Stabilizing and Robustifying the Learning Mechanisms of Artificial Neural Networks in Control Engineering Applications

M. Onder Efe,* Okyay Kaynak†

Electrical and Electronic Engineering Department, Bebek, 80815,
Istanbul, Turkey

This paper discusses the stabilizability of artificial neural networks trained by utilizing the gradient information. 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 excitation of this dynamics typically occurs in the presence of noise and abruptly changing the parameters of the mapping being learned. This adversely influences the learning performance that can be attained during a training cycle. A natural consequence following this excitation is divergence in parameter space. The method analyzed in this paper integrates the gradient descent technique with variable structure systems methodology, which is well known for its robustness to environmental disturbances. In the simulations, control of a three degrees of freedom anthropoid robot is chosen for the evaluation of the performance. For this purpose, a feedforward neural network structure is utilized as the controller. Highly nonlinear dynamics of the plant, existence of a considerable amount of observation noise, and the adverse effects of gravitational forces constitute the difficulties to be alleviated by the neurocontroller trained with the proposed method. In order to come up with a fair comparison, the results obtained with the pure gradient descent technique with the same initial conditions are also presented and discussed. © 2000 John Wiley & Sons, Inc.

1. INTRODUCTION

One of the major problems in the training of artificial neural networks (ANN) is the lack of stabilizing forces, the existence of which prevents the unbounded growth in the adjustable parameters. This fact is intimately related to the analytic explanation of the internal dynamics of the training strategy,

*Author to whom correspondence should be addressed; e-mail: efemond@boun.edu.tr

†e-mail: kaynak@boun.edu.tr

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 15, 365–388 (2000)

which typically concern several tens of variables even for the simplest structures. Another major problem of the training phase is the robustness, i.e., how well the ANN structure, which is trained on-line or off-line, will perform under the existence of strong external disturbances. The methods overcoming the mentioned problems aim to maintain a desired behavior in the face of factors adversely influencing the performance and applicability. Therefore, the concept of stability and robustness constitutes a prime requirement of the design methodologies, particularly in systems and control engineering practice. Strictly speaking, a method violating the stability requirements is a potential danger from a safety point of view. However, the innovations in data mining, data fusion, sensor technology, recognition technology, and fast microprocessors are ever increasingly encouraging the use of ANN structures, whose operating philosophy is suitable to interaction of an *Expert* and *Machine*.

Artificial neural networks are well known for their property of representing complex nonlinear mappings. Earlier works on the mapping properties of these architectures have shown that neural networks are universal approximators.[1,2] The mathematical power of intelligence is commonly attributed to the neural systems because of their massively interconnected and fault tolerant architecture. Various architectures of neural systems are studied in the literature. Feedforward and recurrent neural networks, Gaussian radial basis function neural networks, dynamical neural networks,[3] and Runge–Kutta neural networks[4,5] constitute typical structurally different models. The approaches mentioned have been widely used in applications extending from speech recognition and optoelectronics to identification and control of nonlinear systems.[3–12] Most of the studies reported in the literature adopt the error backpropagation (EBP) method for tuning the parameters of the network structure.[3–16] Since the discovery of the algorithm in 1986 by Rumelhart et al.,[17] the EBP method has become the standard method for training of artificial neural networks. A part of the later studies concentrated on the enhancement of the optimization performance of the EBP technique.[18–21] In this respect, stabilization and robustification of the EBP strategy is a remarkable progression for such a widely used optimization technique.

As stated earlier, issues of stability and robustness are of crucial importance from safety and performance points of view. Because of this, an implementation-oriented control engineering expert is always in pursuit of a design which provides accurate tracking as well as insensitivity to environmental disturbances and structural uncertainties. At this point, it must be emphasized that these ambiguities can never be modeled accurately. When the designer tries to minimize the ambiguities by the use of a detailed model, the design becomes so tedious that its cost increases dramatically. A suitable way of tackling uncertainties without the use of complicated models is to introduce variable structure systems (VSS) theory based components into the design procedure.

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 toward 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 is 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).

The earliest notion of SMC strategy was constructed on a second order system in the late 1960s by Emelyanov.[22] The work stipulated that a special line could be defined on the phase plane, such that any initial state vector can be driven toward the plane and then be maintained on it, while forcing the error dynamics toward the origin. Since then, the theory has been greatly improved and the sliding line has taken the form of a multidimensional surface, called the *sliding surface*; the function defining it is called the *switching function*.

Numerous contribution to VSS theory have been made during the last decade; some of them are as follows. Hung et al.[23] has reviewed the control strategy for linear and nonlinear systems. In Ref. 23, the switching schemes, putting the differential equations into canonical forms and generating simple SMC strategies, are considered in detail. In Refs. 24 and 25, applications of the SMC scheme to robotic manipulators are studied and the quality of the scheme is discussed from the point of robustness. One of the crucial points in SMC is the selection of the parameters of the sliding surface. Some studies devoted to the adaptive design of sliding surfaces have shown that the performance of the control system can be refined by interfacing it with an adaptation mechanism, which regularly redesigns the sliding surface.[26–28] This eventually results in a robust control system. The performance of the SMC scheme is proven to be satisfactory in the face of external disturbances and uncertainties in the system model representation. Another systematic examination of the SMC approach is presented in Ref. 29. In this reference, the practical aspects of SMC design are assessed for both continuous-time and discrete-time cases, and a special consideration is given to the finite switching frequency, limited bandwidth actuators, and parasitic dynamics. In Ref. 30, the design of discrete-time SMC is presented with particular emphasis on the system model uncertainties. Some studies consider the robustness property of the VSS technique by equipping the system with computationally intelligent methods. In Refs. 31 and 32, fuzzy inference systems are proposed for the SMC scheme. A standard fuzzy system is studied and the relevant robustness analyses are carried out. Particularly, the work presented in Ref. 32 emphasizes that the robustness and stability properties of fuzzy control strategies can be analyzed through the use of SMC theory. It is shown in this reference that the approach is robust, i.e., it can compensate the deficiencies caused by poor modeling of plant dynamics and exogenous disturbances.

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 the error backpropagation (EBP) algorithm[17] with VSS philosophy. The early applications of VSS theory in training of ANN have considered the adjustment of the parameters of simple models such as adaptive linear elements (ADALINE).[33] The method presented in Ref. 33 also presents the forward and inverse dynamics identification of a Kapitsa pendulum. The fundamental difference of the algorithm presented in this paper is the fact that the derivation is based on the mixture of two different update values. Furthermore, the eventual form of the parameter update formula alleviates the handicaps of the gradient based training algorithms.

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. In the fourth section, analysis of the imposed dynamics is presented. In this section, it is shown that the desired dynamics and imposed dynamics are stable but structurally different. The fifth section gives the global stability proof of the mixed training strategy and discusses the constraints on the design parameters. In the sixth section, the feedforward neural network structure with standard learning scheme is introduced and the application of the devised training strategy is presented. The seventh section introduces a plant, which is to be controlled by using the feedforward neural network architecture and the proposed learning algorithm. Simulation results are discussed in the eighth section and the conclusions are presented at the end of the paper.

