INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2003; **17**: 489–500 (DOI: 10.1002/acs.755)

A robust on-line learning algorithm for intelligent control systems

Mehmet Önder Efe[1,*,†], Okyay Kaynak[2,‡], Bogdan M. Wilamowski[3,§] and Xinghuo Yu[4,¶]

[1] *Atılım University, Department of Mechatronics Engineering Incek, TR-06836 Ankara, Turkey*
[2] *Bogazici Univerisity, Electrical and Electronic Engineering Department, Bebek, 34342, Istanbul, Turkey*
[3] *University of Idaho, Graduate Center at Boise, 800 Park Blvd., Boise, ID 83712, USA*
[4] *School of Electrical and Computer Engineering, RMIT University, PO Box 2476V, Melbourne, VIC 3001, Australia*

SUMMARY

This paper describes a novel error extraction approach for exploiting the strength of Levenberg–Marquardt (LM) optimization technique in intelligent control systems. Since the target value of the control signal is unknown, tuning of the controller parameters becomes a tedious task if the knowledge about the system and the environment is limited. The suggested methodology utilizes the sliding model control (SMC) technique. The error extraction scheme postulates the form of error on the applied control signal using the discrepancy from the prescribed reaching dynamics. The devised approach has been tested on the non-linear Duffing oscillator, which has been forced to follow a periodic orbit radically different from the natural one. The results obtained through a series of simulations have confirmed the high precision and robustness advantages without knowing the analytical details of the system under investigation. The issues of observation noise and the stability in the parametric space have approximately been addressed from the point of SMC perspective. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: sliding mode control; Levenberg–Marquardt algorithm; Duffing oscillator; learning

1. INTRODUCTION

Autonomous and intelligent control have constituted core research fields in the recent years and tremendous amount of research outcomes have reported the methods to overcome uncertainties and to improve the precision of the realization by flexible structures such as neural networks,

*Correspondence to: Dr. Mehmet Önder Efe, Atılım University, Department of Mechatronics Engineering Incek, TR-06836 Ankara, Turkey.
† E-mail: onderefe@ieee.org
‡ E-mail: kaynak@boun.edu.tr
§ E-mail: wilam@ieee.org
¶ E-mail: x.yu@rmit.edu.au

Contract/grant sponsor: NSF; contract/grant number: 9906233

Accepted 29 May 2003

fuzzy systems and those hybrid combinations with the methods of artificial intelligence. A significant breakthrough in the context of parameter adjustment was the discovery of Error Backpropogation (EBP) algorithm [1]. The studies appeared in the literature after the resurgence of neural networks have focused on the improvement of convergence speed of EBP through momentum addition and variable learning rates [2]. Among many alternatives existing in the literature, the methods exploiting the second order derivatives of the cost hypersurface have been proved to be more successful in terms of realization precision however with extra computational burden. Gauss–Netwon method, LM optimization technique and conjugate gradient algorithm are just a few examples [3–7] used for training of intelligent systems.

When the tuning is performed for the purpose of control, the designer is faced to the alleviation of several difficulties stemming from the uncertainties, corrupted observations and nonlinearities inherently existing in the system dynamics. Therefore the adopted control scheme must adequately be equipped to compensate these difficulties through the refinement of the information content of an intelligent controller. From a stability and robustness point of view, variable structure systems (VSS) theory offers high performance solutions to the problem of parameter tuning. A particular design framework in VSS theory is known as sliding mode control and has extensively been used for motion control systems. A good deal of material addressing the conventional SMC design issues. relevance of intelligence and SMC, practical aspects and SMC in discrete time cases can be found in References [8–13].

The idea of tuning the parameters of an intelligent controller is not new. Some examples utilizing VSS theory have successfully demonstrated that the approach can be utilized for tracking control of uncertain systems and identification purposes [14, 15]. The underlying idea is to integrate the robustness and invariance properties of SMC technique with the power of knowledge based systems like neural networks and fuzzy inference systems. If the task to be achieved in the SMC of a plant with SMC in tuning scheme, one should notice that the tuning mechanism would entail an error critic qualifying the output of the controller. Although the nature of the control problems does not allow the existence of a supervisory information on the controller outputs, it can be shown that an appropriate error measure can be constructed particularly for SMC purposes.

In what follows, the LM optimization technique is briefly presented. The third section figures out the structure of the controller, and postulates the LM updates with the extracted error measure based on SMC framework. In the fourth section, we present a series of simulation studies demonstrating the efficacy of the presented novelties. The concluding remarks constitute the last part of the paper.

