, $q_i(k-1)$, $\dot{q}_i(k)$, $\dot{q}_i(k-1)$ are the joint angles and velocities for the i-th robot joint, and $v_i(k)$ and $v_i(k-1)$ are the implemented voltages (control actions) to the i-th joint DC drive at time instant k and $k-1$ respectively. As shown in equation (21), the manipulator coupled dynamics is taken into account by feeding each of the two NIs with the variables related to both robot joints (see also Fig. 2). The scalar outputs of the two NIs represent the estimated joint coordinates at time step $k+1$ for the second and the third robot joint respectively:

$$\hat{q}_2(k+1)=\hat{\Gamma}_2\left[X(k)\right]; \quad \hat{q}_3(k+1)=\hat{\Gamma}_3\left[X(k)\right] \quad (22)$$

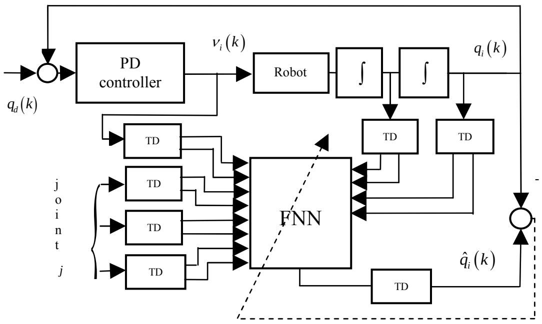
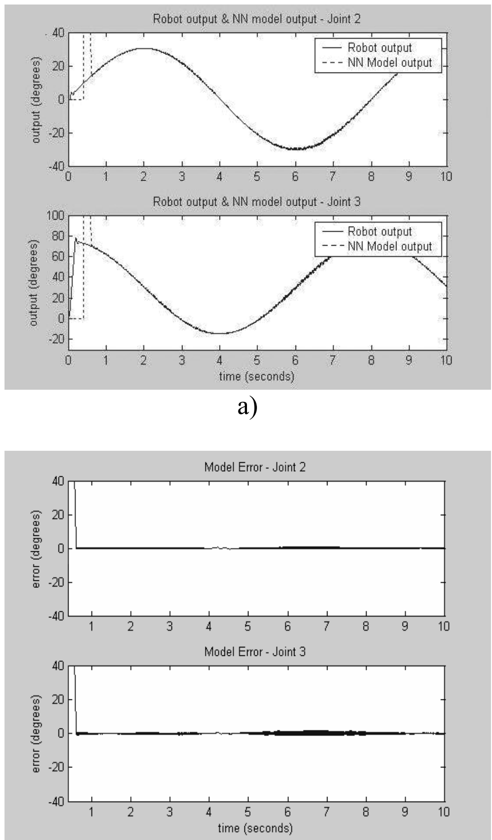


Fig. 2. Forward dynamics identification of the robotic manipulator using FNNs.

where $\Gamma_2(\cdot)$ and $\Gamma_3(\cdot)$ are non-linear functions of the arguments v, q and $\dot{q}$, and $\hat{\Gamma}_2(\cdot)$, $\hat{\Gamma}_3(\cdot)$ are their estimates learned by the NIs. The sampling time has been taken 1 ms and $\alpha=|e(k)|$ has been adopted.
The reference signals to be followed where sinusoidal ones with frequency π/2 rad/s and amplitude 30 degrees, and 45 degrees for the second and third joint respectively. The results are shown on Fig. 3.



a)

b)

Fig. 3. Simulation results in neural network identification of manipulator dynamics.

The NIs learning has been initialized 0.4 s after the beginning of the trajectory tracking task. As it can be seen from the simulations the two identifiers are able to follow accurately the respective joint trajectories without chattering.

4. CONCLUSIONS

An application of the sliding mode technique to the learning algorithms of FNNs which are utilized to implement neural identifiers has been discussed. In order to guarantee the existence of a quasi-sliding mode, new discrete-time learning law is proposed to adapt on-line the weights of NI. This law has a sliding mode structure. The applicability of the proposed learning scheme is illustrated on the example of on-line dynamics identification of an experimental manipulator. The results show the excellent performance of the proposed neural network identifiers with discrete-time sliding mode on-line learning. The learning structures are coming into some of the advantages of variable structure systems, such as high speed of learning and robustness.