2. STANDARD ERROR BACKPROPAGATION TECHNIQUE

In most applications of artificial neural networks, the EBP method constitutes the central part of the learning.[3–16,18–20] 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 was first formulated by Rumelhart et al.[17] in 1986. The approach has successfully been applied to a wide variety of optimization problems. Using the nomenclature given in the appendix, the algorithm can be stated as follows:

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

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

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

The observation error in (1) is used to minimize the realization cost in (2) by utilizing the rule described by (3), which is known as gradient descent or EBP in the related literature:

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

The minimization proceeds recursively as given in (4), for which the sensitivity derivative with respect to the generic parameter ϕ is needed. Since the update rule in (4) entails the observation error e, the algorithm is quite sensitive to the noise 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 OF TRAINING DYNAMICS BY VARIABLE STRUCTURE SYSTEMS APPROACH

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 nonscaled parameter change denoted by $N_\phi = \sum_{j=1}^{\text{outputs}} e_j(\partial f_j(\phi, u)/\partial\phi)$, and a scaling factor denoted by η_ϕ, 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 obviously validates the constructed model in (5) and which leads to the representation in (7):

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

$$\Delta\phi(k+1) = \eta_\phi N_\phi(k) \tag{7}$$

By comparing (4) and (7), 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 variable structure systems methodology. In the design of variable structure controllers, one method that can be followed is the reaching law approach.[23] For the use of this theory in the stabilization of the training dynamics, let us define the switching function as in (8) and its dynamics as in (9):

$$s_\phi = \Delta\phi \tag{8}$$

$$\dot{s}_\phi = -\frac{Q_\phi}{T_s}\tanh\left(\frac{s_\phi}{\varepsilon}\right) - \frac{K_\phi}{T_s}s_\phi \tag{9}$$

where Q_ϕ and K_ϕ are the gains, and ε 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 η_ϕ and the the backpropagated error value N_ϕ. Now we demonstrate that some special selection of this quantity leads to a rule that minimizes the magnitude of parametric displacement. With the quantity defined in (10), equating (9) and (5) and solving for $\Delta\phi$ yields the relation in (11):

$$A_\phi = Q_\phi \tanh\left(\frac{\Delta\phi}{\varepsilon}\right) + K_\phi \, \Delta\phi \tag{10}$$

$$\Delta\phi = \eta_\phi N_\phi + A_\phi \tag{11}$$

The values of the η_ϕ imposed by (11) might be seen as the desired values at the first glance. However, this selection cancels out the backpropagated error value N_ϕ from (5); consequently, the update dynamics exactly behaves as that defined by the adopted switching function (9), which does not necessarily minimize the cost in (2). Therefore, the further analysis explores the restrictions on η_ϕ as well as the construction of the mixed training criterion.

Now we have a model described by (5), and an equality to be enforced and formulated by (11). If one chooses a positive definite Lyapunov function as given in (12), the time derivative of this function must be negative definite for stability of parameters change ($\Delta\phi$) dynamics. Clearly, the stability in parameter change space implies the convergence in system parameters.

$$V_\phi = \tfrac{1}{2} s_\phi^2 = \tfrac{1}{2}(\Delta\phi)^2 \tag{12}$$

$$\dot{V}_\phi = (\Delta\phi)\left(\Delta\dot{\phi}\right) \tag{13}$$

If (5) and (11) are substituted into (13), the constraint stated in (14) is obtained for stability in the Lyapunov sense:

$$\eta_\phi^2 + \frac{1}{N_\phi}(A_\phi - \Delta\phi)\eta_\phi - \frac{1}{N_\phi^2} A_\phi \, \Delta\phi < 0 \tag{14}$$

Equation (14) can be rewritten in a more tractable form as follows:

$$\left(\eta_\phi + \frac{1}{N_\phi} A_\phi\right)\left(\eta_\phi - \frac{1}{N_\phi}\,\Delta\phi\right) < 0 \tag{15}$$

Since A_ϕ and $\Delta\phi$ have the same signs, the roots of the expression in (15) 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 (15), the learning rate must satisfy the constraint given in (16):

$$0 < \eta_\phi < \min\left\{\left|\frac{1}{N_\phi}\Delta\phi\right|, \left|-\frac{1}{N_\phi}A_\phi\right|\right\} \tag{16}$$

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 above. An appropriate selection η_ϕ could be as follows:

$$\eta_\phi = \beta \min\left\{\left|\frac{1}{N_\phi}\Delta\phi\right|, \left|-\frac{1}{N_\phi}A_\phi\right|\right\}, \qquad 0 < \beta < 1 \tag{17}$$

By substituting the learning rate formulated in (17) into the equality given in (11), the stabilizing component $\Delta\phi_{\mathrm{VSS}}$ of the parameter change formula is obtained as

$$\Delta\phi_{\mathrm{VSS}} = \beta \min\{|\Delta\phi|, |A_\phi|\}\operatorname{sgn}(N_\phi) + A_\phi \tag{18}$$

where $\Delta\phi$ on the right-hand side is the final update value yet to be obtained. The law introduced in (18) minimizes the cost of stability, which is the Lyapunov function defined by (12). 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 (18) is derived from the displacement of the parameter vector denoted by $\Delta\phi$, whereas the minimization of (2) is achieved when ϕ tends to $\dot{\phi}$ regardless of what the displacement is. In order to minimize (2), the parameter change anticipated by the EBP technique, which is reviewed in the second section, should somehow be integrated into the final form of parameter update mechanism. As introduced in the second section, the EBP algorithm evaluates a parameter change as given in (19):

$$\Delta\phi_{\mathrm{EBP}} = \zeta_\phi N_\phi \tag{19}$$

where ζ_ϕ is the learning rate. It is reasonable to expect that under certain constraints, a combination of the laws formulated in (18) and (19) 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 (20):

$$\Delta\phi = \frac{\alpha_1\,\Delta\phi_{\mathrm{VSS}} + \alpha_2\,\Delta\phi_{\mathrm{EBP}}}{\alpha_1 + \alpha_2} \tag{20}$$

The parameter update formula given by (20) carries mixed displacement value containing both the parametric convergence, which is introduced by the 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 α_1 and α_2.

4. ANALYSIS OF THE IMPOSED DYNAMICS

In the previous section, the mixed training algorithm is derived. This section analyzes the implications of the learning rate in (17) on the domain of parametric change space. If the learning rate in (17) is substituted into the dynamic model of (5), one ends up with the dynamics formulated in (21), which characterizes the behavior of the system driven solely by the learning rate in (17):

$$\dot{\Delta\phi} = -\frac{1}{T_s}\Delta\phi + \frac{\beta}{T_s}\min\{|\Delta\phi|, |A_\phi|\}\mathrm{sgn}(N_\phi) \tag{21}$$

In (21), two different cases can be of interest, namely, $|\Delta\phi| < |A_\phi|$ or $|A_\phi| < |\Delta\phi|$. In the analysis presented below, the following two facts must be kept in mind.