2. LEVENBERG–MARQUARDT OPTIMIZATION TECHNIQUE

The LM algorithm is an approximation to the Newton's method, and both of them have been designed to solve the nonlinear least squares problem [6]. Consider an intelligent system having K outputs, and N adjustable parameters denoted by the vector $\underline{\omega}$. If there are P data points (or patterns) over which the interpolation is to be performed, a cost function qualifying the performance of the interpolation can be given as $E(\underline{\omega}) = \sum_{p=1}^{P} \sum_{k=1}^{K} (\tau_{kp}^{*} - \tau_{kp}(\underline{\omega}))^2$, where τ_{kp} is the observation at the kth output of the structure in response to the pth pattern, and τ_{kp}^{*} is the corresponding target entry. The parameter update prescribed by Newton's algorithm is given as $\underline{\omega}_{n+1} = \underline{\omega}_n - (\nabla_{\omega}^2 E(\underline{\omega}_n))^{-1} \nabla_{\omega} E(\underline{\omega}_n)$. Knowing $\nabla_{\omega}^2 E(\underline{\omega}_n) = 2J(\underline{\omega})^{\mathrm{T}} J(\underline{\omega}_n) + g(\underline{\omega}_n)$ and

$\nabla_{\omega} E(\underline{\omega}_n) = 2J(\underline{\omega}_n)^{\mathrm{T}} \underline{e}(\underline{\omega}_n)$ with $\underline{e}$ and J being the error vector and the Jacobian as given in (1) and (2) respectively, the Gauss–Newton algorithm can be formulated as $\underline{\omega}_{n+1} = \underline{\omega}_n - (J(\underline{\omega})^{\mathrm{T}} J(\underline{\omega}_n))^{-1} \; J(\underline{\omega}_n)^{\mathrm{T}} \; \underline{e}(\underline{\omega}_n)$, and the LM update can be constructed as $\underline{\omega}_{n+1} = \underline{\omega}_n - (\mu I_{N\times N} + J(\underline{\omega})^{\mathrm{T}} J(\underline{\omega}_n))^{-1} \; J(\underline{\omega}_n)^{\mathrm{T}} \; \underline{e}(\underline{\omega}_n)$ where $\mu > 0$ is a scalar design parameter. Clearly, the Gauss-Newton method assumes that the entries of the matrix $g(\underline{\omega}_n)$ are negligible small in magnitude, and the LM technique improves the rank deficiency problem of the matrix $J(\underline{\omega}_n)^{\mathrm{T}} J(\underline{\omega}_n)$.

$$\underline{e} = [e_{11} \;\; \cdots e_{K1} \;\; e_{12} \;\; \cdots e_{K2} \;\; \cdots e_{1P} \;\; \cdots e_{KP}]^{\mathrm{T}} \tag{1}$$

$$J\left(\underline{\omega}\right) = \begin{bmatrix} \frac{\partial e_{11}\left(\underline{\omega}\right)}{\partial \omega_1} & \frac{\partial e_{11}\left(\underline{\omega}\right)}{\partial \omega_2} & \cdots & \frac{\partial e_{11}\left(\underline{\omega}\right)}{\partial \omega_N} \\ \frac{\partial e_{21}\left(\underline{\omega}\right)}{\partial \omega_1} & \frac{\partial e_{21}\left(\underline{\omega}\right)}{\partial \omega_2} & \cdots & \frac{\partial e_{21}\left(\underline{\omega}\right)}{\partial \omega_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{K1}\left(\underline{\omega}\right)}{\partial \omega_1} & \frac{\partial e_{K1}\left(\underline{\omega}\right)}{\partial \omega_2} & \cdots & \frac{\partial e_{K1}\left(\underline{\omega}\right)}{\partial \omega_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{1P}\left(\underline{\omega}\right)}{\partial \omega_1} & \frac{\partial e_{1P}\left(\underline{\omega}\right)}{\partial \omega_2} & \cdots & \frac{\partial e_{1P}\left(\underline{\omega}\right)}{\partial \omega_N} \\ \frac{\partial e_{2P}\left(\underline{\omega}\right)}{\partial \omega_1} & \frac{\partial e_{2P}\left(\underline{\omega}\right)}{\partial \omega_2} & \cdots & \frac{\partial e_{2P}\left(\underline{\omega}\right)}{\partial \omega_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_{KP}\left(\underline{\omega}\right)}{\partial \omega_1} & \frac{\partial e_{KP}\left(\underline{\omega}\right)}{\partial \omega_2} & \cdots & \frac{\partial e_{KP}\left(\underline{\omega}\right)}{\partial \omega_N} \end{bmatrix} \tag{2}$$

Contrary to what is postulated in the case of EBP method, the framework of LM optimization technique utilizes the second order partial derivatives of the cost measure, and therefore extracts a better path towards the goal in the adjustable parameter space. The cost of this is the computational burden primarily due to the matrix inversion taking place at each iteration.

3. NEUROCONTROLLER AND THE ERROR EXTRACTION SCHEME

Assume that the plant is to be controlled by a feedforward neural network, which has the input output relation given as $\underline{\tau} = W_R \underline{\Xi}(W_L \underline{u} - \underline{B}_L) - \underline{B}_R$. This representation reveals that the network has one hidden layer, the neurons in which have the activation functions forming the vector $\underline{\Xi}$, two matrices denoted by W_R and W_L, and two vectors acting as the multiplying weights of the bias inputs of value minus unity. The output layer has linear neurons. Concatenating W_L, W_R, $\underline{B}_L$ and $\underline{B}_R$ in a single column forms the adjustable parameter vector $\underline{\omega}$ and the structure is excited by the input vector denoted by $\underline{u}$.