5. ACKNOWLEDGEMENT

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

REFERENCES

Edwards C. and S.K. Spurgeon (1998). *Sliding mode control: theory and applications*, Taylor & Francis, Ltd.

Efe, M.O. and O. Kaynak (2000). A comparative study of soft computing methodologies in identification of robotic manipulators. *Robotics and autonomous systems*, **30**, pp.221-230.

Efe, M.O., O. Kaynak and B.M. Wilamowski (2002). A robust identification technique for time-varying ARMA processes based on variable structure systems theory. *Mathematical and computer modeling of dynamical systems*, **8** (2), pp. 185-198.

Gao, W., Y. Wang and A. Homaifa (1995). Discrete-time variable structure control systems. *IEEE Trans. on indust. electr.*, **42**, pp. 117-122.

Haykin, S. (1991). *Adaptive Filter Theory.* Prentice_Hall International, New Jersey.

Karakasoglu, A., S.I. Sudharsanan and M.K. Sundareshan (1993). Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators. *IEEE Trans. on neural networks*, **4**, pp. 919-930.

Katic, D.M. and M.K. Vukobratovic (1995). Highly efficient robot dynamics learning by decomposed connectionist feedforward control structure. *IEEE Trans. on syst., man, and cybernetics*. **25** (1), pp. 145-158.

Krishnapura, V.G. and A. Jutan (1997). ARMA Neuron Networks for Modeling Nonlinear Dynamical Systems. *The Canadian journal of chemical engineering*, **75**, pp. 574-582.

Kuschewski, J.G., S. Hui and S.H. Zak (1993). Application of feedforward networks to dynamical systems identification and control. *IEEE Trans. cont. syst. techn.*, **1** (1), pp.37-49.

Lewis, F.L., S. Jagannathan and A. Yesildirek (1999). *Neural network control of robot manipulators and nonlinear systems*. Taylor & Francis, New York.

Ljung, A. (1999). *System Identification, 2^nd ed.* Prentice-Hall International, New Jersey.

Parma, G.G., B.R. Menezes and A.P. Braga (1998). Sliding mode algorithm for training multilayer artificial neural networks. *Electronic letters*. **34** (1), pp. 97-98.

Parma, G.G., B.R. Menezes and A.P. Braga (1999). Neural network learning with sliding mode control: the sliding mode backpropagation algorithm. *Int. journal neural syst.*, **9** (3), pp.187-193.

Sarpturk, S.Z., Y. Istefanopulos and O. Kaynak (1987). On the stability of discrete-time sliding mode control systems. *IEEE Trans. on automatic control*, **32**, pp.930-932.

Shakev, N.G., A.V. Topalov and O. Kaynak (2003). Sliding mode algorithm for on-line learning in analog multilayer feedforward neural networks, In: *Artificial neural networks and neural information processing - ICANN/ICONIP 2003, Lecture notes in computer science* (Kaynak et al. (Ed)), pp.1064-1072, Springer-Verlag, Berlin.

Sira-Ramirez H. (1991). Non-linear discrete variable structure systems in quasi-sliding mode. *International journal of control*, **54**, pp. 1171-1187.

Sira-Ramirez, H. and S.H. Zak (1991). The adaptation of perceptrons with applications to inverse dynamics identification of unknown dynamic systems. *IEEE Trans. syst. man and cybern.*, **21** (3), pp. 634-643.

Sira-Ramirez, H. and E. Colina-Morles (1995). A sliding mode strategy for adaptive learning in Adalines. *IEEE Trans. on circuits and systems-I: fundamental theory and applications*. **42** (12), pp. 1001-1012.

Topalov A.V., O. Kaynak and N.G. Shakev (2002). Neural network identification, predictive modeling and control with a sliding mode learning mechanism: an application to the robotic manipulators. *Proc. first int. IEEE symp. On intelligent systems*. **1,** Varna, pp. 102-107.

Yu, X., M. Zhihong and S.M. M. Rahman (1998). Adaptive sliding mode approach for learning in a feedforward neural network. *Neural computing & applications*. **7**, pp. 289-294.