# A STABILIZING AND ROBUSTIFYING TRAINING SCHEME FOR ARTIFICIAL NEURAL NETWORKS USED IN CONTROL OF COMPLEX SYSTEMS

**M. Onder Efe[1] and Okyay Kaynak[2]**

[1,2]Bogazici University, Electrical and Electronic Engineering Department
Bebek, 80815, Istanbul, Turkey
{efemond , kaynak}@boun.edu.tr

Abstract: This paper presents a method for stabilizing and robustifying the artificial neural networks trained by utilizing the gradient descent. The method proposed constructs a dynamic model of the conventional update mechanism and derives the stabilizing values of the learning rate. The stability in this context corresponds to the convergence in adjustable parameters of the neural network structure. It is shown that the selection of the learning rate as imposed by the proposed algorithm results in stable training in the sense of Lyapunov. Furthermore, the algorithm devised filters out the high frequency dynamics of the gradient descent method. The method analyzed in this paper integrates the gradient descent technique with variable structure systems methodology. In the simulations, control of a three degrees of freedom anthropoid robot is chosen for the evaluation of the performance.

Keywords: Sliding mode control, Backpropagation, Neural networks

## 1. INTRODUCTION

Artificial neural networks are well known with their property of representing complex nonlinear mappings. Earlier works on the mapping properties of these architectures have shown that neural networks are universal approximators (Hornik, 1989; Funahashi, 1989). Most of the studies reported adopt the Error Backpropagation (EBP) method, which is based on gradient descent, for tuning the parameters of the network structure. The primary drawback of the EBP technique is the lack of stabilizing forces ensuring the convergence.

As stated earlier, the issues of stability and robustness are of crucial importance from safety point of view. Because of this an implementation-oriented control engineering expert is always in pursuit of a design, which provide accurate tracking as well as insensitivity to environmental disturbances and structural uncertainties, which can be achieved by a suitable learning strategy. One way of studying the stability and robustness issues is the use of Variable Structure Systems (VSS) technique.

Variable Structure Control (VSC) has successfully been applied to a wide variety of systems having uncertainties in the representative system models. The philosophy of the control strategy is simple, being based on two goals. First, the system is forced towards a desired dynamics, which is a predefined subspace of the state-space, second, the system is maintained on that differential geometry. In the literature, the former dynamics is named the reaching mode, while the latter is called the sliding mode. This mode has useful invariance properties in the face of uncertainties in the system model and therefore a good candidate for tracking control of nonlinear systems. The control strategy borrows its name from the latter dynamic behavior, and is called *Sliding Mode Control (SMC)*.

Numerous contributions to VSS theory have been made during the last decade; some of them are as follows. Hung *et al.* (1993) has reviewed the control strategy for linear and nonlinear systems. In this reference, the switching schemes, putting the differential equations into canonical forms and generating simple SMC strategies are considered in detail. Gao *et al.* (1993) and Erbatur *et al.* (1999) consider the applications of SMC scheme for robotic manipulators and study and the quality of the scheme from the point of robustness. The performance of SMC scheme is proven to be satisfactory in the face of external disturbances and uncertainties in the system model representation. Kaynak *et al.* (1993) considers the design of discrete time SMC with particular emphasis on the system model uncertainties.

The objective of this paper is to develop a stabilizing training procedure for artificial neural networks. The procedure enforces the adjustable parameters to settle down to a steady state solution while meeting the design specifications. This is achieved through an appropriate combination of EBP and VSS techniques. The eventual form of the parameter update formula alleviates the handicaps of the gradient descent.

This paper is organized as follows. The second section briefly reviews the standard EBP technique, which is responsible for achieving the desired performance specifications. The parameter stabilizing part of the training methodology is derived in the third section. The section starts with a continuous time representation of the EBP algorithm and continues with an explanation of how the VSS based training criterion and EBP based training strategy are combined. The section gives the constraints on the design parameters. In the fourth section, the neural network structure with standard learning scheme is introduced and the application of the devised training strategy is presented. The fifth section introduces a plant, which is to be controlled by using the neural network architecture and the proposed learning algorithm. Simulation results are discussed in the sixth section and the conclusions are presented at the end of the paper.