Consider the control system structure depicted in Figure 1, in which the plant inside the dashed rectangle is a SISO one, whose states are assumed to be observable. The inputs to the plant and the observed states are sampled by Zero Order Holders (ZOH) as shown in the figure. Note that the subscript n stands for discrete time index, and the dynamics inside the dashed rectangle is governed by a set of difference equations of the form given below.

$$\underline{x}_{n+1} = \underline{f}(\underline{x}_n) + \underline{h}(n)\tau_n \tag{3}$$

where $\underline{x}_n = [x_{1n} \quad x_{2n} \quad \dots \quad x_{Mn}]^{\mathrm{T}}$ is the state vector, $\underline{f}(\underline{x}_n)$ is a non-linear vector function of the system state and is unavailable, whereas $\underline{h}(n)$ is a vector function of time and the sign of it is known. The system above can compactly be written as $\underline{x}_{n+1} = \underline{f}_n + \underline{h}_n \tau_n$. According to Figure 1, the error vector at time n is defines as $\underline{u}_n = \underline{x}_n - \underline{r}_n$, where $\underline{r}_n$ is the vector of reference state trajectories at time n. Define the switching function as

$$s_n = \underline{\alpha}^{\mathrm{T}} \underline{u}_n \tag{4}$$

in which the vector $\underline{\alpha}$ is selected such that the dynamics determine by $s_n = 0$ is stable, and it is assumed that $\underline{\alpha}^{\mathrm{T}}\ \underline{h}_n > 0$. Now adopt a closed loop switching dynamics described generically as $s_{n+1} = Q(\mathrm{s}_n)$, and the evaluate s_{n+1} as given below.

$$s_{n+1} = \underline{\alpha}^{\mathrm{T}}(\underline{f}_n + \underline{h}_n \tau_n - \underline{r}_{n+1}) \tag{5}$$

Using $s_n + 1 = Q(S_n)$ and solving for τ_n gives the control sequence formulated as below.

$$\tau_n = -(\alpha^{\mathrm{T}} \underline{h}_n)^{-1}(\underline{\alpha}^{\mathrm{T}}(\underline{f}_n - r_{n+1}) - Q(s_n)) \tag{6}$$

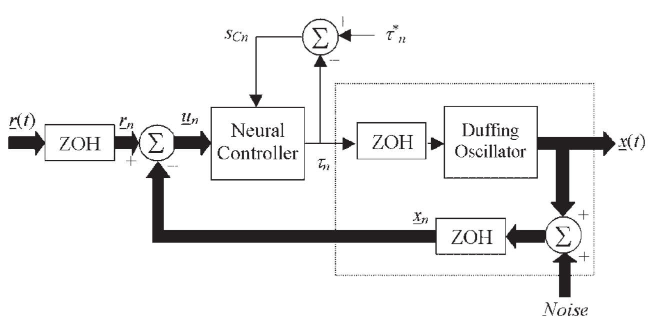

Figure 1. Structure of the feedback control system.

If the values of the vector functions $\underline{f}_n$ and $\underline{h}_n$ were known explicitly, the application of this sequence to the system of (3) would result in $s_{n+1} = Q(s_n)$, where Q must satisfy the condition below to ensure reaching [11,12].

$$s_n(s_{n+1} - s_n) = s_n(Q(s_n) - s_n) < 0 \tag{7}$$

If the condition above is satisfied for $\forall n \geq 0$, the system is driven towards the dynamics characterized by $s_n = 0$. However in practice, $s_n = 0$ is rarely observed as the problem is described in discrete time. A realistic observation is $|s_n| < \varepsilon$, where ε is some positive number. In the literature, this phenomenon is called quasi-sliding mode, or equivalently pseudo-sliding mode [12,16,17]. This mode has useful invariance properties in the face of uncertainties and time variations in the plant and/or environment parameters. Once the quasi-sliding regime starts, the error signal behaves as what is prescribed by $|s_n| < \varepsilon$.

Define the task as the discrete time sliding mode control (DTSMC) of a plant of the form given in (3), whose ultimate behaviour is to be enforced towards what is prescribed by $s_{n+1} = Q(s_n)$. Consider Figure 1, which demonstrates that the quantity s_{Cn} would be the error on the applied control signal if we had a supervisor providing the desired value of the control denoted by τ_n^*. However, the nature of the problem does not allow the existence of such a supervisory information; instead, the designer is forced to extract the value of s_{Cn} from the available quantities. In what follows, we present a method to extract the error on the control signal.

Assumption 3.1

The vector functions $\underline{f}_n$ and $\underline{h}_n$ of (3) are such that a desired quasi-sliding mode can be created with a suitable selection of the design parameters; more explicitly, we assume that the DTSMC task is achievable.

