Penn State ILL Lending - ARTICLE

ILLiad TN: 364490

Call #: QA76.87 .N483

Location: EG

Journal Title: Neural computing & applications.

Volume: 13
Issue: 3
Month/Year: September 2004
Pages: 211-220

Article Author:

Article Title: M. O. EFE;

Notes: Borrowing Notes;

From:
Penn State Interlibrary Loan Lending
127 Paterno Library, Curtin Road
University Park, PA 16802

Fax: 814-863-2653
Ariel: 128.118.152.188

1/25/2005

Request in processing: 20050124

AKM (11p)

ILL# 1911133

Ariel: 128.146.9.51

Patron Kiziltas, Gullu

OSU

CIC Lanter

MAXCOST:$25 IFM

Ship To:
Interlibrary Loan
Ohio State University
1858 Neil Avenue Mall
Columbus OH 43210

Fax: (614)292-3061

Neural Comput & Applic (2004) 13: 211–220
DOI 10.1007/s00521-004-0406-3

ORIGINAL ARTICLE

Mehmet Önder Efe

Discrete time neuro sliding mode control with a task-specific output error

Received: 22 October 2001 / Accepted: 1 March 2004 / Published online: 26 May 2004

Abstract The problem of obtaining the error at the output of a neuro sliding mode controller is analyzed in this paper. The controller operates in discrete time and the method presented describes an error measure that can be used if the task to be achieved is to drive the system under control to a predefined sliding regime. Once the task-specific output error is calculated, the neurocontroller parameters can be tuned so that the task is achieved. The paper postulates the strategy for discrete time representation of uncertain nonlinear systems belonging to a particular class. The performance of the proposed technique has been clarified on a third order nonlinear system, and the parameters of the controller are adjusted by using the error backpropagation algorithm. It is observed that the prescribed behavior can be achieved with a simple network configuration.

Keywords Backpropagation training · Control error extraction · Discrete time sliding mode control · Nonlinear control · Neural networks · Neurocontrol

1 Introduction

Neural networks (NNs) have successfully been used for many purposes extending from image processing and pattern recognition to identification and control of systems. The motivation encouraging the use of NNs in such a wide spectrum of applications has mainly been the ability to represent complex nonlinear mappings, and the learning and generalization of data together with powerful training strategies and anticipatory behavior. Furthermore, the architectural diversity of NNs has constituted an advantage exploited to find the best structure for the problem in hand. The practice of systems and control engineering has therefore extensively benefited from the design alternatives provided through the use of NNs.

Among many strategies existing for parameter tuning, the error backpropagation (EBP) method has been a standard approach in most applications [1, 2, 3, 4, 5]. When the target values of the desired map are given, EBP is a good start for NN training, yet the cases in controller design are not so trivial. The common problem in the neurocontroller training is the unavailability of the error on the applied control signal [6]. In other words, to be able to modify the weights of a neurocontroller, there has to be a measure of error at the output of the controller. In the literature, one approach is to identify the plant and to propagate the output error back through the identifier until the controller outputs are reached [3, 4]. When the output error is obtained, the controller parameters can be tuned by the EBP technique so that a specified task is fulfilled. Some practical drawbacks are the increase in the computational burden due to the identification process and the problems associated with persistent excitation. In this paper, we derive the error measure for the discrete time sliding mode control (DTSMC) task for a class of uncertain nonlinear systems. The contribution of this paper to the literature is the introduction of a new error measure that:

- can be used for the training of the feedback neurocontroller online,
- can be used for all sorts of supervised training schemes
- does not necessitate the exact knowledge of the plant dynamics,
- drives the plant dynamics into the desired sliding regime and
- quantifies the control signal according to the enforced task, which is the DTSMC regime.

The design of sliding mode controllers is a well-developed framework especially for systems represented

M. Ö. Efe
Department of Mechatronics Engineering,
Atilim University, Incek, TR-06836 Ankara, Turkey
E-mail: onderefe@ieee.org
Tel.: +90-532-3941081

in continuous time. The approach has been utilized particularly in the applications where there are strong interdependencies between nonlinearities, time varying parameters, time delays and noise [7, 8]. In the literature, various techniques towards the integration of sliding mode control with NNs have been presented [9, 10, 11]. The design issues for conventional DTSMC have later been addressed in [12], which scrutinizes the design of DTSMC with particular emphasis on reaching the law approach, and exemplify the results on a second order linear system having uncertain parameters. One of the notable works discussing the stability issues in DTSMC is presented in [13], in which the sufficient conditions for convergence are discussed. Pieper et al. [14] analyze the optimality in DTSMC from the point of designing optimal sliding surfaces with a linear quadratic criterion, and confirm the results on a gantry crane apparatus. Sira-Ramirez [15] discusses the convergence during quasi-sliding mode for nonlinear single input single output (SISO) systems, and Chen et al. elaborate the sampling time selection problem in computer controlled systems with a sliding mode [16]. In [17], Misawa analyzes the construction of DTSMC under the presence of unmatched uncertainties. One of the recent studies in DTSMC formulates a recursive control signal for linear systems and proves that the state of the system is uniformly ultimately bounded in the presence of time-varying disturbance and uncertainties [18]. In [19], the DTSMC task is studied for discrete time input-output models, and in [20], the design based on the Euler discretization is analyzed. Aside from the approaches analyzing the state space representations, a number of studies have demonstrated that NNs can successfully be used in DTSMC systems [21, 22, 23].

In what follows, we describe the task and the proposed technique for control error extraction, which is the primary difficulty in most intelligent control systems. The third section describes the plant on which the performance of the scheme is visualized, and presents the simulation results. The concluding remarks are given at the end of the paper.