Fact 1: $|x| = x\,\mathrm{sgn}(x)$, where $x \in \Re$.

Fact 2: $\mathrm{sgn}(x_1)\mathrm{sgn}(x_2) \leq 1$, where $x_1, x_2 \in \Re$.

For the first case, (21) becomes

$$\dot{\Delta\phi} = \frac{-1 + \beta\,\mathrm{sgn}(N_\phi)\mathrm{sgn}(\Delta\phi)}{T_s}\Delta\phi \leq \frac{-1+\beta}{T_s}\Delta\phi \tag{22}$$

Since $\beta < 1$, the imposed dynamics is globally stable. For the second case, (21) turns out to be as follows with the aid of Fact 3.

Fact 3: $A_\phi = Q_\phi \tanh(\Delta\phi/\varepsilon) + K_\phi\,\Delta\phi < (Q_\phi + K_\phi)\,\Delta\phi$, where $0 < \varepsilon \leq 1$, which is the admissible interval for the boundary layer width. The lower bound on ε is due to the physical meaning, whereas the upper bound is due to the inequality given above:

$$\begin{aligned} \dot{\Delta\phi} &= -\frac{1}{T_s}\Delta\phi + \frac{\beta}{T_s}A_\phi\,\mathrm{sgn}(N_\phi)\mathrm{sgn}(A_\phi) \\ &\leq \frac{-1+\beta(Q_\phi + K_\phi)}{T_s}\Delta\phi \end{aligned} \tag{23}$$

If the term $\beta(Q_\phi + K_\phi)$ is constrained to be less than unity, the imposed dynamics becomes globally stable. The analysis presented in this section reveals that the imposed dynamics is somewhat different from the adopted switching function because of the constraints on the learning rate (η_ϕ) selection; nevertheless, the imposed dynamics is globally stable. In the next section, the overall stability proof of the training algorithm is discussed.

5. EXTRACTING THE CONDITIONS FOR THE GLOBAL STABILITY OF THE MIXED TRAINING DYNAMICS

In this section, the global stability of the mixed training strategy is analyzed. For this purpose, a Lyapunov function given in (24) is defined. In (24), γ_ϕ is a positive constant and its properties are discussed at the end of the section:

$$V_\phi = \frac{1}{2}(\Delta\phi)^2 + \frac{\gamma_\phi}{2}(N_\phi)^2 \tag{24}$$

The time derivative of the Lyapunov function is as given in (25):

$$\dot{V}_\phi = \Delta\dot{\phi}\,\Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \tag{25}$$

Since the analysis in this section concerns the stability of the mixed training strategy, the combined form of the learning algorithm, as given below, should be used in the formulation:

$$\dot{V}_\phi = \left(-\frac{1}{T_s}\Delta\phi + \frac{\alpha_1\,\Delta\phi_{\mathrm{VSS}} + \alpha_2\,\Delta\phi_{\mathrm{EBP}}}{(\alpha_1+\alpha_2)T_s} \right)\Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \tag{26}$$

If the $\Delta\phi_{\mathrm{VSS}}$ of (18) and the $\Delta\phi_{\mathrm{EBP}}$ of (19) are substituted into (26), one obtains the following relation, which can assume two different forms due to the minimum operator:

$$\begin{aligned}\dot{V}_\phi = {} & -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}\min\{|\Delta\phi|, |A_\phi|\}\mathrm{sgn}(N_\phi)\,\Delta\phi \\ & + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}A_\phi\,\Delta\phi + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi\end{aligned} \tag{27}$$

Case 1: $|\Delta\phi| < |A_\phi|$.

Since $|\Delta\phi| = \Delta\phi\,\mathrm{sgn}(\Delta\phi)$, (27) can be rewritten as follows:

$$\begin{aligned}\dot{V}_\phi = {} & -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}\Delta\phi^2\,\mathrm{sgn}(\Delta\phi)\mathrm{sgn}(N_\phi) \\ & + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}A_\phi\,\Delta\phi + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi\end{aligned} \tag{28}$$

$$\begin{aligned} = {} & -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}\Delta\phi^2\,\mathrm{sgn}(\Delta\phi)\mathrm{sgn}(N_\phi) \\ & + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}Q_\phi\tanh(\Delta\phi)\,\Delta\phi + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}K_\phi\,\Delta\phi^2 \\ & + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi\end{aligned} \tag{29}$$

Due to Fact 3 of Section 4, the equality in (29) satisfies the following inequality:

$$
\begin{aligned}
\dot{V}_\phi &< -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}\Delta\phi^2 \operatorname{sgn}(\Delta\phi)\operatorname{sgn}(N_\phi) + \frac{\alpha_1 Q_\phi}{(\alpha_1+\alpha_2)/T_s}\Delta\phi^2 \\
&\quad + \frac{\alpha_1 K_\phi}{(\alpha_1+\alpha_2)T_s}\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi \Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \qquad (30) \\
&\le \left(\frac{(\beta + Q_\phi + K_\phi)\alpha_1}{(\alpha_1+\alpha_2)T_s} - \frac{1}{T_s} \right)\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi \Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \qquad (31)
\end{aligned}
$$

If the right-hand side of (30) is rearranged with Fact 2 of Section 4, (31) is obtained. The inequality in (31) constitutes a time varying quantity, which is always larger than the quantity in (28). The further analysis for this case will proceed together with the result of the second case.

Case 2: $|A_\phi| < |\Delta\phi|$.

$$
\begin{aligned}
\dot{V}_\phi &= -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}|A_\phi|\operatorname{sgn}(N_\phi)\Delta\phi + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}A_\phi\Delta\phi \\
&\quad + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi \Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \qquad (32) \\
&= -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}\operatorname{sgn}(A_\phi)\operatorname{sgn}(N_\phi)A_\phi\Delta\phi \\
&\quad + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}A_\phi\Delta\phi + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi \Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \qquad (33) \\
&= -\frac{1}{T_s}\Delta\phi^2 + \left(\frac{\alpha_1\beta}{(\alpha_1+\alpha_2)T_s}\operatorname{sgn}(A_\phi)\operatorname{sgn}(N_\phi) + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s} \right) A_\phi\Delta\phi \\
&\quad + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi \Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \qquad (34) \\
&= -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}\big(\beta\operatorname{sgn}(A_\phi)\operatorname{sgn}(N_\phi) + 1\big)A_\phi\Delta\phi \\
&\quad + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi \Delta\phi + \gamma_\phi \dot{N}_\phi N_\phi \qquad (35)
\end{aligned}
$$

Fact 1: $A_\phi\Delta\phi < (Q_\phi + K_\phi)\Delta\phi^2$ and $A_\phi\Delta\phi > 0$.

Fact 2: $\min(\beta\,\mathrm{sgn}(A_\phi)\mathrm{sgn}(N_\phi)+1) = 1-\beta$ and $\max(\beta\,\mathrm{sgn}(A_\phi)\mathrm{sgn}(N_\phi)+1) = 1+\beta$ and $0<\beta<1$.