Remark 3.2

A control sequence leading to desired DTSMC can be formulated if the dynamics of the system described by (3) is totally known or if the nominal representation is known with the bounds of the uncertainties. It must be noted that the disturbances and uncertainties are assumed to enter the system through the control channel [8]. When the control sequence in (6) is applied to the system of (3), we call the resulting behaviour as the *target DTSMC* and the input signal leading to it as the *target control sequence* (τ_n). If at least the explicit forms of the nominal representations of the vector functions $\underline{f}_n$ and $\underline{h}_n$ are not known, it should be obvious that the target control sequence cannot be constructed under such an uncertainty by following the traditional DTSMC design approaches.

Definition 3.3

Given an uncertain plant, which has the structure described as in (3), and a command trajectory $\underline{r}_n$ for $n \geqslant 0$, the input sequence denoted by τ_n^* satisfying the following difference equation is defined to be the *idealized control sequence*, and the difference equation itself is defined to be the *reference DTSMC model*. In this representation, $\underline{r}_n = [r_{1n} \;\; r_{2n} \;\; \ldots \;\; r_{Mn}]^{\mathrm{T}}$ stands for the vector of command trajectories.

$$\underline{r}_{n+1} = \underline{f}(\underline{r}_n) + \underline{h}(n)\tau_n^* \tag{8}$$

Mathematically, the existence of such a model and the sequence means that the system of (3) perfectly follows the command trajectory ($\underline{r}_n$) if both the idealized control sequence (τ_n^*) is known and the initial conditions are set as $\underline{x}_0 = \underline{r}_0$; more explicitly $\underline{u}_n \equiv \underline{0}$ for $\forall n \geq 0$. Undoubtedly, the reference DTSMC model is an abstraction as the functions appearing in it are not available. However, the concept of idealized control sequence should be viewed as the synthesis of the command signal $\underline{r}_n$ from the time solution of the difference equation in (8).

Fact 3.4

If the target control sequence formulated in (6) were applied to the system of (3), the idealized control sequence would be the steady state solution of the control signal, i.e. $\lim_{n\to\infty} \tau_n = \tau_n^*$. However, under the assumption of the achievability of the DTSMC task, the difficulty here again is the unavailability of the functional forms of $\underline{f}_n$ and $\underline{h}_n$. Therefore, the aim is to discover an equivalent form of the discrepancy between the control applied to the system and its target value by utilizing the idealized control viewpoint. This discrepancy measure is denoted by $s_{Cn} = \tau_n - \tau_n^*$. If the target control sequence of (6) is rewritten by using (8), one gets

$$\begin{aligned}
\tau_n &= -\left(\underline{\alpha}^{\mathrm{T}}\underline{h}_n\right)^{-1}\left(\underline{\alpha}^{\mathrm{T}}\left(\underline{f}\left(\underline{x}_n\right) - \underline{f}\left(\underline{r}_n\right) - \underline{h}_n\tau_n^*\right) - Q(s_n)\right) \\
&= -\left(\underline{\alpha}^{\mathrm{T}}\underline{h}_n\right)^{-1}\left(\underline{\alpha}^{\mathrm{T}}\left(\underline{f}\left(\underline{x}_n\right) - \underline{f}\left(\underline{r}_n\right)\right) - Q(s_n)\right) + \tau_n^* \\
&= -\left(\underline{\alpha}^{\mathrm{T}}\underline{h}_n\right)^{-1}\left(\underline{\alpha}^{\mathrm{T}}\,\underline{\Delta f}_n - Q(s_n)\right) + \tau_n^*
\end{aligned} \tag{9}$$

where $\underline{\Delta f}_n = \underline{f}(\underline{x}_n) - \underline{f}(\underline{r}_n)$. The target control sequence becomes identical to the idealized control sequence, i.e. $\tau_n = \tau_n^*$ as long as $\underline{\alpha}^{\mathrm{T}}\underline{\Delta f}_n - Q(s_n) = 0$ holds true for $\forall n \geq 0$. However, this condition is of no practical importance as the analytic form of the function $\underline{f}_n$ is not available. Therefore, one should consider this equality as an equality to be enforced instead of an equality that holds true all the time, because its implication is $s_{Cn} = 0$, which is the ultimate goal of the design. It is obvious that to enforce this equality to hold true will let us synthesize the target control sequence, which will eventually converge to the idealized control sequence by the adaptation algorithm yet to be discussed. Consider s_{n+1} given below.

$$\begin{aligned}
s_{n+1} &= \underline{\alpha}^{\mathrm{T}}\left(\underline{x}_{n+1} - \underline{r}_{n+1}\right) \\
&= \underline{\alpha}^{\mathrm{T}}\left(\underline{f}\left(\underline{x}_n\right) + \underline{h}_n\tau_n - \underline{f}(r_n) - \underline{h}_n\tau_n^*\right) \\
&= \underline{\alpha}^{\mathrm{T}}\left(\underline{\Delta f}_n + \underline{h}_n s_{Cn}\right) \\
&= Q(s_n) + \underline{\alpha}^{\mathrm{T}}\underline{h}_n s_{Cn}
\end{aligned} \tag{10}$$