2 Task definition and the control error extraction

Consider the control system structure depicted in Fig. 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 (ZOHs) as shown in the figure, and the technique yet to be discussed enjoys the discrete nature of the data. Note that the subscript k stands for the discrete time index, and the dynamics inside the dashed rectangle is governed by a set of difference equations of the form given below.

$$\underline{x}_{k+1} = \underline{f}(\underline{x}_k) + \underline{g}(k)\, u_k \qquad (1)$$

where $\underline{x}_k = [x_{1k} \quad x_{2k} \quad \cdots \quad x_{nk}]^T$ is the state vector $\underline{f}(\underline{x}_k)$, is a nonlinear vector function of the system state and is unavailable, whereas $g(k)$ is a vector function of time and the sign of it is known. The abovementioned system can compactly be written as

$$\underline{x}_{k+1} = \underline{f}_k + \underline{g}_k\, u_k$$

According to Fig. 1, the error vector at time k is defined as $\underline{e}_k = \underline{x}_k$-$\underline{r}_k$, where $\underline{r}_k$ is the vector of reference state trajectories at time k. Define the switching function as

$$s_k = \underline{\alpha}^T \underline{e}_k \qquad (2)$$

in which the vectorα is selected such that the dynamics determined by $s_k = 0$ is stable, and it is assumed that $\underline{\alpha}^T$ $g_k > 0$. Now adopt a closed loop switching dynamics described generically as $s_{k+1} = Q(s_k)$, and evaluate s_{k+1} as given below.

$$s_{k+1} = \underline{\alpha}^T\left(\underline{f}_k + \underline{g}_k u_k - \underline{r}_{k+1}\right) \qquad (3)$$

Using $s_{k+1} = Q(s_k)$, and solving for u_k gives the control sequence formulated as follows.

$$u_k = -\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}\left(\underline{\alpha}^T\left(\underline{f}_k - \underline{r}_{k+1}\right) - Q^i(s_k^i)\right) \qquad (4)$$

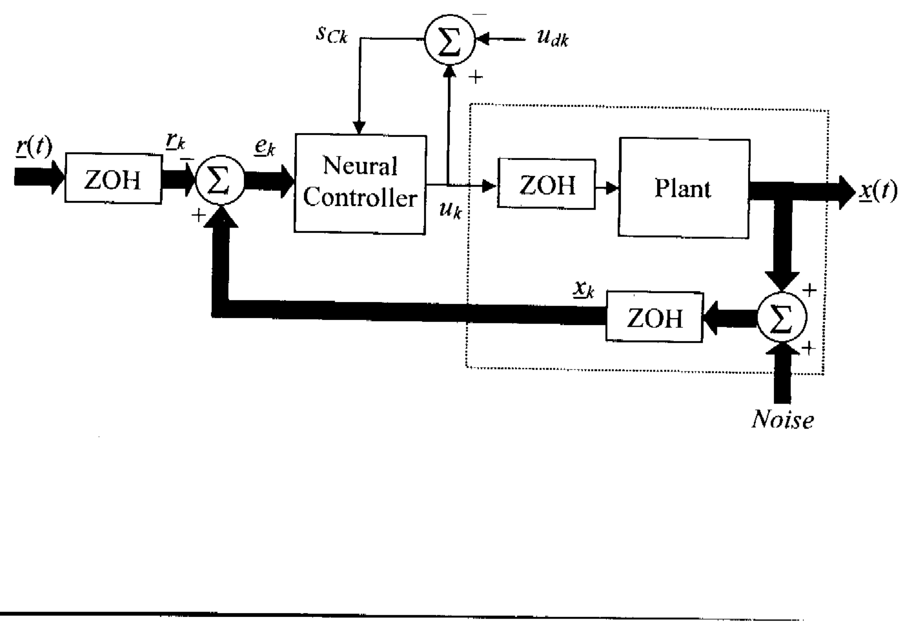


Fig. 1 The structure of the feedback control system

If the values of the vector functions f_k and g_k were known explicitly, the application of this sequence to the system of Eq. 1 would result in $s_{k+1} = Q(s_k)$, where Q must satisfy the condition below to ensure reaching [12, 13, 15, 16].

$$s_k(s_{k+1} - s_k) = s_k(Q(s_k) - s_k) < 0 \tag{5}$$

If the abovementioned condition is satisfied for $\forall k \geq 0$, the system is driven towards the dynamics characterized by $s_k = 0$. However, in practice, $s_k = 0$ is rarely observed as the problem is described in discrete time. A realistic observation is $|s_k| < \epsilon$, where ϵ is some positive number. In the literature, this phenomenon is called the quasi-sliding mode, or equivalently the pseudo-sliding mode [12, 16, 23]. 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 prescribed by $|s_k| < \epsilon$.

2.1 The calculation of the task specific controller output error

Define the task as the DTSMC of a plant of the form given in Eq. 1, whose ultimate behavior is to be enforced towards what is prescribed by $s_{k+1} = Q(s_k)$. Consider Fig. 1, which demonstrates that the quantity s_{Ck} would be the error on the applied control signal if we had a supervisor providing the desired value of the control denoted by u_{dk}. 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_{Ck} from the available quantities. In what follows, we present a method to extract the error on the control signal.

Assumption 2.1: The vector functions f_k and g_k of Eq. 1 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 2.2: A control sequence leading to the desired DTSMC can be formulated if the dynamics of the system described by Eq. 1 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 [7]. When the control sequence in Eq. 4 is applied to the system of Eq. 1, we call the resulting behavior as the *target DTSMC* and the input signal leading to it as the *target control sequence* (u_k). If at least the explicit forms of the nominal representations of the vector functions f_k and g_k 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 2.3: Given an uncertain plant, which has the structure described as in Eq. 1, and a command trajectory $\underline{r}_k$ for $k \geq 0$, the input sequence denoted by u_{dk} 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, $\underline{r}_k = [r_{1k} \;\; r_{2k} \;\; \cdots \;\; r_{nk}]^T$ stands for the vector of command trajectories.

$$\underline{r}_{k+1} = \underline{f}(\underline{r}_k) + \underline{g}(k)\, u_{dk} \tag{6}$$

Mathematically, the existence of such a model and the sequence means that the system of Eq. 1 perfectly follows the command trajectory ($\underline{r}_k$) if both the idealized control sequence (u_{dk}) is known and the initial conditions are set as $\underline{x}_0 = \underline{r}_0$; more explicitly $\underline{e}_k \equiv 0$ for $\forall k \geq 0$. Undoubtedly, the reference DTSMC model is an abstraction as the functions appearing in it are not available. However, the concept of an idealized control sequence should be viewed as the synthesis of the command signal $\underline{r}_k$ from the time solution of the difference equation in Eq. 6.