Due to the facts given above, the following rearrangements can be made:

$$\dot{V}_\phi < -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}\big(\beta\,\mathrm{sgn}(A_\phi)\mathrm{sgn}(N_\phi)+1\big)(Q_\phi+K_\phi)\,\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi\dot{N}_\phi N_\phi \tag{36}$$

$$\le -\frac{1}{T_s}\Delta\phi^2 + \frac{\alpha_1}{(\alpha_1+\alpha_2)T_s}(1+\beta)(Q_\phi+K_\phi)\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi\dot{N}_\phi N_\phi \tag{37}$$

$$< -\frac{1}{T_s}\Delta\phi^2 + \frac{2\alpha_1}{(\alpha_1+\alpha_2)T_s}(Q_\phi+K_\phi)\,\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi\dot{N}_\phi N_\phi \tag{38}$$

$$= \left(\frac{2\alpha_1(Q_\phi+K_\phi)}{(\alpha_1+\alpha_2)T_s} - \frac{1}{T_s}\right)\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi\dot{N}_\phi N_\phi \tag{39}$$

$$< \left(\frac{2\alpha_1(\beta+Q_\phi+K_\phi)}{(\alpha_1+\alpha_2)T_s} - \frac{1}{T_s}\right)\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi\dot{N}_\phi N_\phi \tag{40}$$

If the negativity of the quantity on the right-hand side of the inequality (40) is ensured, the negativity of the quantity in (31) becomes trivial. Therefore, the two inequalities can be reduced to one inequality, which is given below. The global stability of the mixed training dynamics will clearly require the negativity of the quantity on the right-hand side of (41):

$$\dot{V}_\phi < -\frac{C_\phi}{T_s}\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1+\alpha_2)T_s}\zeta_\phi N_\phi\,\Delta\phi + \gamma_\phi\dot{N}_\phi N_\phi \tag{41}$$

where

$$C_\phi = 1 - \frac{2\alpha_1(\beta+Q_\phi+K_\phi)}{\alpha_1+\alpha_2} \tag{42}$$

Set

$$\gamma_\phi = \frac{\sigma_\phi^2}{\sup\limits_t \left| \dot{N}_\phi N_\phi \right|} \tag{43}$$

where σ_ϕ^2 is the least nonzero value of $\Delta\phi^2$ observed during a training course. It should be noted here that one may not know the numerical value of this number, but there exists such a number in the course of each training trial. With this value of γ_ϕ, (41) becomes

$$\dot{V}_\phi < -\frac{C_\phi}{T_s}\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1 + \alpha_2)T_s}\zeta_\phi N_\phi \,\Delta\phi + \frac{\sigma_\phi^2}{\sup\limits_t \left| \dot{N}_\phi N_\phi \right|}\dot{N}_\phi N_\phi \tag{44}$$

$$< -\frac{C_\phi}{T_s}\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1 + \alpha_2)T_s}\zeta_\phi N_\phi \,\Delta\phi + \sigma_\phi^2 = B_\phi \tag{45}$$

The inequality in (45) follows from the inequality given below:

$$\frac{\dot{N}_\phi N_\phi}{\underset{t\quad t}{\sup} \left| \dot{N}_\phi N_\phi \right|} < 1 \tag{46}$$

Since $\sigma_\phi^2 \le \Delta\phi^2$ for all $t \ge 0$,

$$B_\phi \le -\frac{C_\phi}{T_s}\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1 + \alpha_2)T_s}\zeta_\phi N_\phi \,\Delta\phi + \Delta\phi^2 \tag{47}$$

$$< -\left(\frac{C_\phi}{T_s} - 1\right)\Delta\phi^2 + \frac{\alpha_2}{(\alpha_1 + \alpha_2)T_s}\zeta_\phi |N_\phi|\,|\Delta\phi| \tag{48}$$

In order to ensure the negativeness of the right-hand side of (48), the following inequality must be satisfied:

$$\zeta_\phi < \frac{\alpha_1 + \alpha_2}{\alpha_2}\frac{(C_\phi - T_s)|\Delta\phi|}{|N_\phi|} \tag{49}$$

This selection of the learning rate for the EBP part ensures the negative definiteness of the time derivative of the Lyapunov function in (24). It is clear that the parameter γ_ϕ exists, is nonzero, nonnegative, and finite. These facts justify the particular chosen form of the Lyapunov function, and the analysis proves that the suggested form of the parameter update rule given in (20) leads to the stable training of artificial neural networks.

It is clear that the derivation and the analysis presented impose some conditions on the design parameters. In the rest of this section, these conditions are discussed.

1. Due to the requirement on the negative definiteness of the time derivative of the Lyapunov function, the inequality given in (50) must be satisfied:

$$C_\phi = 1 - \frac{2\alpha_1(\beta + Q_\phi + K_\phi)}{\alpha_1 + \alpha_2} > 0 \tag{50}$$

The selection for the learning rate ζ_ϕ imposes the following condition:

$$1 - \frac{2\alpha_1(\beta + Q_\phi + K_\phi)}{\alpha_1 + \alpha_2} > T_s \tag{51}$$

The inequality in (51) ensures the learning rate γ_ϕ assumes positive values. Since the condition in (51) includes the condition in (50), the constraint in (51) is one of the restrictions on the design parameters.

2. In order to ensure the stability of the imposed dynamics, which has already been analyzed in the fourth section, the following condition must hold true:

$$\beta(Q_\phi + K_\phi) < 1 \tag{52}$$

6. TRAINING OF ARTIFICIAL NEURAL NETWORKS

In this section, application of the devised scheme to feedforward neural networks is presented. In Refs. 5, 8–9, and 28, it is demonstrated that the structure can effectively be used for identification and control purposes. In the conventional EBP technique, propagating the output error back through the neural network, whose structure is illustrated in Fig. 1, minimizes the cost function given in (2). Based on the derivation presented in detail in Ref. 17, the delta values for the neurons belonging to the output layer and the hidden layers are evaluated as given by (53) and (54) respectively:

$$\delta_j^{k+1} = (d_j - f_j)\Psi'\left(S_j^{k+1}\right) \tag{53}$$

$$\delta_j^{k+1} = \left(\sum_{h=1}^{\text{neurons}_{k+2}} \delta_h^{k+2} w_{jh}^{k+1}\right)\Psi'\left(S_j^{k+1}\right) \tag{54}$$

Having evaluated the delta valus during the backward pass, the gradient based weight update rule described by (55) is applied for each training pair:

$$\Delta w_{ij\,\mathrm{EBP}}^k = \zeta_{w_{ij}^k}\delta_j^{k+1}o_i^k \tag{55}$$

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

$$\Delta w_{ij\,\mathrm{VSS}}^k = \beta \min\left(|\Delta w_{ij}^k|, |A_{w_{ij}^k}|\right)\mathrm{sgn}\left(N_{w_{ij}^k}\right) + A_{w_{ij}^k} \tag{56}$$

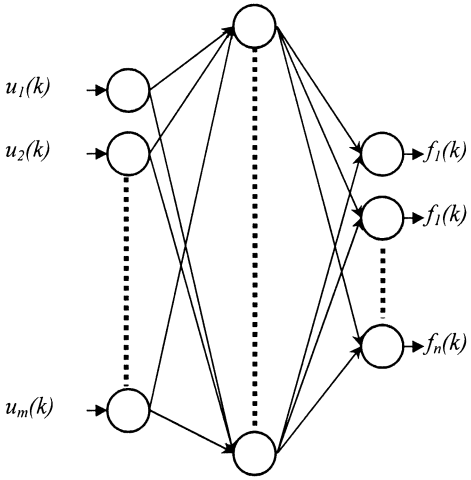

Figure 1. Structure of the feedforward neural network.