## 2. STANDARD EBP TECHNIQUE

In most applications of artificial neural networks, EBP method constitutes the central part of the learning. In this section, the technique is briefly reviewed for systems in which the outputs are differentiable with respect to the parameter of interest. The method has first been formulated by Rumelhart *et al.* (1986). The approach has successfully been applied to a wide variety of optimization problems. The algorithm can be stated as follows. The observation error ($e$) in (1) is used to minimize the realization cost ($J_r$) in (2) by utilizing the rule described by (3), or more explicitly by (4), which is known as gradient descent or EBP in the related literature. Here $\phi$ is a generic parameter of the network structure and $\eta_\phi$ is the learning rate.

$$e_j = d_j - f_j(\phi, u) \tag{1}$$

$$J_r = \frac{1}{2}\sum_{j=1}^{outputs} e_j^2 \tag{2}$$

$$\Delta\phi = -\eta_\phi \frac{\partial J_r}{\partial \phi} \tag{3}$$

$$\Delta\phi = \eta_\phi \sum_{j=1}^{outputs} e_j \frac{\partial f_j(\phi, u)}{\partial \phi} = \eta_\phi N_\phi \tag{4}$$

The minimization proceeds recursively as given in (4). Since the update rule in (4) entails the observation error $e$, the algorithm is quite sensitive to the noisy observations, which directly influence the value of the adjustable parameter and degrade the learning performance. The next section presents the derivation of a method capable of reducing the adverse effects of noise thereby increasing the robustness in this sense.

## 3. STABILIZATION ALGORITHM

A continuous-time dynamic model of the parameter update rule prescribed by the EBP technique can be written as in (5).

$$\dot{\Delta\phi} = -\frac{1}{T_s}\Delta\phi + \frac{\eta_\phi}{T_s} N_\phi \tag{5}$$

The above model is composed of the sampling time denoted by $T_s$, the gradient based non-scaled parameter change denoted by $N_\phi$ and a scaling factor denoted by $\eta_\phi$, for the selection of which, a detailed analysis is presented in the subsequent discussion. Using Euler's first order approximation for the derivative term, one obtains the following relation, which validates the constructed model in (5).

$$\Delta\phi(k+1) = \eta_\phi N_\phi(k) \tag{6}$$

By comparing (4) and (6), the equivalency between the continuous and discrete forms of the update dynamics is thus clarified. The synthesis of the parameter stabilizing component is based on the integration of the system in (5) with VSS methodology. In the design of variable structure controllers, one method that can be followed is the reaching law approach (Gao, *et al.*, 1993). For the use of this theory in the stabilization of the training dynamics, define the switching function as in (7) and its dynamics as in (8).

$$s_\phi = \Delta\phi \tag{7}$$

$$\dot{s}_\phi = -\frac{Q_\phi}{T_s}\tanh\left(\frac{s_\phi}{\varepsilon}\right) - \frac{K_\phi}{T_s} s_\phi \tag{8}$$

where, $Q_\phi$ and $K_\phi$ are the gains, and $\varepsilon$ is the width of the boundary layer. In the derivations presented below, a key point is the fact that the system described by (5) is driven both by the learning rate $\eta_\phi$ and by the backpropagated error value $N_\phi$. Now it is demonstrated that some special selection of this quantity leads to a rule that minimizes the magnitude of parametric displacement. With the quantity defined in (9), equating (8) and (5) and solving for $\Delta\phi$ yields the relation in (10). The values of the $\eta_\phi$ imposed by (10) might be seen as the desired values at the first glance. However, this selection cancels out the backpropagated error value $N_\phi$ from (5), consequently the update dynamics exactly behaves as that defined by the adopted switching function (8), which does not necessarily minimize the cost in (2). Therefore the further analysis explores the restrictions on $\eta_\phi$ as well as the construction of the mixed training criterion.

$$A_\phi = Q_\phi \tanh\left(\frac{\Delta\phi}{\varepsilon}\right) + K_\phi \Delta\phi \tag{9}$$

$$\Delta\phi = \eta_\phi N_\phi + A_\phi \tag{10}$$

Now there is a model described by (5), and an equality to be enforced and formulated by (10). If one chooses a positive definite Lyapunov function as given in (11), the time derivative of this function must be negative definite for stability of parameter change ($\Delta\phi$) dynamics. Clearly the stability in the parameter change space implies the convergence in system parameters.