Fact 2.4: If the target control sequence formulated in Eq. 4 were applied to the system of Eq. 1, the idealized control sequence would be the steady state solution of the control signal, i.e. $\lim_{k\to\infty} u_k = u_{dk}$. However, under the assumption of the achievability of the DTSMC task, the difficulty here again is the unavailability of the functional forms of f_k and g_k. Therefore, the aim in this subsection 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 $s_{Ck} = u_k - u_{dk}$. If the target control sequence of Eq. 4 is rewritten by using Eq. 6, one gets

$$\begin{aligned} u_k &= -\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}\left(\underline{\alpha}^T\left(\underline{f}(\underline{x}_k) - \underline{f}(\underline{r}_k) - \underline{g}_k u_{dk}\right) - Q(s_k)\right) \\ &= -\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}\left(\underline{\alpha}^T\left(\underline{f}(\underline{x}_k) - \underline{f}(\underline{r}_k)\right) - Q(s_k)\right) + u_{dk} \\ &= -\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}\left(\underline{\alpha}^T \underline{\Delta f}_k - Q(s_k)\right) + u_{dk} \end{aligned} \tag{7}$$

where $\underline{\Delta f}_k = \underline{f}(\underline{x}_k) - \underline{f}(\underline{r}_k)$. The target control sequence becomes identical to the idealized control sequence, i.e. $u_k \equiv u_{dk}$ as long as $\underline{\alpha}^T \underline{\Delta f}_k - Q(s_k) = 0$ holds true for $\forall k \geq 0$. However, this condition is of no practical importance as the analytic form of the function f_k 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 $s_{Ck} = 0$, 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_{k+1} given below.

$$
\begin{aligned}
s_{k+1} &= \underline{\alpha}^T(\underline{x}_{k+1} - \underline{r}_{k+1}) \\
&= \underline{\alpha}^T\left(\underline{f}(\underline{x}_k) + \underline{g}_k u_k - \underline{f}(\underline{r}_k) - \underline{g}_k u_{dk}\right) \\
&= \underline{\alpha}^T\left(\underline{\Delta f}_k + \underline{g}_k s_{Ck}\right) \\
&= Q(s_k) + \underline{\alpha}^T \underline{g}_k s_{Ck}
\end{aligned}
\tag{8}
$$

Solving the above equation for s_{Ck} yields the following

$$
s_{Ck} = \left(\underline{\alpha}^T \underline{g}_k\right)^{-1}(s_{k+1} - Q(s_k))
\tag{9}
$$

The interpretation of the abovementioned control error measure is as follows: since we are in pursuit of enforcing $s_{k+1} = Q(s_k)$ 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 Eq. 9 to approach zero, this enforces $\underline{\alpha}^T\underline{\Delta f}_k - Q(s_k) = 0$ to approach zero, i.e. $s_{k+1} \rightarrow Q(s_k)$, consequently $u_k \rightarrow u_{dk}$ as k increases.

Remark 2.5: Notice that the application of u_{dk} for $\forall k \geq 0$ to the system of Eq. 1 with zero initial errors will lead to $\underline{e}_k \equiv 0$ for $\forall k \geq 0$. On the other hand, the application of u_k for $\forall k \geq 0$ to the system of Eq. 1 will lead to $s_k = 0$ for $\forall k \geq k_h$, where k_h is the hitting time index, at which the quasi-sliding regime starts. Therefore, the adoption of Eq. 9 as the equivalent measure of the control error loosens $\underline{e}_k \equiv 0$ for $\forall k \geq 0$ requirement and enforces $s_{k+1} \rightarrow Q(s_k)$. Consequently, the tendency of the control scheme will be to generate the target DTSMC sequence u_k of Eq. 4.

Remark 2.6: Referring to Eq. 9, it should be obvious that if $s_{Ck}(s_{Ck+1} - s_{Ck}) < 0$ is satisfied $s_k(s_{k+1} - s_k) < 0$ 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.

Proposition 2.7: Since $s_{Ck} = u_k - u_{dk}$, the cost at each instant of time can be defined as

$$
J_{Ck} = \frac{1}{2} s_{Ck}^2
\tag{10}
$$

which instantly qualifies the similarity between u_k and u_{dk}. If the parameters of the neurocontroller are tuned such that the cost in Eq. 10 is enforced toward zero, the task implied by $s_{Ck} = 0$ is achieved. More explicitly, a system of structure Eq. 1 in the feedback loop illustrated in Fig. 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 Eq. 10.

In what follows, we describe the structure of the controller and the chosen tuning scheme together with the relevance with what have been derived so far.

2.2 The neurocontroller and the tuning scheme

In the analysis presented so far, we have described the task and the analytic representation of the error to be used in training. Although the presented approach is applicable to any NN model, in this study, we consider the feedforward NN structure because of its widespread use.

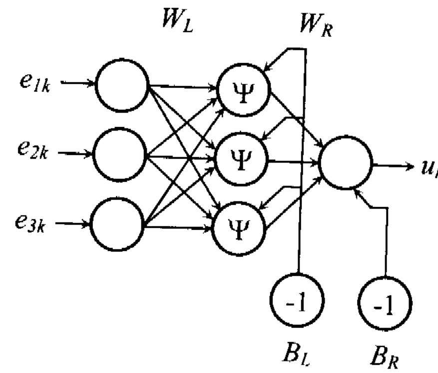


Fig. 2 The structure of a feedforward NN

The neurocontroller utilized in this paper has the architecture and input output definitions as depicted in Fig. 2. In Eq. 11, the mathematical representation for such a three-layered NN is given.

$$
u_k = W_R^T \Psi\left(W_L^T \underline{e}_k - B_L\right) - B_R
\tag{11}
$$

In the abovementioned formula, W_R and W_L are weight matrices, B_L and B_R are the bias vectors and Ψ is the nonlinear activation function of the neurons contained in the hidden layer. The activation function of the output layer neuron is linear. Among many alternatives existing in the literature, we choose the hyperbolic tangent function for the neurons in the hidden layer.