Solving the above equation for s_{Cn} yields the following:

$$s_{Cn} = \left(\underline{\alpha}^{\mathrm{T}}\underline{h}_n\right)^{-1}\left(s_{n+1} - Q(s_n)\right) \tag{11}$$

The interpretation of the above control error measure is as follows: since we are in pursuit of enforcing $s_{n+1} = Q(s_n)$ in the closed loop, during the time until which this equality does not hold true, the applied control sequence carries some error. However, if the tuning activity in the neurocontroller enforces (11) to approach zero, this enforces $\underline{\alpha}^{\mathrm{T}}\underline{\Delta f}_n - Q(s_n) = 0$ to approach zero. i.e. $s_{n+1} \to Q(s_n)$, consequently $\tau_n \to \tau_n^*$ as n increases.

Remark 3.5

Notice that the application τ_n^* for $\forall n \geq 0$ to the system of (3) with zero initial errors will lead to $\underline{u}_n \equiv \underline{0}$ for $\forall n \geq 0$. On the other hand, the application of τ_n for $\forall n \geq 0$ to the system of (3) will lead to $s_n = 0$ for $\forall n \geq n_h$, where n_h is the hitting time index, at which the quasi-sliding regime starts. Therefore, the adoption of (11) as the equivalent measure of the control error loosens $\underline{u}_n \equiv \underline{0}$ for $\forall n \geq 0$ requirement and enforces $s_{n+1} \to Q(s_n)$. Consequently, the tendency of the control scheme will be to generate the target DTSMC sequence τ_n of (6).

Remark 3.6

Referring to (11), it should be obvious that if $s_{Cn}(s_{C\,n+1} - s_{Cn}) < 0$ is satisfied $s_n(s_{n+1} - s_n) < 0$ is enforced. In other words, if the control signal approaches the target control sequence, the DTSMC task is achieved and the plant follows the command signal.

Remark 3.7

For an on-line training strategy, there occurs only one training pair at a time ($P = 1$). Since $s_{Cn} = \tau_n - \tau_n^*$, and since the plant is a SISO one, the cost at each instant of time can be defined as

$$E_{Cn} = s_{Cn}^2 = e\left(\underline{\omega}_n\right)^2 \tag{12}$$

which instantly qualifies the similarity between τ_n and τ_n^*. If the parameters of the neurocontroller are tuned with LM strategy, the cost in (12) is enforced toward zero and the task implied by $s_{Cn} = 0$ is achieved. More explicitly, the update rule takes the form $\underline{\omega}_{n+1} = \underline{\omega}_n - \mu I_{N\times N} + J(\underline{\omega}_n)^{\mathrm{T}} J(\underline{\omega}_n)^{\mathrm{T}} s_{Cn}$. This apparently suggests that a system of structure (3) in the feedback loop illustrated in Figure 1 can be driven towards a predefined quasi-sliding mode if the training algorithm for the adopted neurocontroller enforces the minimization of the cost measure given in (12), which is enforced by the above tuning law.

It should again be emphasized that the design presented here does not assume the availability of the functions seen in the dynamics of the system of (3). Therefore, one might argue the usability of (11) due to the causality requirement. However, in the implementation stage, the information provided by (11) supervises the controller as a task specific estimate, therefore, at time n, the quantity calculated from the available signals is constructed and used as the control error, i.e. the control error is estimated as $s_{Cn} = \kappa(s_n - Q(s_{n-1}))$, where κ is a constant of known sign due to $\underline{\alpha}^{\mathrm{T}}\underline{h}_n > 0$.

4. SIMULATION STUDIES

Understanding the chaotic behaviour has constituted a challenge for years as the outputs from which represent the entire richness of nonlinear phenomena during the course of even the finite-time observations. The behavioural diversity in chaos is therefore attributed to its deterministic unpredictability or the unpredictable determinism, which exists in the nature of the system. Furthermore, sensitive dependence to the initial conditions makes the chaotic systems attractive test beds to test the performance of novel control algorithms.

In the simulation studies, we discuss the Duffing system studies in Reference [18], which focuses on the identification and control issues for a number of chaotic systems. The dynamics of the system in continuous time is given as follows:

$$\dot{x}_1 = x_2 \quad \text{and} \quad \dot{x}_2 = -\varphi_1 x_1 - \varphi_2 x_1^3 - \sigma x_2 + \rho\cos(\beta t) + \tau \tag{13}$$

where, $\varphi_1 = 1.1$, $\varphi_2 = 1$, $\sigma = 0.4$, $\rho = 2.1$ and $\beta = 1.8$. The control problem is to enforce the states to the periodic orbit described as follows:

$$\dot{r}_1 = r_2 \quad \text{and} \quad \dot{r}_2 = -\sin(r_1), \quad r_1(0) = 1, \quad r_2(0) = 0 \tag{14}$$