$$V_\phi = \frac{1}{2}s_\phi^2 = \frac{1}{2}(\Delta\phi)^2 \tag{11}$$

$$\dot{V}_\phi = (\Delta\phi)(\dot{\Delta\phi}) \tag{12}$$

If (5) and (10) are substituted into (12), the constraint stated in (13) is obtained for stability in the Lyapunov sense.

$$\left(\eta_\phi + \frac{1}{N_\phi}A_\phi\right)\left(\eta_\phi - \frac{1}{N_\phi}\Delta\phi\right) < 0 \tag{13}$$

Since $A_\phi$ and $\Delta\phi$ have the same signs, the roots of the expression in (13) clearly have opposite signs. The expression on the left-hand side assumes negative values between the roots. Therefore, in order to satisfy the inequality in (13), the learning rate must satisfy the constraint given in (14). In order to preserve the compatibility between the traditional gradient based approaches and the proposed approach, the interval of learning rate is restricted to positive values as described below. An appropriate selection of $\eta_\phi$ could be as in (15).

$$0 < \eta_\phi < \min\left\{\left|\frac{1}{N_\phi}\Delta\phi\right|, \left|-\frac{1}{N_\phi}A_\phi\right|\right\} \tag{14}$$

$$\eta_\phi = \beta \min\left\{\left|\frac{1}{N_\phi}\Delta\phi\right|, \left|-\frac{1}{N_\phi}A_\phi\right|\right\}, \quad 0 < \beta < 1 \tag{15}$$

By substituting the learning rate formulated in (15) into the equality given in (10), the stabilizing component $\Delta\phi_{VSS}$ of the parameter change formula is obtained as;

$$\Delta\phi_{VSS} = \beta \min\left\{|\Delta\phi|, |A_\phi|\right\}\operatorname{sgn}(N_\phi) + A_\phi \tag{16}$$

where, $\Delta\phi$ on the right hand side is the final update value yet to be obtained. The law introduced in (16) minimizes the cost of stability, which is the Lyapunov function defined by (11). The question now reduces to the following; can this law minimize the cost defined by (2)? The answer is obviously not, because the stabilizing component in (16) is derived from the displacement of the parameter vector denoted by $\Delta\phi$, whereas the minimization of (2) is achieved when $\phi$ tends to $\phi^*$ regardless of what the displacement is. In order to minimize (2), the parameter change anticipated by EBP technique, which is given in (17), should somehow be integrated into the final form of parameter update mechanism.

$$\Delta\phi_{EBP} = \zeta_\phi N_\phi \tag{17}$$

where, $\zeta_\phi$ is the learning rate. It is reasonable to expect that under certain constraints, a combination of the laws formulated in (16) and (17) in a weighted average will meet the objectives of both the parametric stabilization and the cost minimization, which means the fulfillment of the design specifications. The parameter update rule will then be as in (18).

$$\Delta\phi = \frac{\alpha_1\Delta\phi_{VSS} + \alpha_2\Delta\phi_{EBP}}{\alpha_1 + \alpha_2} \tag{18}$$

The parameter update formula given by (18) carries mixed displacement value containing both the parametric convergence, which is introduced by VSS part, and the cost minimization, which is due to the EBP technique. The balancing in this mixture is left to the designer by an appropriate selection of $\alpha_1$ and $\alpha_2$.

Lastly, the constraints for the global stability of the proposed training strategy are given. For this purpose, a Lyapunov function given in (19) is defined. In (19), $\gamma_\phi$ is a positive constant. The time derivative of the Lyapunov function is as given in (20).

$$V_\phi = \frac{1}{2}(\Delta\phi)^2 + \frac{\gamma_\phi}{2}(N_\phi)^2 \tag{19}$$

$$\dot{V}_\phi = \dot{\Delta\phi}\Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \tag{20}$$

In order to ensure the negativeness of the right hand side of (20), following three conditions must be satisfied.

$$\zeta_\phi < \frac{\alpha_1 + \alpha_2}{\alpha_2}\frac{(C_\phi - T_s)|\Delta\phi|}{|N_\phi|} \tag{21}$$

$$1 - \frac{2\alpha_1(\beta + Q_\phi + K_\phi)}{\alpha_1 + \alpha_2} > T_s \tag{22}$$

$$\beta(Q_\phi + K_\phi) < 1 \tag{23}$$

The details of the stability proof can be found in Efe *et al.* (2000).