The parameter tuning can be done by using EBP technique as well as higher order methods, e.g., the Levenberg-Marquardt optimization method, the Gauss-Newton algorithm or conjugate gradients [24]. In order to demonstrate the viability of the extracted error measure, we use the EBP technique for parameter adjustment. According to the EBP-based tuning strategy, in order to minimize the cost of Eq. 10, if ϕ is defined to be a generic adjustable parameter of the neurocontroller, the adjustment of ϕ is carried out by the rule given as

$$
\begin{aligned}
\phi_{k+1} &= \phi_k - \eta \frac{\partial J_{Ck}}{\partial \phi_k} \\
&= \phi_k - \eta s_{Ck} \frac{\partial s_{Ck}}{\partial \phi_k} \\
&= \phi_k - \eta s_{Ck} \frac{\partial (u_k - u_{dk})}{\partial \phi_k} \\
&= \phi_k - \eta s_{Ck} \frac{\partial u_k}{\partial \phi_k}
\end{aligned}
\tag{12}
$$

where η is the learning rate chosen from the interval (0,1) and J_{Ck} and s_{Ck} are defined as in Eq. 10 and Eq. 9, respectively. It is apparent that $J_{Ck} \rightarrow 0$ means $s_{Ck} \rightarrow 0$, hence $u_k \rightarrow u_{dk}$. The update rule of Eq. 12 is applied to all entries of W_R, W_L, B_L and B_R at each sampling instant.

Note that we assumed $\underline{\alpha}^T \underline{g}_k > 0$. With s_{Ck} of Eq. 9, the rule in Eq. 12 becomes

$$\phi_{k+1} = \phi_k - \eta\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}(s_{k+1} - Q(s_k))\frac{\partial u_k}{\partial \phi_k}$$

in which we can set $\zeta = \eta\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}$ and choose the value of ζ as g_k is unknown.

2.3 Practical Issues

The analysis presented so far has concentrated on the class of systems having the structure described in Eq. 1. It should be obvious that the system under control in real life will be a sampled form of a continuous system, which can generically be represented as

$$\dot{x} = \underline{\sigma}(\underline{x}, u) \tag{13}$$

The abovementioned system can be viewed as the plant block in Fig. 1. Consequently, the system of Eq. 1 will correspond to the sampled system inside the dashed rectangle of the figure.

2.3.1 i. The sampling time

Since the design presented is based on the discrete time representation of a continuous time system, the selection of the sampling time gains a substantial importance. We assume that the sampling period is small enough so that the response of the discrete time system matches sufficiently to that of the continuous time system. Furthermore, the dicretized form of the system belongs to the class described in Eq. 1.

2.3.2 ii. Causality

In Eq. 9, we have postulated the error on the applied control at time k. However, the right hand side of Eq. 9 requires the value of s_{k+1}. In the application example, we set $s_{Ck} = \left(\underline{\alpha}^T \underline{g}_{k-1}\right)^{-1}(s_k - Q(s_{k-1}))$, the right hand side of which is actually the control error at time k-1. Assuming this form as a practically equivalent measure of the control error, we introduce some amount of uncertainty into the control system, which can be represented in the system dynamics that has already been assumed to be unknown.

2.3.3 iii. The actuation speed

Another important issue is the actuation speed of the system under control, i.e., the ability to respond to what is imposed in a timely fashion. Since we assume that the details concerning the dynamic model of the system are unavailable, what causes a difficulty from a practical point of view is the selection of $s_{k+1} = Q(s_k)$, which characterizes the behavior during the reaching mode. The parameters of this quantity can only be set by trial-and-error due to the lack of system-specific details.

In the application example, we utilize $s_{k+1} = (1-\lambda_1 T_s)s_k - \lambda_2 T_s \text{sgn}(s_k)$, where $\lambda_1 > 0$, $\lambda_2 > 0$ and $1-\lambda_1 T_s > 0$ with T_s being the sampling period [12].

2.3.4 iv. An enhancement of the behavior in the quasi-sliding mode

It is a well known fact that the use of the sgn(.) function, particularly during the sliding mode for continuous time variable structure control systems, affects the performance during the sliding mode adversely as the measured quantity is very close to zero, and this leads to the chattering phenomenon [25]. However, in discrete time, once the trajectory in the phase space crosses the switching hyperplane, it maintains the crossings repetitively and a zigzag motion along the switching hyperplane occurs [12]. Although the stability requirements ensure that the magnitude of the zigzagging motion is bounded, adopting a smooth transition about the decision boundary can enhance the tracking performance in terms of reducing the magnitude of the zigzagging during the sliding mode. For this purpose, we adopt the following approximation for the sgn(.) function.

$$\text{sgn}(s_{Ck}) \cong \frac{s_{Ck}}{|s_{Ck}| + \delta} \tag{14}$$

where, δ determines the sharpness around the origin. Since the function in Eq. 14 is not discontinuous at the origin, the decision mechanism provides a soft switching in the vicinity of the boundary characterized by $s_{Ck} = 0$.