When the system of (13) and the reference dynamics of (14) are discretized with first order Euler approximation, one ends up with the generic representation of (3), and the approach discussed becomes applicable. The response of the uncontrolled system for several different initial conditions is illustrated together with the desired periodic orbit in Figure 2. It can easily be seen that the desired orbit is sufficiently away from the natural limit cycle of the system, and therefore a continuous control action is required to enforce the system towards the reference dynamics. Parallel to what is discussed in Reference [12], in the simulations, we used the following reaching behaviour.

$$Q(s_n) = -\Lambda_1 T_s \operatorname{sgn}(s_n) - (1 - \Lambda_2 T_s) s_n \tag{15}$$

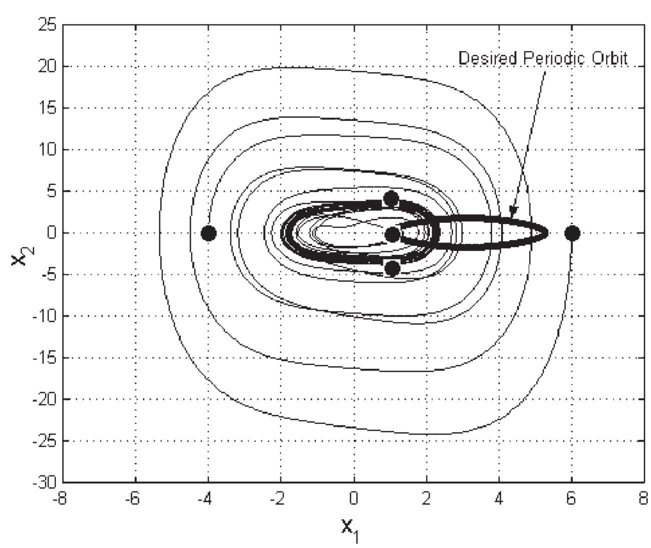

Figure 2. The behaviour of the uncontrolled Duffing oscillator for five different initial conditions and the desired periodic orbit.

where T_s is the sampling period, and Λ_1 and Λ_2 are positive valued design parameters. The continuous time representation of the enforced reaching law with (15) becomes $\dot{s} = -\Lambda_1 \operatorname{sgn}(s) - \Lambda_2 s$, which corresponds to constant plus proportional rate reaching law of SMC terminology [8]. In the simulations, we set $\Lambda_1 = \Lambda_2 = 5$ with $T_s = 2.5$ ms satisfying the condition $1 - \Lambda_2 T_s > 0$. Furthermore, in order to reduce the effect of chattering phenomenon, we use the approximation $\operatorname{sgn}(s_n) \approx s_n/(|s_n| + \delta)$ with $\delta = 0.05$. The vector characterizing the sliding motion is set as $\underline{\alpha} = [1\ 1]^{\mathrm{T}}$.

Under these conditions, according to the feedback control structure depicted in Figure 1, the states of the oscillator are measured and corrupted by Gaussian distributed, zero mean noise sequences. Both noise sequences are in ± 0.001 interval in magnitude, and have variances equal to $6\mathrm{e}{-8}$.

Initial parameters of the neurocontroller have been set randomly to numbers from the interval [0, 0.1] and the same initial parameters have been used in all trials presented in this paper. The initial value of the adjustable parameter vector norm ($\sqrt{\underline{\omega}_0^{\mathrm{T}} \underline{\omega}_0}$) is calculated to be 0.2522. The controller has the structure 2-6-1 with hyperbolic tangent functions in the neuronal