## 4. TRAINING OF ARTIFICIAL NEURAL NETWORKS

In this section, application of the devised scheme to feedforward neural networks is presented. It is a well-known fact that the structure can effectively be used for identification and control purposes (Efe *et al.*, 2000; Efe *et al.*, 1999; Efe and Kaynak, 1999). In the conventional EBP technique, propagating the output error back through a feedforward neural network minimizes the cost function given in (2). The delta values for the neurons belonging to the output layer and the hidden layers are evaluated as given by (24) and (25) respectively.

$$\delta_j^{k+1} = (d_j - f_j)\,\Psi'(S_j^{k+1}) \tag{24}$$

$$\delta_j^{k+1} = \left( \sum_{h=1}^{neurons_{k+2}} \delta_h^{k+2} w_{jh}^{k+1} \right) \Psi'(S_j^{k+1}) \quad (25)$$

Having evaluated the delta values during the backward pass, the gradient based weight update rule described by (26) is applied for each training pair.

$$\Delta w_{ij\,EBP}^{k} = \zeta_{w_{ij}^k} \delta_j^{k+1} o_i^k \quad (26)$$

The VSS part of the proposed approach estimates the following update value for parametric stability.

$$\Delta w_{ij\,VSS}^{k} = \beta \min\left( \left|\Delta w_{ij}^k\right|, \left|A_{w_{ij}^k}\right| \right) \operatorname{sgn}\left( N_{w_{ij}^k} \right) + A_{w_{ij}^k} \quad (27)$$

The two update laws are then combined as a weighted average as before.

$$\Delta w_{ij}^k = \frac{\alpha_1 \Delta w_{ij\,VSS}^k + \alpha_2 \Delta w_{ij\,EBP}^k}{\alpha_1 + \alpha_2} \quad (28)$$

## 5. PLANT MODEL

In the simulations the dynamic model of a three degrees of freedom anthropoid robotic manipulator, which is illustrated in Fig. 1, is used as the test bed. Since the dynamics of such a mechatronic system is modeled by highly nonlinear and coupled differential equations, precise output tracking becomes a difficult objective. Therefore the methodology adopted must have the capability of coping with the stated difficulties.

The general form of the dynamics of a robotic manipulator is described by (29) where $M(q)$, $C(q,\dot{q})$, $g(q)$ and $u$ stand for the state varying inertia matrix, vector of coriolis and centrifugal terms, gravitational forces and applied torque inputs respectively. The nominal values of the plant parameters are given in Table 1 in standard units.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \quad (29)$$

Table 1. Manipulator parameters

| | | |
|---|---|---|
| Link 1 length | 0.50 | $l_1$ |
| Link 2 length | 0.40 | $l_2$ |
| Link 3 length | 0.40 | $l_3$ |
| Link 1 mass | 4.00 | $m_1$ |
| Link 2 mass | 3.00 | $m_2$ |
| Link 3 mass | 3.00 | $m_3$ |
| Distance link 1 CG-joint 1 | 0.20 | $l_{c1}$ |
| Distance link 2 CG-joint 2 | 0.20 | $l_{c2}$ |
| Cylindrical link radius | 0.05 | $R$ |
| i$^{th}$ cylindrical link inertial parameter | $E_1=m_1R^2/2$, $E_i=m_il_i^2/12$ | $E_i$ |
| i$^{th}$ cylindrical link inertial parameter | $A_i=m_iR^2/2$ | $A_i$ |
| i$^{th}$ cylindrical link inertial parameter | $I_i=m_il_i^2/12$ for i=2,3 | $I_I$ |
| Link 1 torque limits | ±50.00 | $\tau_{sat\,1}$ |
| Link 2 torque limits | ±40.00 | $\tau_{sat\,2}$ |
| Link 3 torque limits | ±20.00 | $\tau_{sat\,3}$ |

If the angular positions and angular velocities are described as the state variables of the system, six coupled and first order differential equations can define the model. In (30) through (33), the nonzero entries of the state varying inertia matrix are described. The nonzero Cristoffel symbols are given in (34) through (37). The details of the plant model are presented by Stadler (1995).