3 The dynamics of the plant under control and simulation results

In this section, we demonstrate the performance of the algorithm on a third order system studied previously in [26, 27]. The continuous time dynamic equation describing the system is given in Eq. 15-17. Clearly, the system will be of structure described in Eq. 1 when discretized by using the Euler method.

$$\dot{x}_1 = x_2 + d_1 \tag{15}$$

$$\dot{x}_2 = x_3 + d_2 \tag{16}$$

$$\begin{aligned} \dot{x}_3 = {} & -0.5x_1 - 0.5x_2^3 - 0.5x_3|x_3| + \left(1 + 0.1\sin\left(\frac{\pi t}{3}\right)\right)u \\ & + d_3 + d_4 + (-0.05 + 0.25\sin(5\pi t))x_1 \\ & + (-0.03 + 0.3\cos(5\pi t))x_2^3 \\ & + (-0.05 + 0.25\sin(7\pi t))x_3|x_3| \end{aligned} \tag{17}$$

where $d_4(t) = 0.2\sin(4\pi t)$ is the disturbance used in [26, 27], and $d_i(t)$ with $i = 1,2,3$ are the Gaussian noise sequences corrupting the state information to be used by the neurocontroller additively. The mean and variance of each noise sequence are equal to zero and $0.33{*}10^{-7}$, respectively. Furthermore, $|d_i(t)| \leq 0.007$ with a probability very close to unity. The work presented in [26] assumes that the nominal system dynamics is known and the uncertain part is comprised of what we give as the last three terms in Eq. 17. The primary difference between what we discuss and what is assumed in [26] is that the approach we propose assumes only the achievability of the DTSMC task; hence, the uncertainties are represented in the system dynamics, whose form is known but the details are not.

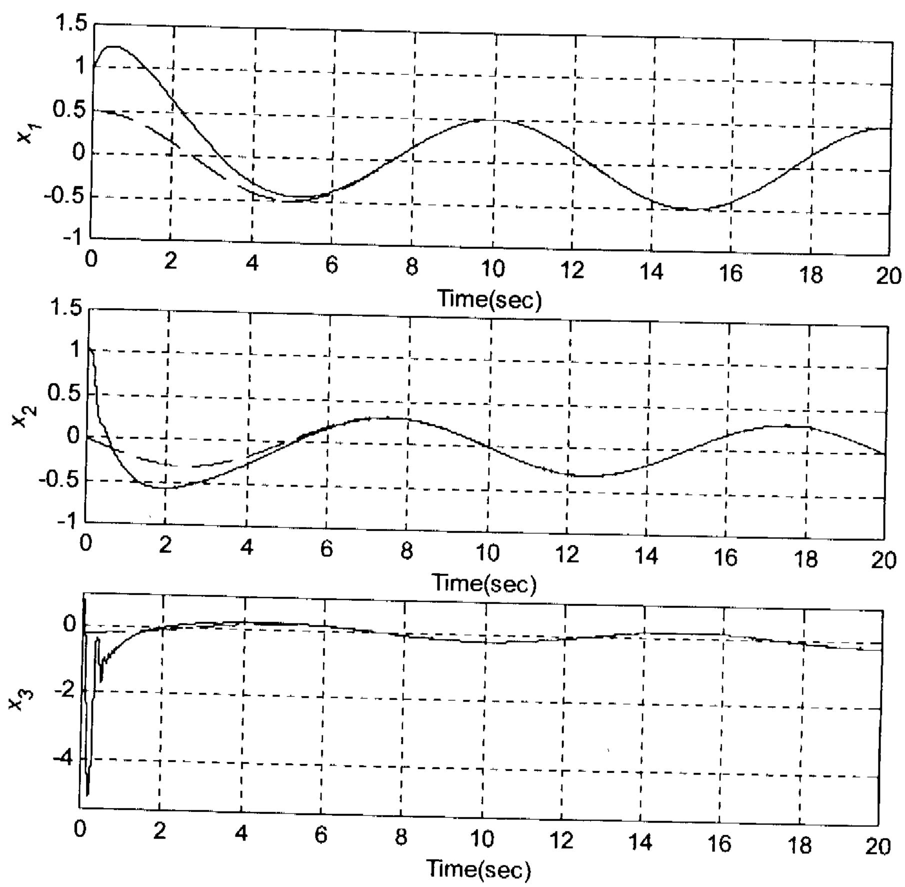
The sampling time has been set as $T_s = 2.5$msec, the switching function parameters have been selected as $\underline{\alpha} = [1\ 2\ 1]^T$ and the parameters of the reaching law are chosen as $\lambda_1 = 380$ and $\lambda_2 = 1$. The neurocontroller possesses three inputs, a single hidden layer containing three neurons and a single output neuron. Initially, the weights and the biases of the network have been chosen randomly from the interval [0, 0.1]. Furthermore, the learning rate of the EBP-based parameter adjustment strategy has been chosen as $\zeta = 0.01$, and we set $\delta = 0.05$ for the sign function smoothing. As δ tends to zero, the adverse effects of the discontinuity at the origin becomes distinguishable. However, with large values of this quantity, Eq. 14 becomes no longer an approximation to the sgn(.) function. A similar tradeoff exists for the selection of the learning rate ζ, whose small values increase the convergence time, whereas the values closer to unity increase the parametric mobility and undesired overshoots become effective. Parallel to [26], the reference state trajectory described in Eq. 18is used in the simulations.

$$
\begin{aligned}
r_1(t) &= 0.5\cos(0.2\pi t) \\
r_2(t) &= -0.1\pi\sin(0.2\pi t) \\
r_3(t) &= -0.02\pi^2\cos(0.2\pi t)
\end{aligned}
\tag{18}
$$

Initially, the states of the system have the following values: $x_1(0) = 1$, $x_2(0) = 1$ and $x_3(0) = 1$. In Fig. 3, the reference state trajectories and the response of the system are illustrated together. Although the tracking performance is clear from Fig. 3, the tracking errors are depicted in Fig. 4, which apparently justifies the truth of the extracted error measure. In order to confirm that the extracted error measure is specific to the described DTSMC task, we figure out the phase space behavior in Fig. 5. The error vector hits the hyperplane several times and moves towards the origin along with the hyperplane. In Fig. 6, the applied control signal is depicted. After an admissibly fast transient lasting approximately 0.9 seconds, the magnitude of the control signal decreases significantly as the reaching phase ends and the sliding regime starts. The lower subplot of Fig. 6 shows that the applied control signal has a feasible characteristic in terms of the duration between consecutive hittings.