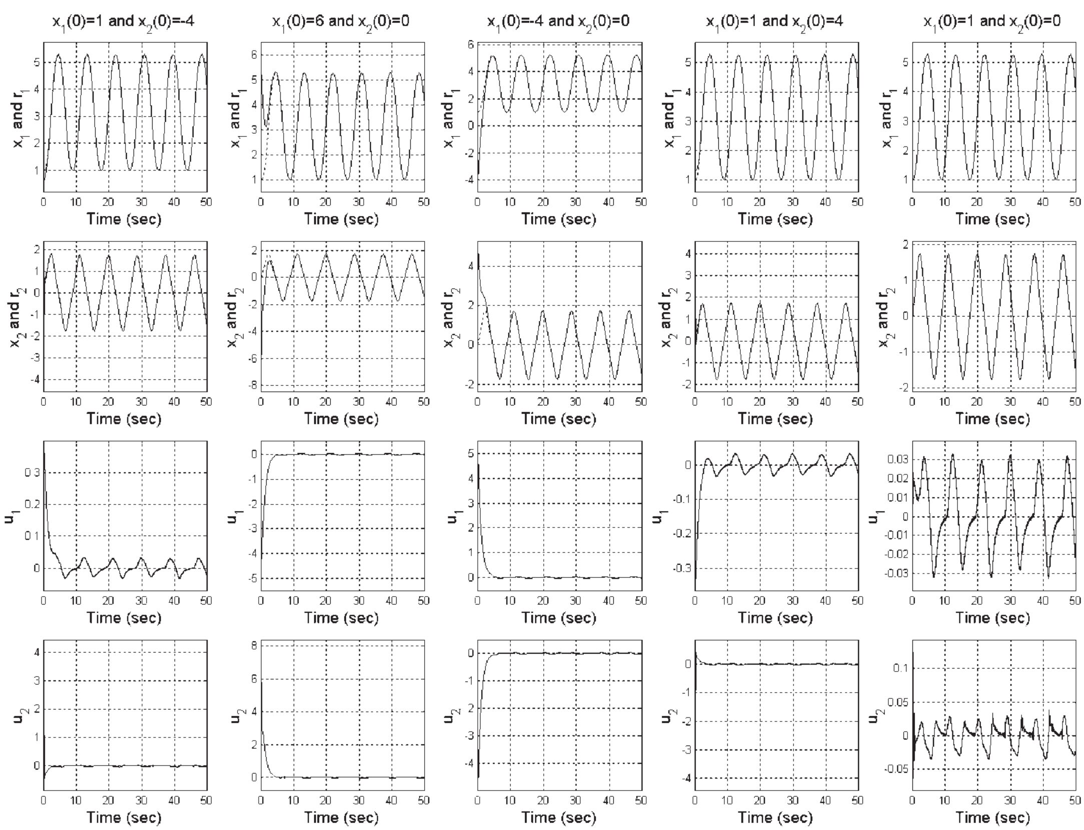

Figure 3. Simulation results.

activation of the six hidden neurons. With the chosen neural controller $N = 25$, $\underline{u}_n = [(r(t) - x(t)) \;\; (\dot{r}(t) - \dot{x}(t))]^{\mathrm{T}}\big|_{t=nT_s}$ and we set $\mu = 10$ for the LM update strategy.

Under the conditions described above, five sets of simulations have been performed, and in each case the initial conditions of the Duffing oscillator have been set to different values. The results are depicted in Figures 3 and 4. The subplots on each column illustrate the results of an individual trial. In Figure 3, the desired and observed states of the system and the tracking errors are depicted. Clearly, the error dynamics has a convergent characteristic in each case. Figure 4 illustrates the extracted control signal error (s_{Cn}), the applied control signal (τ), the evolution of the norm of the parameter vector $\left(\sqrt{\underline{\omega}_n^{\mathrm{T}}\underline{\omega}_n}\right)$ in logarithmic time axis and the behaviour in the phase space for each case. It is apparent that the extracted control signal produced displays a smooth characteristic, which is an important feature recommending the algorithm for nonlinear control application. From the third row of Figure 2, it can be claimed that the parameters evolve bounded, which indicates that the parametric stability (or the internal stability of the training dynamics) has been achieved. The last row of Figure 2 demonstrates the proof of the concept introduced. The error vector hits the sliding line characterized by $\underline{\alpha} = [1 \;\; 1]^{\mathrm{T}}$ (or equivalently $\mathrm{d}(r-x)/\mathrm{d}t = -(r-x)$) and moves towards the origin as imposed by the differential dynamics above.