$$\begin{aligned} M_{11} &= m_2 l_{c2}^2 \cos^2(q_2) + \\ & m_3\left(l_2\cos(q_2) + l_{c3}\cos(q_2+q_3)\right)^2 + E_1 + \\ & A_2\sin^2(q_2) + E_2\cos^2(q_2) + \\ & A_3\sin^2(q_2+q_3) + E_3\cos^2(q_2+q_3) \end{aligned} \quad (30)$$

$$\begin{aligned} M_{22} &= m_2 l_{c2}^2 \sin^2(q_2) + \\ & m_3\left(l_2^2 + l_{c3}^2 + 2l_2l_{c3}\cos(q_3)\right) + I_2 + I_3 \end{aligned} \quad (31)$$

$$M_{23} = M_{32} = m_3\left(l_{c3}^2 + l_{c3}l_2\cos(q_3)\right) + I_3 \quad (32)$$

$$M_{33} = m_3 l_{c3} + I_3 \quad (33)$$

$$\begin{aligned} hc_1 &= \left(-m_2l_{c2}^2 + A_2 - E_2\right)\cos(q_2)\sin(q_2) + \\ & (A_3 - E_3)\cos(q_2+q_3)\sin(q_2+q_3) + \\ & m_3\left(l_2\cos(q_2) + l_{c3}\cos(q_2+q_3)\right)* \\ & \left(-l_2\sin(q_2) - l_{c3}\sin(q_2+q_3)\right) \end{aligned} \quad (34)$$

$$\begin{aligned} hc_2 &= \sin(q_2+q_3)* \\ & \left(-m_3l_{c3}l_2\cos(q_2) + (-m_3l_{c3}^2 + A_3 - E_3)\cos(q_2+q_3)\right) \end{aligned} \quad (35)$$

$$hc_3 = m_2 l_{c2}^2 \cos(q_2)\sin(q_2) \quad (36)$$

$$hc_4 = -m_2 l_2 l_{c3}\sin(q_3) \quad (37)$$

Coriolis, centrifugal terms and gravity terms are formulated as follows, where $G$ represents the gravity constant.

$$C(q,\dot{q}) = \begin{bmatrix} 2hc_1\dot{q}_1\dot{q}_2 + 2hc_2\dot{q}_1\dot{q}_3 \\ -hc_1\dot{q}_1^2 + 2hc_4(\dot{q}_2\dot{q}_3 + \dot{q}_3^2) + hc_3\dot{q}_2^2 \\ -hc_2\dot{q}_1^2 - hc_4\dot{q}_2^2 \end{bmatrix} \quad (38)$$

$$g(q_1,q_2,q_3) = \begin{bmatrix} 0 \\ \begin{pmatrix} (m_2l_{c2} + m_3l_2)G\cos(q_2) \\ + m_3l_{c3}G\cos(q_2+q_3) \end{pmatrix} \\ m_3l_{c3}G\cos(q_2+q_3) \end{bmatrix} \quad (39)$$

## 6. SIMULATION RESULTS

In the simulations, the plant introduced in the Sec. 5 is controlled by the neural network structure considered in Sec. 4. The architecture of the control system is an ordinary feedback loop, in which the neural controller has one hidden layer being comprised of neurons having hyperbolic tangent type neuronal activation functions. The output layer neurons have linear activation functions.

During the simulations, all weights and biases of the neural network have been adjusted. The initial values of the parameters of the neural network have been set such that the initial control surfaces for all three links approximately resemble to that of a Proportional plus Derivative (PD) controller having the following parameter set.

$$\begin{bmatrix} K_{p\,BASE} & K_{d\,BASE} \\ K_{p\,SHOULDER} & K_{d\,SHOULDER} \\ K_{p\,ELBOW} & K_{d\,ELBOW} \end{bmatrix} = \begin{bmatrix} 40 & 40 \\ 180 & 260 \\ 150 & 70 \end{bmatrix} \quad (40)$$

The reference angular position and velocity profiles used in all simulations are depicted in Fig. 2. The simulations are started with initial rest conditions.

Apart from the dynamic complexity of the system under control, a considerable difficulty to be alleviated by the algorithm discussed is the existence of the observation noise. It is assumed that the encoders provide noisy measurements to the controller. The noise sequence is Gaussian distributed and has the same statistical properties for all six state variables, namely, each sequence has zero mean and variance equal to 33e-4. It is expected that the stabilizing forces created on the adjustable design parameters will lead to the elimination of the adverse effects of the observation noise. Thus the results obtained will enable the designer to make a fair comparison between the pure EBP technique and the proposed combination especially in the sense of rejecting the high frequency components exciting the training dynamics.