Another design issue that should be figured out is the behavior in the adjustable parameter space. Clearly the displacements given to the neurocontroller parameters

Fig. 3 System response



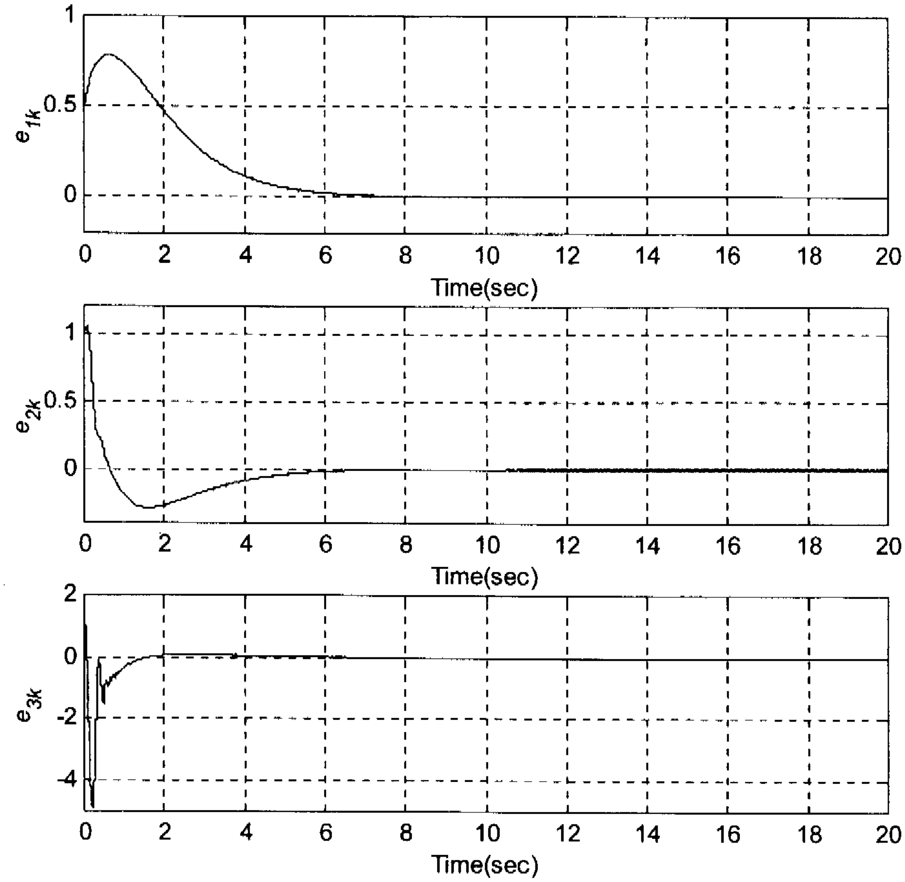


Fig. 4 State tracking errors

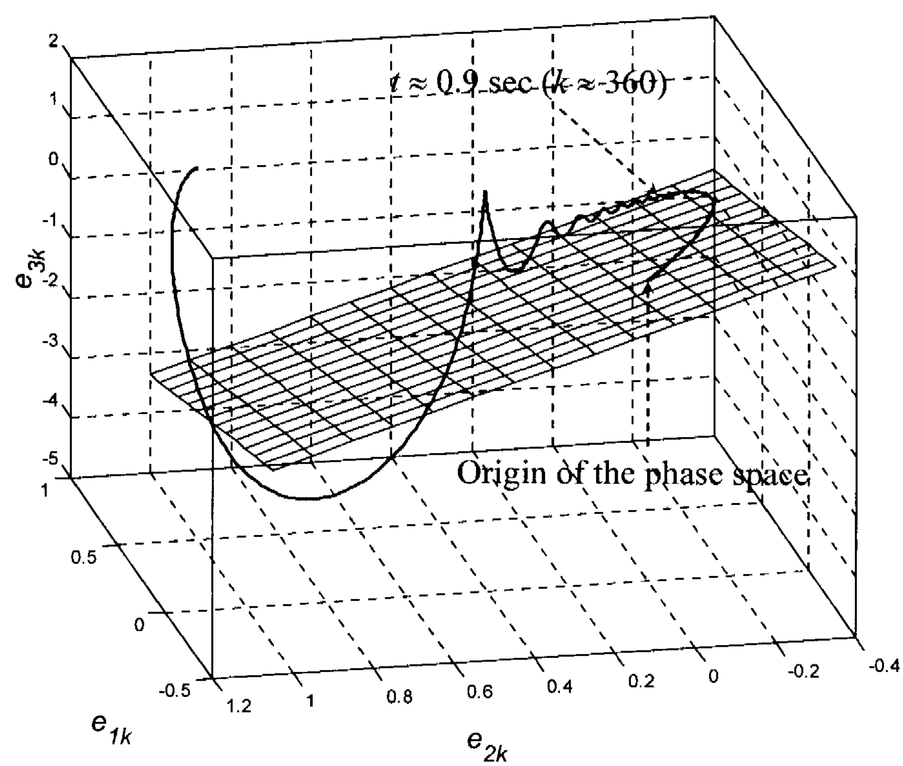


Fig. 5 Behavior in the phase space

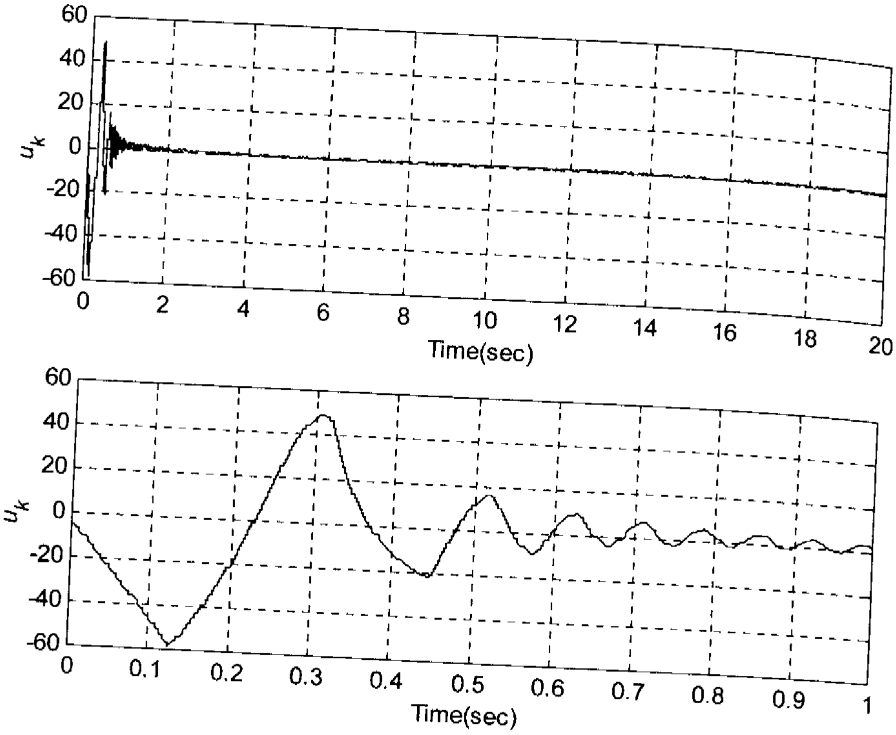


Fig. 6 The applied control signal and its transient behavior