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

$$\Delta w_{ij}^{k} = \frac{\alpha_1 \, \Delta w_{ij\,\mathrm{VSS}}^{k} + \alpha_2 \, \Delta w_{ij\,\mathrm{EBP}}^{k}}{\alpha_1 + \alpha_2} \tag{57}$$

7. PLANT MODEL

In the simulations, the dynamic model of a three degrees of freedom anthropoid robotic manipulator, whose physical structure is illustrated in Figure 2, is used as the test bed. Since the dynamics of such a mechatronic system is modeled by nonlinear and coupled differential equations, precise output tracking becomes a difficult objective due to the strong interdependency between the variables involved and the existence of gravitational forces. 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 (58) 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 I in standard units:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \tag{58}$$

If the angular position and angular velocities are described as the state variables of the system, six coupled and first order differential equations can define the model. In (59) through (62), the nonzero entries of the state varying inertia

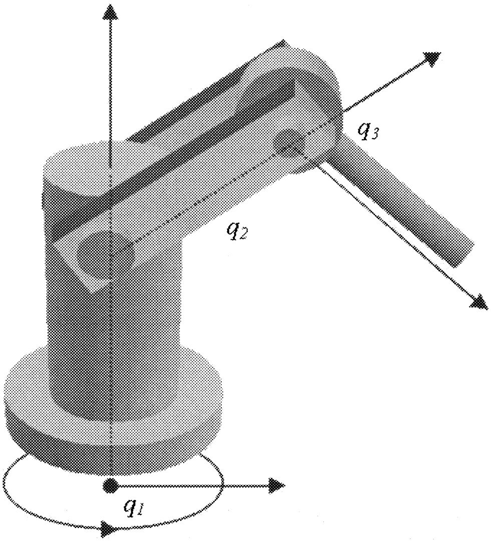

Figure 2. Physical structure of the 3-DOF robot used in the simulations.

Table I. 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
ith cylindrical link inertial parameter	$E_1 = m_1R^2/2$, $E_i = m_il_i^2/12$	E_i
ith cylindrical link inertial parameter	$A_i = m_iR^2/2$	A_i
ith 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	$\tau_{sat\,3}$

matrix are described. The nonzero Cristoffel symbols are given in (63) through (66). The details of the plant model are presented in Refs. 34 and 35.

$$\begin{aligned} M_{11} &= m_2l_{c2}^2\cos^2(q_2) + m_3\big(l_2\cos(q_2) + l_{c3}\cos(q_2+q_3)\big)^2 + E_1 \\ &\quad + A_2\sin^2(q_2) + E_2\cos^2(q_2) \\ &\quad + A_3\sin^2(q_2+q_3) + E_3\cos^2(q_2+q_3) \end{aligned} \tag{59}$$

$$M_{22} = m_2l_{c2}^2\sin^2(q_2) + m_3\big(l_2^2 + l_{c3}^2 + 2l_2l_{c3}\cos(q_3)\big) + I_2 + I_3 \tag{60}$$

$$M_{23} = M_{32} = m_3\big(l_{c3}^2 + l_{c3}l_2\cos(q_3)\big) + I_3 \tag{61}$$

$$M_{33} = m_3 l_{c3} + I_3 \tag{62}$$

$$\begin{aligned} hc_1 = {} & \big(-m_2 l_{c2}^2 + A_2 - E_2\big)\cos(q_2)\sin(q_2) \\ & + (A_3 - E_3)\cos(q_2 + q_3)\sin(q_2 + q_3) \\ & + m_3(l_2\cos(q_2) + l_{c3}\cos(q_2 + q_3))(-l_2\sin(q_2) - l_{c3}\sin(q_2 + q_3)) \end{aligned} \tag{63}$$

$$hc_2 = \sin(q_2 + q_3)\big(-m_3 l_{c3} l_2\cos(q_2) + \big(-m_3 l_{c3}^2 + A_3 - E_3\big)\cos(q_2 + q_3)\big) \tag{64}$$

$$hc_3 = m_2 l_{c2}^2\cos(q_2)\sin(q_2) \tag{65}$$

$$hc_4 = -m_2 l_2 l_{c3}\sin(q_3) \tag{66}$$

Coriolis and centrifugal terms are formulated as follows:

$$C(q, \dot{q}) = \begin{bmatrix} 2hc_1\dot{q}_1\dot{q}_2 + 2hc_2\dot{q}_1\dot{q}_3 \\ -hc_1\dot{q}_1^2 + 2hc_4\big(\dot{q}_2\dot{q}_3 + \dot{q}_3^2\big) + hc_3\dot{q}_2^2 \\ -hc_2\dot{q}_1^2 - hc_4\dot{q}_2^2 \end{bmatrix} \tag{67}$$

Lastly, the gravity terms are obtained as given in (68), where G represents the gravity constant:

$$g(q_1, q_2, q_3) = \begin{bmatrix} 0 \\ (m_2 l_{c2} + m_3 l_2)G\cos(q_2) + m_3 l_{c3} G\cos(q_2 + q_3) \\ m_3 l_{c3} G\cos(q_2 + q_3) \end{bmatrix} \tag{68}$$

8. SIMULATION RESULTS

In the simulations, the plant introduced in Section 7 is controlled by the neural network structure considered in Section 6. The architecture of the control system is illustrated in Figure 3, 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.