In the training of the controller structure discussed in the paper, the squared sum of parametric changes is defined to be the cost of stability, which runs over all adjustable weights and biases of the neurocontroller.

$$J_s(t) = \sum_{\phi} (\Delta\phi(t))^2 \quad (41)$$

For the use of the proposed algorithm, $\alpha_1$ is set to 2 while $\alpha_2$ is equal to unity. The state tracking errors are depicted in Fig. 3. It is evident from the figure that the proposed combination results in precise state tracking under the existence of environmental and structural difficulties stated above. With the same initial conditions, if $\alpha_1$ is set to zero, which disables the VSS part of the algorithm, a divergent behavior is observed in the state tracking errors, which are depicted in Fig. 4.

The behavior of the total parametric cost described by (41) is depicted in Fig. 5. The left subplot of Fig. 5 indicates that the cost in (41) reaches to very small values during the early phases of the simulation. This is due to the parameter stabilizing property of the approach discussed. It must be stressed what the EBP method can achieve at best is the marginal stability in the parameter change space. This characteristic of the standard technique makes it highly sensitive to the environmental disturbances. In the simulations discussed, the existence of noise makes this aspect of EBP technique more visible. Clearly, the presence of observation noise and the requirements of the problem in hand stimulate the unstable internal dynamics of EBP method. This is apparent from the right subplot of Fig. 5 that the average magnitude is increasing in time.

The simulation settings are tabulated in Table 2, in which it is apparent that the constraints stated in (22) and (23) are satisfied.

Table 2. Simulation parameters

| | |
|---|---|
| $T_s$ | 1.0 msec. |
| $\beta$ | 0.1 |
| $\alpha_{1,2}$ | See text |
| $Q$ | 0.1 |
| $K$ | 0.1 |
| $\varepsilon$ | 1.0 |
| *#Input Neurons* | 6 |
| *#Hidden Neurons* | 12 |
| *#Output Neurons* | 3 |

## 7. CONCLUSIONS

One of the major problems in applications of gradient based training strategies is the lack of stabilizing forces to prevent the adjustable parameters to grow unboundedly. This aspect of training without safety conditions constitutes a barrier between the theoretical developments and industrial applications, whose prime concern is stability and robustness. The application examples utilizing the gradient information in training have therefore used the architectures of artificial neural networks, which are typically trained off-line with a priori data. In this paper, a method for creating stabilizing forces on the training dynamics is proposed. The results stipulate that the proposed approach fulfills the task to be accomplished much better that that can be observed with ordinary EBP technique. The comparison strongly recommends the use of the algorithm for the applications requiring on-line tuning of the parameters, stability in the parameter change space and insensitivity to environmental disturbances.

## 8. ACKNOWLEDGMENTS

This work is supported by Bogazici University Research Fund (Project no: 99A202)

## 9 REFERENCES

Efe, M. O., O. Kaynak and B. M. Wilamowski (2000). Stable Training of Computationally Intelligent Systems by Using Variable Structure Systems Technique, *IEEE Trans. on Industrial Electronics*, **47**, 2.

Efe, M. O., E. Abadoglu and O. Kaynak (1999). Analysis and Design of a Neural Network Assisted Nonlinear Controller for a Bioreactor, *Int. Journal of Robust and Nonlinear Control*, **9**, 799-815.

Efe, M. O., and O. Kaynak (1999). A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems, Mechatronics, **9**, 3, 287-300.

Erbatur, K., O. Kaynak and A. Sabanovic (1999). A Study on Robustness Property of Sliding-Mode Controllers: A Novel Design and Experimental Investigations, *IEEE Trans. on Industrial Electronics*, **46**, 5, 1-7.

Funahashi, K. (1989). On the Approximate Realization of Continuous Mappings by Neural Networks, *Neural Networks*, **2**, 183-192.

Gao, W., J. C. Hung (1993). Variable Structure Control of Nonlinear Systems: A New Approach, *IEEE Trans. on Industrial Electronics*, **40**, 1, 45-55.

Hornik, K. (1989). Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, **2**, 359-366.

