2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Robust On-line Training of Multilayer Perceptrons via Direct Implementation of Variable Structure Systems Theory

Andon V. Topalov* and Okyay Kaynak
Bogazici University, 80815, Bebek, Istanbul, Turkey
*on leave from Technical University of Sofia, branch Plovdiv, Bulgaria

Abstract-**An algorithm based on direct implementation of variable structure systems theory approach is proposed for on-line training of multilayer perceptrons. Network structures which have multiple inputs, single output and one hidden layer are considered and the weights are assumed to have capabilities for continuous time adaptation. The zero level set of the network learning error is regarded as a sliding surface in the learning parameters space. A sliding mode trajectory can be brought on and reached in finite time on such a sliding manifold. Results from simulated on-line identification task for a two-link planar manipulator dynamics are also presented.**

I. INTRODUCTION

Variable structure systems (VSS) with sliding mode were first proposed in the early 1950s as a simple robust control approach. Recent studies have accentuated that the robustness and stability properties of intelligent control strategies can be also analyzed through the use of sliding mode control (SMC) theory [1]. The results in [2] have shown that the convergence properties of the gradient-based training algorithms widely used in artificial neural networks can be improved by an indirect implementation of SMC approach. Direct use of SMC strategy for adaptive learning in Adaline networks has been suggested in [8] and [10] and control applications of the method considered in [8] have been studied in [3]. An on-line learning algorithm for training of multilayer perceptrons, has been recently proposed by G. G. Parma et al. [6]. In [9] the application of the above learning algorithm in neuro-adaptive control schemes has been investigated.

In the present paper the sliding mode strategy for adaptive learning in analog Adaline networks proposed in [8] is further extended to more general classes of multilayer neuron arrangements which do not have the limited approximation capabilities of early proposed Perceptron and Adaline networks. The imposed here limitation related to the required scalar network output cannot be considered as too restrictive with respect to the applicability of the proposed algorithm because it is always possible to have structures consisting of two or more multilayer feedforward neural networks (FNN) sharing the same inputs. The main difference of the developed new algorithm from the one presented earlier in [6] is that it makes use of only one sliding surface instead of two which makes it simpler.

The main body of the paper contains four sections. Section II presents the proposed SMC-based adaptive learning algorithm for analog FNN with a scalar output. Results from simulation experiments are shown in Section III. Finally, section IV summarizes the findings of this work.

II. ON-LINE WEIGHTS ADAPTATION IN MULTILAYER FEEDFORWARD NETWORKS WITH A SCALAR OUTPUT BASED ON SLIDING MODE CONCEPT

A. *Initial Assumptions and Definitions*

Consider the two-layered feedforward neural network shown on Figure 1.

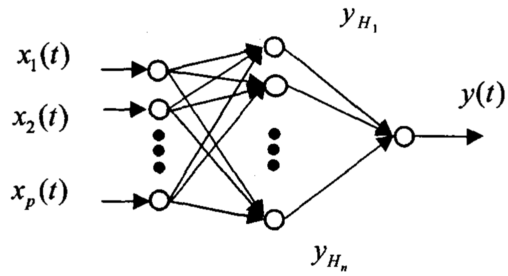

Fig. 1. Multilayer perceptron network with a scalar output

We will use the following definitions:

$X(t)=\left[x_1(t), \ldots, x_p(t)\right]^T$ - vector of the time-varying input signals augmented by the bias term.

$Y_H(t)=\left[y_{H1}(t), \ldots, y_{Hn}(t)\right]^T$ - vector of the output signals of the neurons in the hidden layer.

$y(t)$ - scalar signal representing the time-varying output of the network.

$W1(t)_{(n\times p)}$ - matrix of the time-varying connections' weights between the neurons in the input and the hidden layer, where each matrix's element $w1_{i,j}(t)$ means the weight of the connection of the neuron i from its input j.

$W2(t)_{(1\times n)}=\left[w2_1(t),\ldots,w2_n(t)\right]$ - vector of the time-varying connections' weights between the neurons in the hidden layer and the output node. Both $W1(t)_{(n\times p)}$ and $W2(t)_{(1\times n)}$ are considered augmented by including the bias weight components for the neurons in the hidden layer and the output neuron respectively.

$f(\cdot)$ - nonlinear, differentiable, monotonously increasing activation function of the neurons in the hidden layer of the network. The neuron in the output layer is considered with a linear activation function.

This work was supported by Bogazici University Research Grant 03A202.

An assumption is made that the input vector $X(t)$ and it's time derivative are bounded, i.e.

$$\|X(t)\| = \sqrt{x_1^2(t) + \ldots + x_p^2(t)} \le B_X \quad \forall t$$
$$\|\dot{X}(t)\| = \sqrt{\dot{x}_1^2(t) + \ldots + \dot{x}_p^2(t)} \le B_{\dot{X}} \quad \forall t \tag{1}$$

where B_X and $B_{\dot{X}}$ are known positive constants.