The main objective of the design presented is to achieve precise state tracking together with small parameter update effort. This is achieved through a suitable combination of the EBP algorithm and VSS methodology. 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 that of a proportional plus derivative (PD) controller having the following parameter

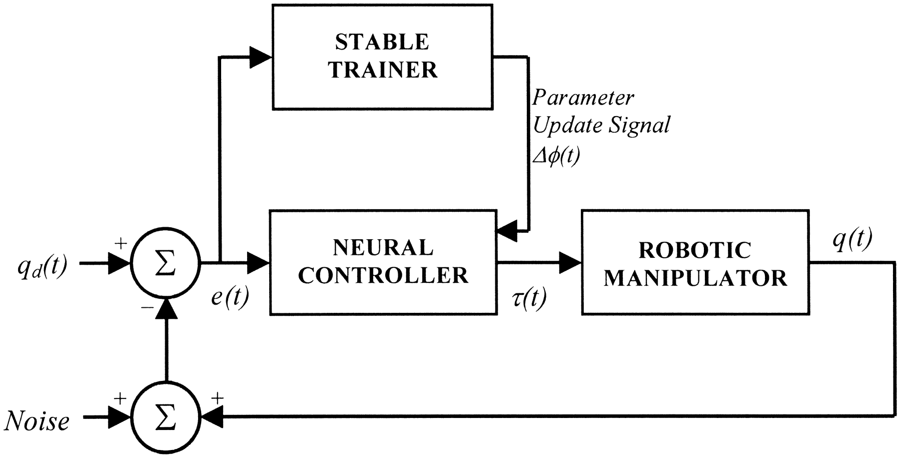

Figure 3. The architecture of the control system.

set:

$$\begin{bmatrix} K_{p\,\mathrm{BASE}} & K_{d\,\mathrm{BASE}} \\ K_{p\,\mathrm{SHOULDER}} & K_{d\,\mathrm{SHOULDER}} \\ K_{p\,\mathrm{ELBOW}} & K_{d\,\mathrm{ELBOW}} \end{bmatrix} = \begin{bmatrix} 40 & 40 \\ 180 & 260 \\ 150 & 70 \end{bmatrix} \tag{69}$$

The reference angular position and velocity profiles used in all simulations are depicted in Figure 4. 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. The perturbing signal is illustrated in Figure 5. 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, which excites the high frequency dynamics of the learning algorithm. 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 structures 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:

$$J_s(t) = \sum_{\phi} (\Delta\phi(t))^2 \tag{70}$$

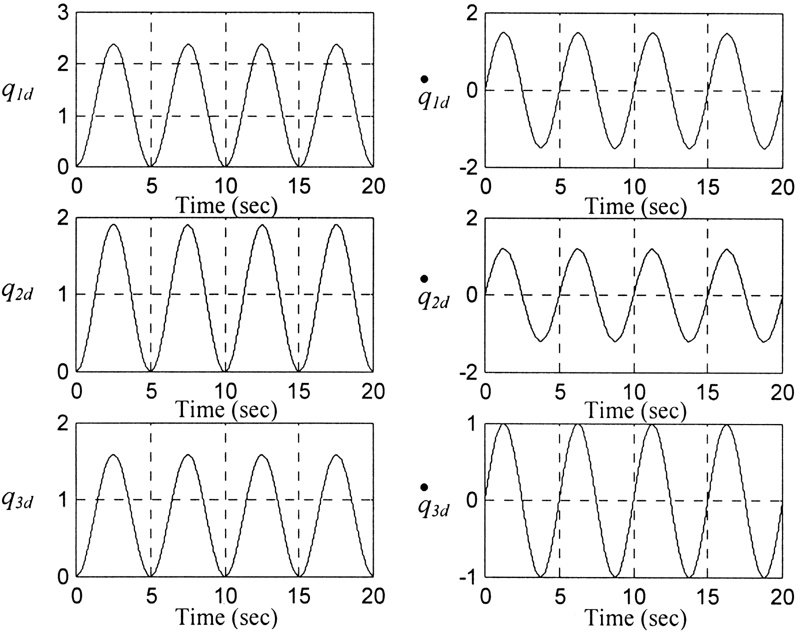

Figure 4. The reference angular position and velocity profiles.

For the use of the proposed algorithm, α_1 is set to 2 while α_2 is equal to 1. The state tracking errors and applied torque inputs are depicted in Figures 6 and 7, respectively. It is evident from Figure 6 that the proposed combination results in precise state tracking under the existence of environmental and structural difficulties stated above. Furthermore, Figure 7 emphasizes that the control signals evaluated by the neurocontroller lie within the limits of applicable control ranges. Therefore, the produced torque signals are directly applied to the manipulator without requiring saturation. With the same initial conditions, if α_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 Figure 8. Unsurprisingly, the produced torque signals, which are illustrated in Figure 9, tend to diverge and control saturation becomes active. It is apparent from Figure 9 that neither the magnitudes nor the frequencies of the produced torque signals are suitable for actuating such a mechatronic device.

The behavior of the total parametric cost described by (70) is depicted in Figure 10. The left plot of Figure 10 indicates that the cost in (70) reaches 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 that 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 the EBP technique more visible.

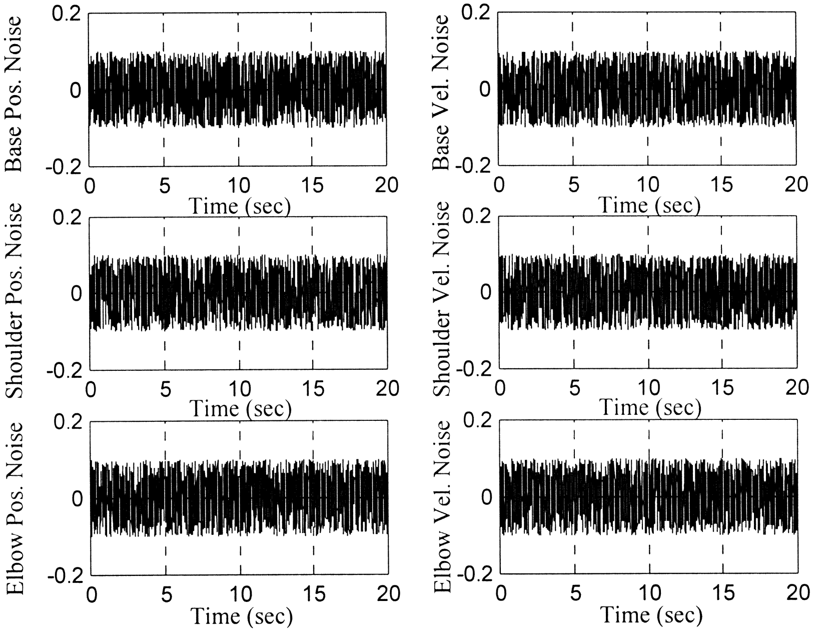

Figure 5. Noise sequences perturbing the state variables.

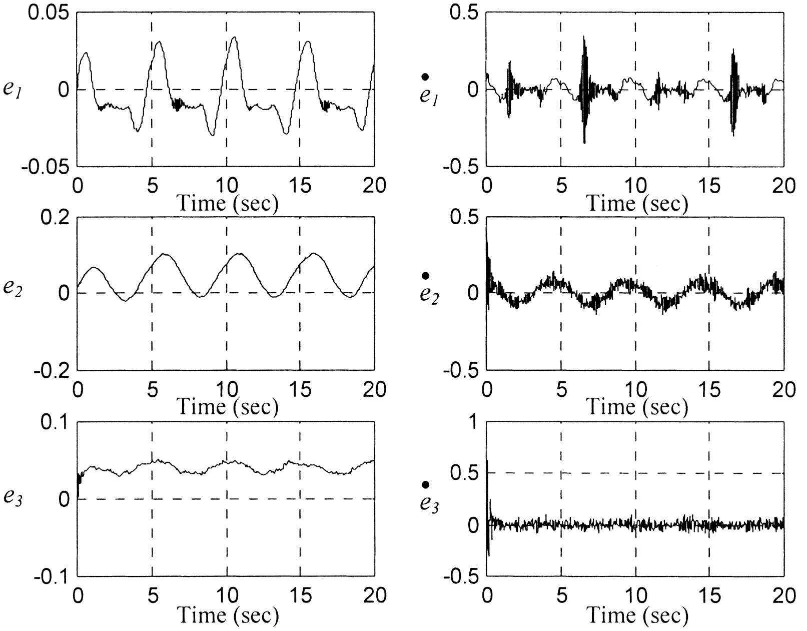

Figure 6. State tracking errors observed with the proposed technique.