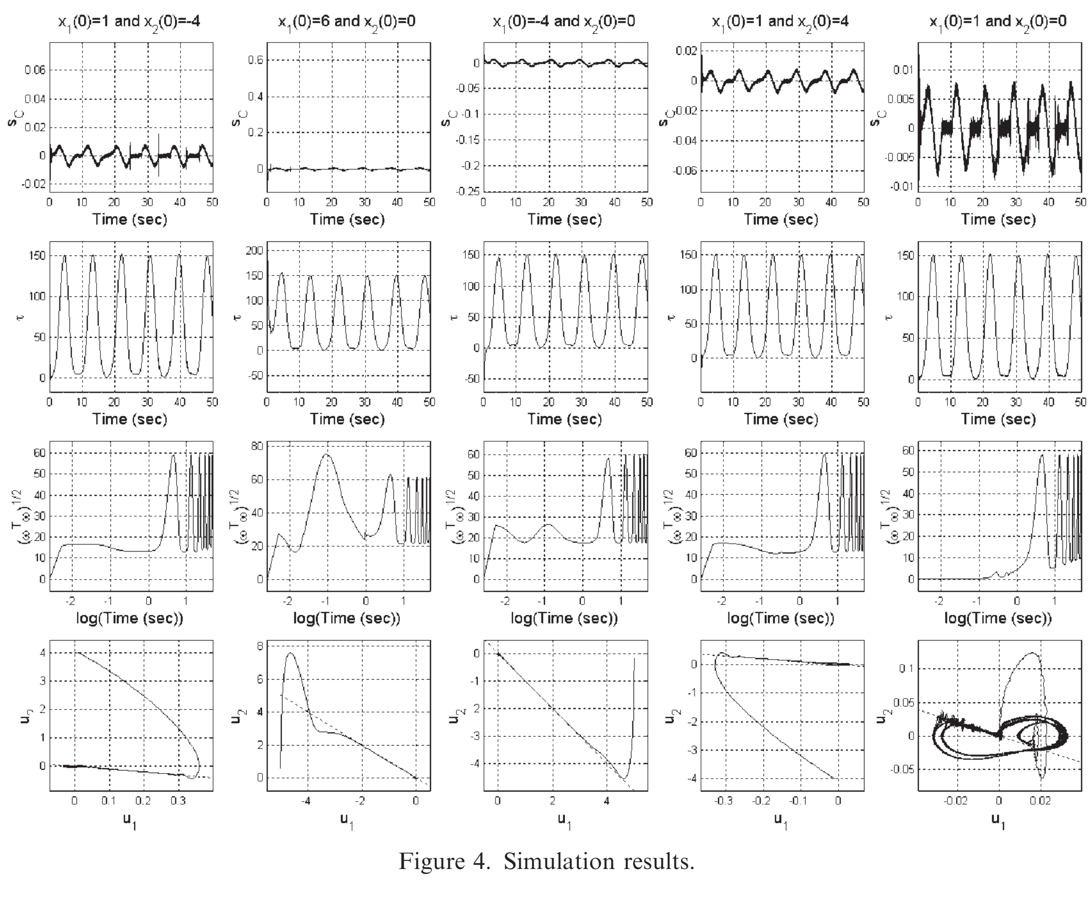

Figure 4. Simulation results.

Lastly, the computational aspects should be assessed. For the given neurocontroller structure (i.e. $N = 25$), a single forward pass for control calculation and a single backward pass for parameter tuning requires 38,408 floating point operations. This fact constitutes a drawback, however, it is still affordable for today's high speed microprocessors.

5. CONCLUSIONS

This paper describes a task specific calculation of an error critic for intelligent control systems. A simple feedforward neural network structure has been used as the controller. The parameters of the controller have been tuned by LM optimization technique. The approach addresses the use of the devised tuning scheme for the sliding mode control of an unknown non-linear system in discrete time basis. The method introduced is in good compliance with the objectives of the control engineering. It has been observed that the proposed scheme is capable of alleviating the uncertainties and compensating the disturbances. The results observed justify the analytical claims of the paper and the prescribed design specifications have been met through the use of the strategy discussed.

ACKNOWLEDGEMENT

This work is supported by NSF (Grant No: 9906233).

REFERENCES

1. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In Rumelhart DE and McClelland JL (eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol.1. MIT Press: Cambridge 1986; 318–362.
2. Haykin S. Neural Networks. Macmillan College Printing Company: NJ, 1994.
3. Andersen TJ, Wilamowski BM. A modified regression algorithm for fast one layer neural network training. *World Congress of Neural Networks*, vol.1. Washington DC, USA; 1995; 687–680.
4. Battiti R. First-second-order methods for learning: between steepest descent and Newton's method. *Neural Computation* 1992; **4**:141–166.
5. Chalalombous C. Conjugate gradient algorithm for efficient training of artificial neural networks. *IEE Proceedings* 1992; **139**:301–310.
6. Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. *IEEE Transaction on Neural Networks* 1994; **NN-5**:989–993.
7. Shah S, Palmieri F. MEKA-A fast, local algorithm for training feedforward neural networks. *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, USA, 1990; **3**:41–46.
8. Hung JY, Gao W, Hung JC. Variable structure control: a survey. *IEEE Transactions on Industrial Electronics*, 1993; **IE-40**:2–22.
9. Kaynak O, Erbatur K, Ertugrul M. The fusion of computationally intelligent methodologies and sliding-mode control—a survey. *IEEE Trans. on Industrial Electronics* 2001; **IE-48**:4–17.
10. Young KD, Utkin VI, Ozguner U. A control engineer's guide to sliding mode control. *IEEE Transaction on Control Systems Technology* 1999; **CST-7**:328–342.
11. Sarpturk SZ, Istefanopulos Y, Kaynak O. On the stability of discrete-time sliding mode control systems. *IEEE Transactions on Automatic Control* 1987; **AC-32**:930–932.
12. Gao W, Wang Y, Homaifa A. Discrete-time variable structure control systems. *IEEE Transaction on Industrial Electronics* 1995; **IE-42**:117–122.
13. Sira-Ramirez H. Non-linear discrete variable structure systems in quasi-sliding mode. *International Journal of Control* 1991; **54**:1171–1187.
14. Sira-Ramirez H, Colina-Morles E. A sliding mode strategy for adaptive learning in adalines. *IEEE Transaction on Circuits and Systems—I: Fundamental Theory and Applications*, 1995; **CS-I-42**:1001–1012.

15. Parma GG, Menezes BR, Braga AP. Sliding mode algorithm for training multilayer artificial neural networks. *Electronics Letters* 1998; **34**:97–98.
16. Chen X, Fukuda T. Computer-controlled continuous-time variable structure systems with sliding modes. *International Journal of Control* 1997; **67**:619–639.
17. Muñoz D, Sbarbaro D. An adaptive sliding-mode controller for discrete nonlinear systems. *IEEE Transaction on Industrial Electronics* 2000; **IE-47**:574–581.
18. Poznyak AS, Yu W, Sanchez EN. Identification and control of unknown chaotic systems via dynamic neural networks. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications* 1999; **CST-I-46**:1491–1495.