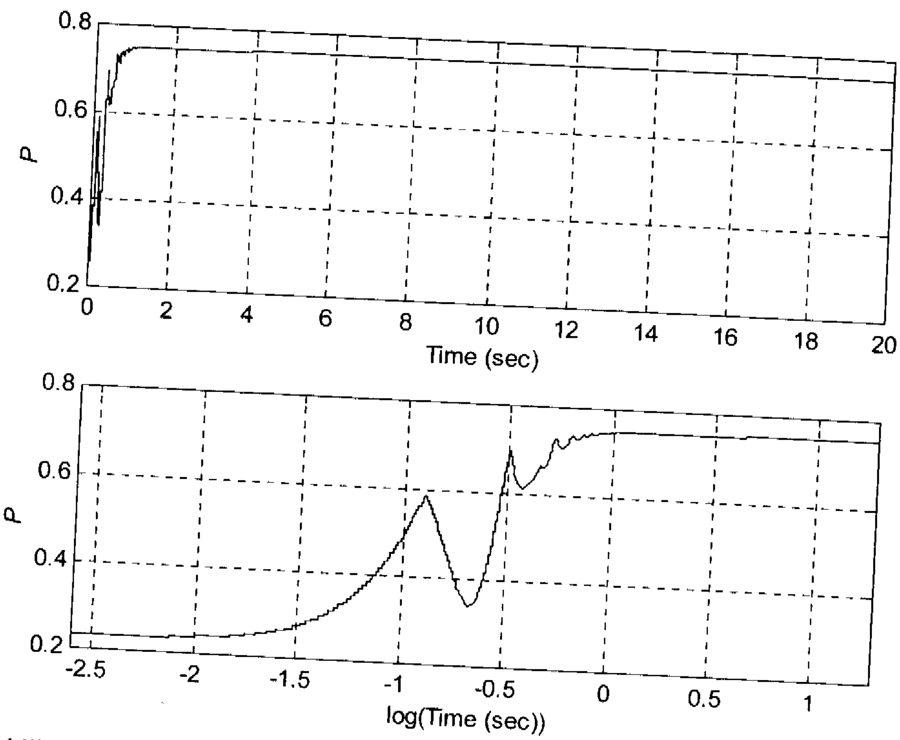


Fig. 7 The behavior of the P measure in linear and logarithmic time axes

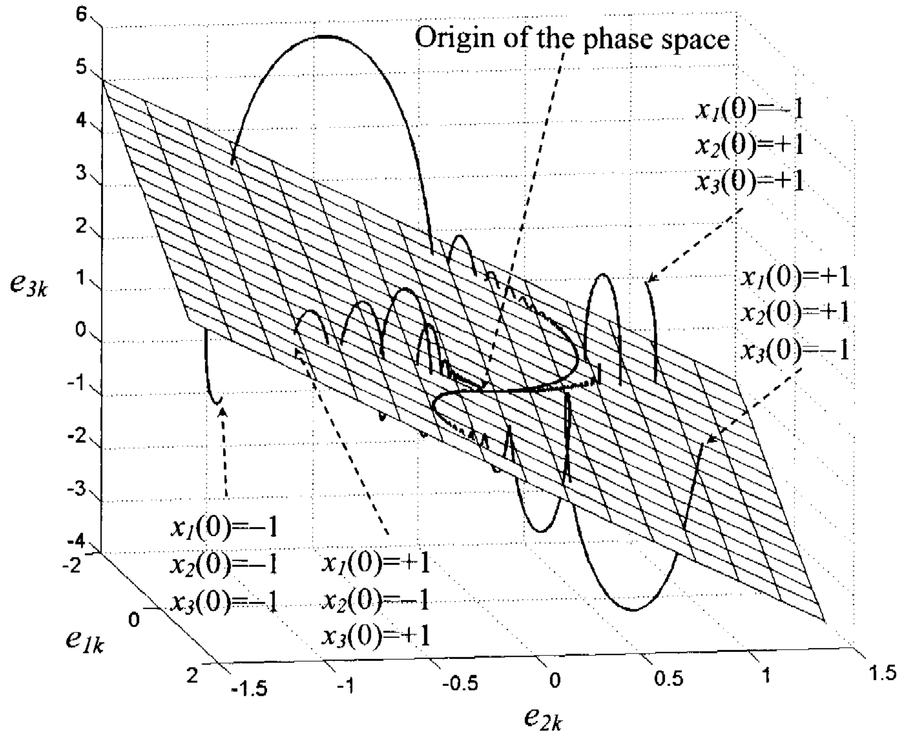
should be convergent to maintain the stability and safety during the training. Assuming that the hidden layer contains H neurons, and defining $\Omega_H = [1\ 1\ 1]^T$, which is of $H{\times}1$ dimensional, we can define the following quantity