It will be assumed that, due to the physical constraints, the magnitude of all vectors row $W1_i(t)$ constituting the matrix $W1(t)$ and the elements of the vector $W2(t)$ are also bounded at each instant of time t by means of

$$\|W1_i(t)\| = \sqrt{wl_{i,1}^2(t) + wl_{i,2}^2(t) + \ldots + wl_{i,p}^2(t)} \le B_{W1} \quad \forall t$$
$$|w2_i(t)| \le B_{W2} \quad \forall t \tag{2}$$

for some known constants B_{W1} and B_{W2}, where $i = 1, 2, \ldots, n$.

The scalar signal $y_d(t)$ represents the time-varying desired output of the network. It will be assumed that $y_d(t)$ and $\dot{y}_d(t)$ are also bounded signals, i.e.

$$|y_d(t)| \le B_{y_d}, \quad |\dot{y}_d(t)| \le B_{\dot{y}_d} \quad \forall t \tag{3}$$

where B_{y_d} and $B_{\dot{y}_d}$ are positive constants.

The output signal y_{H_i} of the i-th neuron from the hidden layer and the output signal of the network $y(t)$ are defined as

$$y_{H_i} = f\left(\sum_{j=1}^{p} wl_{i,j} x_j\right) \tag{4}$$

$$y(t) = \sum_{i=1}^{n} w2_i y_{H_i} \tag{5}$$

It is assumed that the derivative of the neurons activation function is also bounded, i.e.

$$0 < A_i(t) = f'\left(\sum_{j=1}^{p} wl_{i,j} x_j\right) \le B_A \quad \forall i, j \tag{6}$$

We define the learning error $e(t)$ as the scalar quantity obtained from

$$e(t) = y(t) - y_d(t) \tag{7}$$

B. The SMC-based On-line Learning Algorithm

Using the SMC approach, we define the zero value of the learning error coordinate $e(t)$ as a time-varying sliding surface, i.e.

$$S(e(t)) = e(t) = y(t) - y_d(t) = 0 \tag{8}$$

which is the condition that guarantees that the neural network output $y(t)$ coincides with the desired output signal $y_d(t)$ for all time $t > t_h$ where t_h is the hitting time of $e = 0$.

Definition 2.1: A sliding motion will have place on a sliding manifold $S(e(t)) = e(t) = 0$, after time t_h if the condition $S(t)\dot{S}(t) = e(t)\dot{e}(t) < 0$ is true for all t in some nontrivial semi open subinterval of time of the form $[t, t_h) \subset (-\infty, t_h)$.

The learning algorithm for the neural network weights $W1(t)$ and $W2(t)$ should be derived in such a way that the sliding mode condition of the above definition will be enforced.

Let us denote as "$sign(e(t))$" the signum function, defined as follows:

$$sign(e) = \begin{cases} 1 & \text{for } e(t) > 0 \\ 0 & \text{for } e(t) = 0 \\ -1 & \text{for } e(t) < 0 \end{cases} \tag{9}$$

To enable $S = 0$ is reached, we have the following theorem:

Theorem 2.2: If the learning algorithm for the weights $W1(t)$ and $W2(t)$ is chosen respectively as

$$\dot{w}1_{i,j} = -\left(\frac{w2_i x_j}{X^T X}\right)\alpha \, sign(e) \tag{10}$$

$$\dot{w}2_i = -\left(\frac{y_{Hi}}{Y_H^T Y_H}\right)\alpha \, sign(e) \tag{11}$$

with α being sufficiently large positive constant satisfying

$$\alpha > nB_A B_{W1} B_{\dot{X}} B_{W2} + B_{\dot{Y}_d} \tag{12}$$

then, for any arbitrary initial condition $e(0)$, the learning error $e(t)$ will converge to zero during a finite time t_h which may be estimated as

$$t_h \le \frac{|e(0)|}{\alpha - nB_A B_{W2} B_{W1} B_{\dot{X}} - B_{\dot{Y}_d}} \tag{13}$$

and a sliding motion will be maintained on $e = 0$ for all $t > t_h$.

The proof of the theorem 2.2 is presented in [7].

C. Bounded noise added to the network inputs

Let us consider now an augmented vector-valued norm-bounded external perturbation input, denoted by $H(t) = (\eta_1(t),...,\eta_p(t))$, which is added to the input vector $X(t)$. We will assume here that its last element $\eta_p(t)$ equals zero. This means that the constant input to the bias weight is considered as a fixed value without an influence of perturbation signals on it.