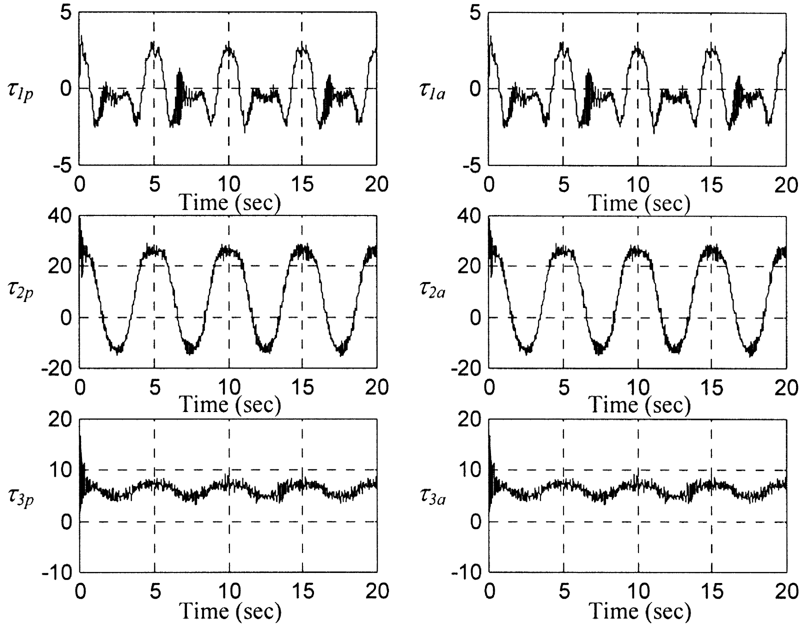

Figure 7. Produced and applied torque signals with proposed technique.

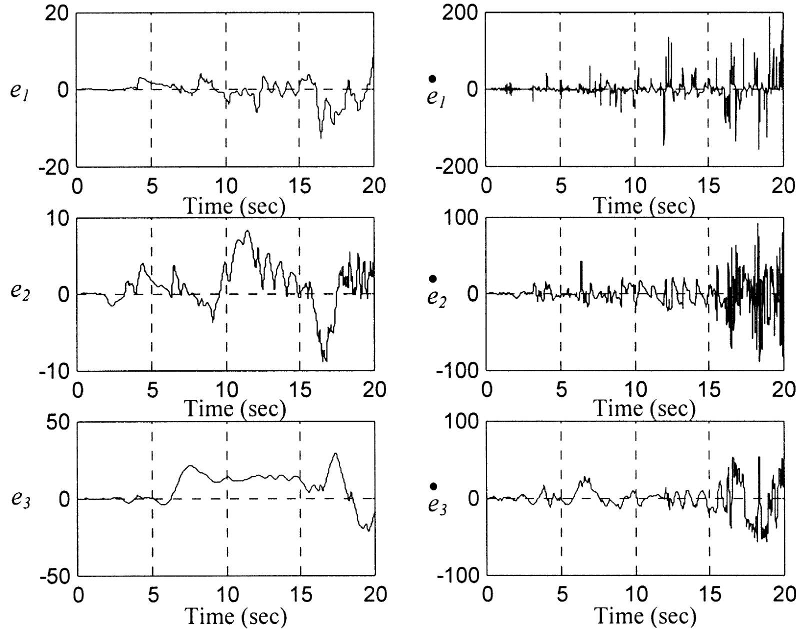

Figure 8. State tracking errors observed with pure EBP technique.

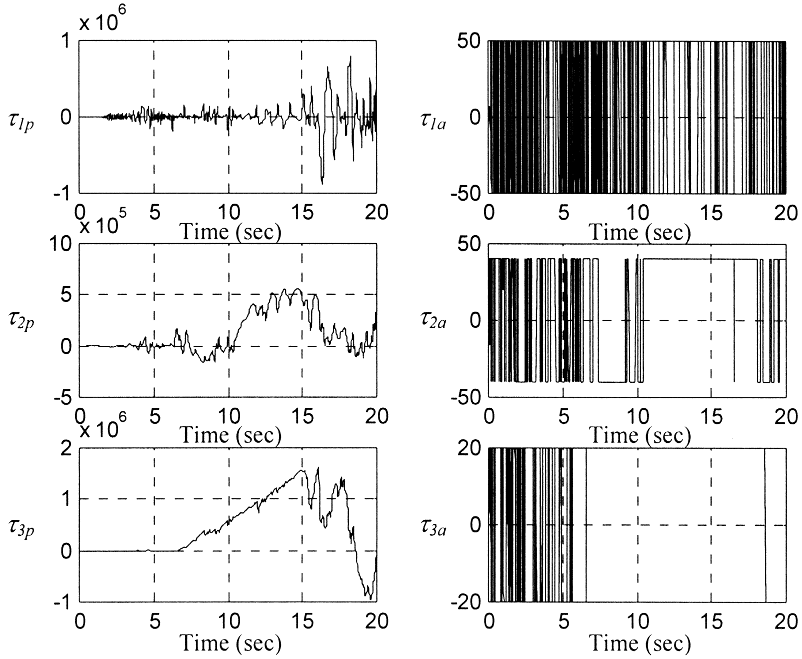

Figure 9. Produced and applied torque signals with pure EBP technique.

Clearly, the presence of observation noise and the requirements of the problem in hand stimulate the unstable internal dynamics of the EBP method. It is apparent from the right subplot of Figure 10 that the average magnitude is increasing in time.

The simulation settings are tabulated in Table II, in which it is apparent that the constraints stated in (51) and (52) are satisfied.

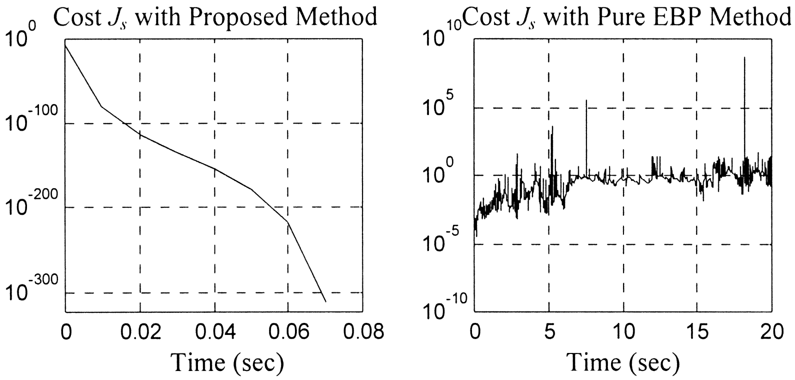

Figure 10. Time behavior of the parametric cost.