$$P = \sqrt{\Omega_H^T W_L^T W_L \Omega_H + B_L^T B_L + W_R^T W_R + B_R^T B_R} \tag{19}$$

which instantly qualifies the behavior of the parametric evolution. In the top row of Fig. 7, the quantity P is illustrated, whereas the bottom row depicts the same quantity on a logarithmic horizontal axis. The figure suggests that the neurocontroller parameters converge to some values, the use of which fulfills the specified DTSMC task.

Lastly, we illustrate the phase space behavior for different initial conditions in Fig. 8. In all four cases, we used the same parameter selections with the same neurocontroller initial parameters. As the reference trajectory, we used the one described in Eq. 18. The results

The behavior in the space for different initial itions



shown justify the claims on creating and maintaining the prescribed DTSMC task.

4 Conclusions

A remedy to the problem of unavailability of the desired outputs of a neurocontroller is studied in this paper. It has been demonstrated that an error measure can be obtained if the plant under control is to be driven towards a predefined quasi-sliding regime. A feedforward NN has been used as the controller and the parameters of it have been adjusted by utilizing the EBP technique. In order to justify the truth of the extracted error measure, a third order system has been considered. The analytic representation of the system is assumed to be unknown, together with the knowledge of its membership to a particular class. Under such an uncertain environment, the results have proven that the prescribed task can be fulfilled together with high tracking precision, convergent evolution in parameter space, robustness against disturbances and above all, with a simple controller structure.

References

1. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. MIT Press, Cambridge, MA
2. Jordan MI, Rumelhart DE (1992) Forward models: supervised learning with a distal teacher. Cog Sci 16:307–354
3. Psaltis D, Sideris A, Yamamura AA (1988) A multilayered neural network controller. IEEE Contr Sys Mag 17–20
4. Narendra KS, Parthasarathy K (1990) Identification and control of dynamical systems using neural networks. IEEE Trans Neur Net 1:4–27
5. Efe MO, Kaynak O (1999) A comparative study of neural network structures in identification of nonlinear systems. Mechatronics 9:287–300
6. Miller WT, Sutton RS, Werbos PJ (1991) Neural networks for control. MIT Press, Cambridge, MA
7. Hung JY, Gao W, Hung JC (1993) Variable structure control: a survey. IEEE Trans Industr Elect 40:2–22
8. Young KD, Utkin VI, Özgüner Ü (1999) A control engineer's guide to sliding mode control. IEEE Trans Contr Sys Technol 7:328–342
9. Sanner RM, Slotine J-JE (1992) Gaussian networks for direct adaptive control. IEEE Trans Neur Net 3:837–863
10. Hsu L, Real JA (1997) Dual mode adaptive control using Gaussian neural networks. In: Proceedings of the 36th Conference on Decision and Control, (CDC), New Orleans, LA, December 1997
11. Ertugrul M, Kaynak O (2000) Neuro sliding mode control of robotic manipulators. Mechatronics 10:239–263
12. Gao W, Wang Y, Homaifa A (1995) Discrete-time variable structure control systems. IEEE Trans Indust Electr 42:117–122
13. Sarpturk SZ, Istefanopulos Y, Kaynak O (1987) On the stability of discrete-time sliding mode control systems. IEEE Trans Auto Cont 32:930–932
14. Pieper JK, Surgenor BW (1992) Optimal discrete sliding mode control with application. In: Proceedings of the First IEEE Conference on Control Applications, September 1992
15. Sira-Ramirez H (1991) Non-linear discrete variable structure systems in quasi-sliding mode. Int J Cont 54:1171–1187
16. Chen X, Fukuda T (1997) Computer-controlled continuous-time variable structure systems with sliding modes. Int J Cont 67:619–639
17. Misawa EA (1997) Discrete-time sliding mode control for nonlinear systems with unmatched uncertainties and uncertain control vector. Trans ASME J Dyn Sys Meas Cont 119:503–512

18. Park KB (2000) Discrete-time sliding mode controller for linear time-varying systems with uncertainty. Elect Lett 36:2111–2112
19. Chen X, Fukuda T, Young KD (2001) Adaptive quasi-sliding-mode tracking control for discrete uncertain input-output systems. IEEE Trans Indust Elect 48:216–224
20. Tesfaye A, Tomizuka M (1993) Sliding control of discretized continuous systems via the Euler operator. In: Proceedings of the 32nd Conference on Decision and Control, San Antonio, TX, December 1993
21. Xu H, Sun F, Sun Z (1996) The adaptive sliding mode control based on a fuzzy neural network for manipulators. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 3:1942–1946
22. Fang Y, Chow TWS, Li XD (1999) Use of a recurrent neural network in discrete sliding-mode control. IEEE Proc Cont Theor App 146:84–90
23. Muñoz D, Sbarbaro D (2000) An adaptive sliding-mode controller for discrete nonlinear systems. IEEE Trans Indust Elect 47:574–581
24. Haykin S (1994) Neural networks. Prentice-Hall, Englewood Cliffs, NJ
25. Slotine J-JE, Li W (1991) Applied nonlinear control. Prentice-Hall, Englewood Cliffs, NJ
26. Roy RG, Olgaç N (1997) Robust nonlinear control via moving sliding surfaces—n-th order case. In: Proceedings of the 36th Conference on Decision and Control, San Diego, CA, December 1997
27. Yılmaz C, Hürmüzlü Y (2000) Eliminating the reaching phase from variable structure control. Trans ASME J Dyn Sys Meas Cont 122:753–757