It is assumed also that the perturbation input $H(t)$ is not "larger" than the input $X(t)$, i.e.,

$$\|H(t)\| = \sqrt{\eta_1^2(t) + ... + \eta_p^2} \le B_\eta < B_x \quad \forall t \tag{14}$$

$\dot{H}(t)$ is assumed to be also bounded i.e.

$$\|\dot{H}(t)\| = \sqrt{\dot{\eta}_1^2(t) + ... + \dot{\eta}_p^2(t)} \le B_{\dot{\eta}} \quad \forall t \tag{15}$$

The perturbed learning error $\hat{e}(t) = y(t) - y_d(t)$ is now as follows

$$\hat{e}(t) = \sum_{i=1}^{n} w2_i f\left[\sum_{j=1}^{p} w1_{i,j}(x_j + \eta_j)\right] - y_d(t) =$$
$$= \sum_{i=1}^{n} w2_i f\left[W1_i(t)\left(X(t) + \mathrm{H}(t)\right)\right] - y_d(t) \tag{16}$$

Let us consider the following perturbed adaptation law for the weights in FNN:

$$\dot{w}1_{i,j} = -\left(\frac{w2_i(x_j + \eta_j)}{(X+H)^T (X+H)}\right)\alpha\, sign(\hat{e}) \tag{17}$$

$$\dot{w}2_i = -\left(\frac{\hat{y}_{Hi}}{\hat{Y}_H^T \hat{Y}_H}\right)a\, sign(\hat{e}) \tag{18}$$

where $\hat{y}_{H_i} = f\left[\sum_{j=1}^{p} w1_{i,j}(x_j + \eta_j)\right]$, and

$$\hat{Y}_H(t) = \left[\hat{y}_{H_1}(t), \ldots, \hat{y}_{H_n}(t)\right]^T$$

The robustness result is summarized in the following theorem whose proof is similar to that of Theorem 2.2.

Theorem 2.3: A sliding motion will have place on the zero learning error manifold of a FNN including a perturbed input vector if the adaptation law for the weight vectors $w1_{i,j}$ and $w2_i$ is chosen as in (17) and (18) with α being a positive constant satisfying

$$\alpha > nB_A B_{W1} B_{W2}\left(B_{\dot{X}} + B_{\dot{\mathrm{H}}}\right) + B_{\dot{Y}_d} \tag{19}$$

For any arbitrary initial condition $\hat{e}(0)$, the perturbed learning error will converge to zero in $\hat{t}_h$, estimated by

$$\hat{t}_h \le \frac{|\hat{e}(0)|}{\alpha - nB_A B_{W1} B_{W2}\left(B_{\dot{X}} + B_{\dot{\mathrm{H}}}\right) - B_{\dot{Y}_d}} \tag{20}$$

inspite of all possible assumed (bounded) values of the perturbation inputs and their time derivatives. Moreover a sliding motion is sustained on $\hat{e}(t) = 0$ for all $t > \hat{t}_h$.

III. Simulation Results

FNN are commonly used for on-line modeling, identification and adaptive control purposes in case variations in process dynamics or in disturbance characteristics are present. In this section, the effectiveness of the proposed on-line learning algorithm is evaluated on the example of on-line forward dynamics identification task of a simple two-link planar manipulator. The manipulator was modeled as two rigid links of length 0.5 m each with point masses equal to 10 kg and 8 kg placed at the distal ends of the links. The dynamic equations of the manipulator can be found in [4]. A decentralized control strategy with independent PID joint controls was implemented during the experiment.

Two identical FNN structures (one per joint) having 25 neurons in the hidden layer each were used and the so-called *series-parallel* identification model [5] was implemented. The manipulator coupled dynamics was also taken into account, so each of the two neural identifiers was receiving on its inputs signals from both manipulator joints. The value used for the variable structure gain α was set to $\alpha = 5$. To alleviate the "chattering" phenomena the following standard substitution was adopted for the ideal switch function.

$$sign\, e(t) \approx \frac{e(t)}{|e(t)| + \delta} \tag{21}$$

with $\delta = 0.05$.

A typical trajectory task was considered during which the manipulator arm was required to follow reference trajectories q_{d1} and q_{d2}. They were selected as:

$$q_{d1} = -0.77 + 0.8\sin((2\pi t/4.8) - \pi/2)$$
$$q_{d2} = -0.8 - 0.8\sin((2\pi t/4.8) - \pi/2) \tag{22}$$

The results from the simulations are shown on Figure 2. The following denotations are used: dash-doted lines are the joints actual trajectories, neural model outputs and joints control signals are plotted with solid lines. The neural network learning (tracking) error response $e_i(t)$ for each of the two robot joints is shown to converge to zero fast.

For comparison and network learning performance evaluation, simulations were also carried out with the input signals subject to a computer generated additive bounded noise. The generated noise signals for the two inputs of the manipulator links were discrete-time stochastic processes normally distributed at each instant of time with zero mean and standard deviation $\sigma = 1$ and $\sigma = 0.1$ for the first and second robot joint respectively. The simulation results are shown on Figure 3.