Hung, J. Y., W. Gao and J. C. Hung (1993). Variable Structure Control: A Survey, *IEEE Trans. on Industrial Electronics*, **40**, 1, 2-22.

Kaynak, O., and A. Denker, (1993). Discrete-Time Sliding Mode Control in the Presence of System Uncertainty, *Int. Journal of Control*, **57**, 5, 1177-1189.

Rumelhart, D. E., G. E. Hinton and R. J. Williams (1986). Learning Internal Representations by Error Propagation," (D. E. Rumelhart and J. L. McClelland (Ed)), *Parallel Distributed Processing*, **1**, 318-362, MIT Press, Cambridge, M.A.

Stadler, W. (1995). Analytical Robotics and Mechatronics, McGraw Hill, 476.

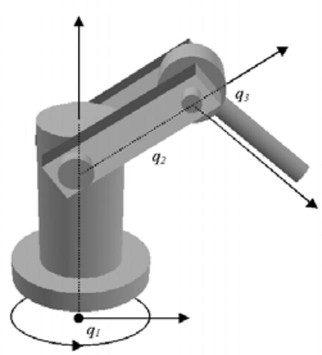


Fig. 1. Physical structure of the plant

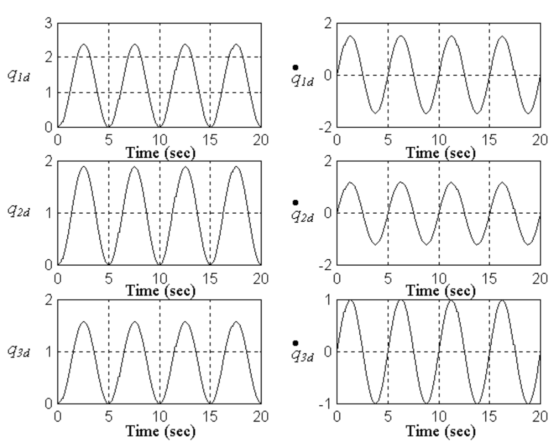


Fig. 2. The reference angular position and velocity profiles

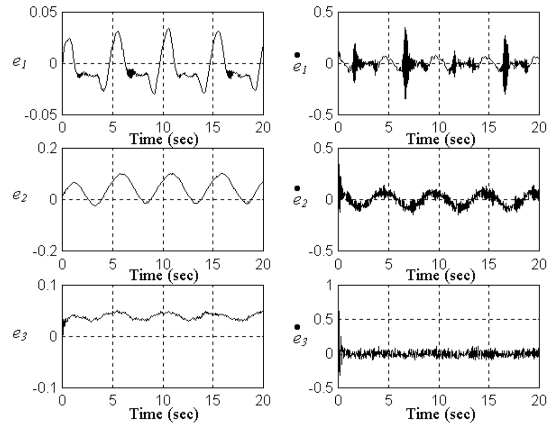


Fig. 3. State tracking errors observed with the proposed technique

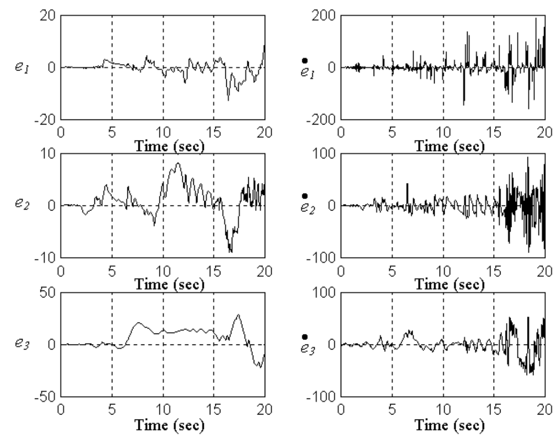


Fig. 4. State tracking errors observed with pure EBP technique

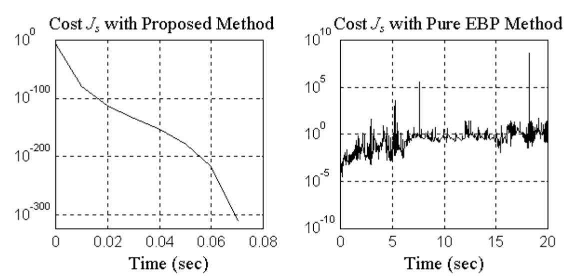


Fig. 5. Time behavior of the parametric cost