Table 2. Simulation parameters for FNN controller

T_s	1.0 msec.
β	0.1
$\alpha_{1,2}$	See section 8
Q	0.1
K	0.1
ε	1.0
#Input neurons	6
#Hidden neurons	12
#Output neurons	3

9. CONCLUSIONS

One of the major problems in applications of gradient based training strategies is the lack of stabilizing forces preventing the unbounded growth in the magnitudes of the adjustable parameters. 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, we propose a method for creating stabilizing forces on the training dynamics. The method is based on the integration of EBP strategy with VSS technique to benefit from the robustness property of the VSS approach as well as the cost minimizing property of the EBP method. An analytical study of the conditions of stability in the parameter change space is presented. Simulation studies are carried out to compare the performance of the proposed scheme with that obtained with pure EBP technique. For this purpose, a feedforward neural network is used and the adaptation is performed on the weights and biases of the structure. The task assigned to the neural network is the control of a three degrees of freedom anthropoid robotic manipulator. The difficulties to be compensated are the existence of observation noise and the dynamic complexity of the system under control. The results stipulate that the proposed approach fulfills the task to be accomplished much better than 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.

This work is supported by Bogazici University Research Fund, Project 99A202.

References

1. Hornik K. Neural Networks 1989;2:359–366.
2. Funahashi K. Neural Networks 1989;2:183–192.
3. Gupta MM, Rao DH. J Intell Fuzzy Syst 1993;1:73–92.
4. Wang Y-J, Lin C-T. IEEE Trans Neural Networks 1998;9:294–307.

5. Efe MO, Kaynak O. Mechatronics 1999;9:287–300.
6. Moon S, Hwang J-N. IEEE Trans Neural Networks 1997;8:194–207.
7. Allen TJ, Hung K-S, Curtis KM, Orton JW. IEEE Trans Neural Networks 1996;7:488–500.
8. Narendra KS, Parthasarathy K. IEEE Trans Neural Networks 1990;1:4–27.
9. Efe MO, Abadoglu E, Kaynak O. Int J Robust Nonlin Contr 1999;9:799–815.
10. Kandadi RM, Tien JM. IEEE Trans Neural Networks 1997;8:1531–1541.
11. Plumer ES. IEEE Trans Neural Networks 1996;7:408–418.
12. Park Y-M, Choi M-S, Lee KY. IEEE Trans Neural Networks 1996;7:1099–1110.
13. Lawrence S, Back AD, Tsoi AC, Giles CL. IEEE Trans Neural Networks 1997;8:1507–1517.
14. Zhang X, Hang C-C, Tan S, Wang P-Z. IEEE Trans Neural Networks 1996;7:1139–1150.
15. Campolucci P, Uncini A, Piazza F, Rao BD. IEEE Trans Neural Networks 1999;10:253–271.
16. Maeda Y, De Figueiredo RJP. IEEE Trans Neural Networks 1997;8:1119–1130.
17. Rumelhart DE, Hinton GE, Williams RJ. In Parallel Distributed Processing, Rumelhart DE. and McClelland JL., Eds.; MIT Press: Cambridge, MA, 1986, vol. 1, pp 318–362.
18. Bouchard M, Paillard B, Dinh CTL. IEEE Trans Neural Networks 1999;10:391–401.
19. Kwon TM, Cheng H. IEEE Trans Neural Networks 1996;7:515–524.
20. Fu LM, Hsu H-H, Principe JC. IEEE Trans Neural Networks 1996;7:757–761.
21. Younger AS, Conwell PR, Cotter NE. IEEE Trans Neural Networks 1991;10:272–283.
22. Emelyanov SV. Variable Structure Control Systems; Nauka: Moscow, 1967.
23. Hung JY, Gao W, Hung JC. IEEE Trans Indust Electron 1993;40:2–22.
24. Gao W, Hung JC. IEEE Trans Indust Electron 1993;40:45–55.
25. Erbatur K, Kaynak O, Sabanovic A. IEEE Trans Indust Electron 1999;46:1–7.
26. Kaynak O, Harashima F, Hashimoto H. Trans IEE Japan 1984;104:47–52.
27. Bekiroglu N. Adaptive sliding surface design for sliding mode control systems, Doctoral Thesis, Bogazici University, 1996.
28. Ertugrul M. Neuro-sliding mode control of robotic manipulators, Doctoral Thesis, Bogazici University, 1999.
29. Young KD, Utkin VI, Ozguner U. IEEE Trans Contr Syst Technol 1999;7:328–342.
30. Kaynak O, Denker A. Int J Contr 1993;57:1177–1189.
31. Erbatur K, Kaynak O, Sabanovic A, Rudas I. Robotics Auton Syst 1996;19:215–227.
32. Byungkook Y, Ham W. IEEE Trans Fuzzy Syst 1998;6:315–321.
33. Sira-Ramirez H, Colina-Morles F. IEEE Trans Circuits Syst 1995;42:1001–1012.
34. Erbatur K, Vinter RB, Kaynak O. Proc 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, Vol. 4, 1994, pp. 2979–2984.
35. Stadler W. Analytical Robotics and Mechatronics, McGraw Hill: New York, 1995, p 476.

APPENDIX

NOMENCLATURE

f_j	jth output of ANN structure
ϕ	A generic parameter of ANN
ϕ^*	Optimal value of the generic parameter
$\Delta\phi$	Change in parameter ϕ
e_j	jth entry of observed output error vector
d_j	jth entry of desired output vector
J_r	Realization cost

J_s	Parametric cost
η_ϕ	Learning rate for parameter ϕ
T_s	Sampling interval of update dynamics
s_ϕ	Switching function for parameter ϕ
Q_ϕ	Gain of the switching scheme
K_ϕ	Gain of the switching scheme
ε	Boundary layer parameter
N_ϕ	Backpropagated error value for parameter ϕ
β	Scaling factor for parameter stabilizing law
ζ_ϕ	Learning rate for cost minimizing law
V_ϕ	Lyapunov function for parameter ϕ
α_i	Weighting factor
u_j	jth input of ANN structure
q_{id}	Desired state trajectory for ith link
q_i	Actual state trajectory for ith link
Ψ	Neuronal nonlinear activation function
S_j^{k+1}	Net summation of the jth neuron in the $(k+1)$th layer
δ^{k+j}	Delta value of the jth neuron in the $(k+1)$th layer
o_i^k	Output of the ith neuron in the kth layer
τ_{ip}	Produced torque input for ith link
τ_{ia}	Applied torque input for ith link