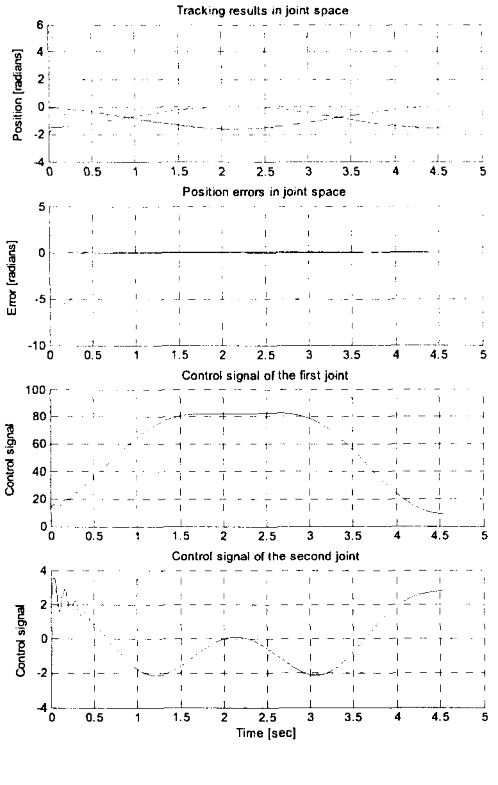

Fig. 2. Neural network identification of a two-link manipulator forward dynamics

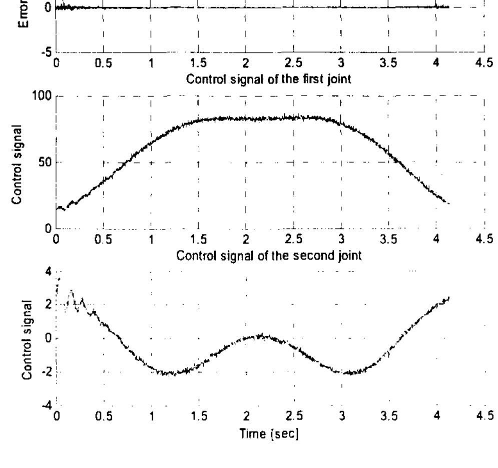

Fig. 3. Forward dynamics identification with bounded noise input for a two-link manipulator.

The value of α was set to be $\alpha = 12$. The noisy states were used to confirm the sliding mode adaptive strategy in accordance with (17) and (18).

IV. Conclusions

In this paper a new on-line learning algorithm for analog FNN with a scalar output has been presented. It is based on direct implementation of VSS theory. The algorithm robustly drives the learning error to zero in a finite time. Its convergence has been analyzed based on the task for on-line forward dynamics identification of a two-link robot manipulator. The presented simulation results show the effectiveness of the proposed approach.

References

[1] Y. Byungkook and W. Ham, "Adaptive fuzzy sliding mode control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, Vol. 6, pp. 315-321, May 1998.

[2] M. O. Efe and O. Kaynak, "Stabilizing and robustifying the learning mechanisms of artificial neural networks in control engineering applications," *Int. J. Intelligent Systems*, Vol. 15, No. 5, pp. 365-388, 2000.

[3] M. O. Efe, O. Kaynak, and X. Yu, "Sliding mode control of three degrees of freedom anthropoid robot by driving the controller parameters to an equivalent regime," *ASME J. of Dynamic Systems, Measurement and Control*, Vol. 122, pp. 632-640, 2000.

[4] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*. Macmillan Publishing Company, 1993.

[5] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1992.

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

[7] N. G. Shakev, A. V. Topalov and O. Kaynak "Slding Mode Algorithm for Online Learning in Analog Multilayer Feedforward Neural Networks", In the book: Kaynak et al. (Eds.): *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003, Lecture Notes in Computer Science*, Springer-Verlag, pp.1064-1072, 2003.

[8] H. Sira-Ramirez and E. Colina-Morles, "A sliding mode strategy for adaptive learning in Adalines," *IEEE Trans. on Circuits and Systems – I: Fundamental Theory and Applications*, Vol. 42, No. 12, pp. 1001-1012, 1995.

[9] A. V. Topalov and O. Kaynak, "On-line learning in adaptive neurocontrol schemes with a sliding mode algorithm," *IEEE Trans. on Syst., Man and Cyber.*, (part B), Vol. 31, No. 3, pp. 445-450, June 2001.

[10] X. Yu, M. Zhihong, and S. M. M. Rahman, "Adaptive sliding mode approach for learning in a feedforward neural network," *Neural Computing & Applications*, Vol. 7, pp. 289-294, 1998.