Refining the Intelligence with a Task Specific Error Measure

Mehmet Önder Efe[1], Okyay Kaynak[2] and Bogdan M. Wilamowski[3]

[1]Carnegie Mellon University, Electrical and Computer Engineering Department, Pittsburgh, PA15213-3890, U.S.A., efemond@andrew.cmu.edu

[2]Bogazici University, Electrical and Electronics Engineering Department, Bebek, 80815, Istanbul, Turkey, kaynak@boun.edu.tr

[3]University of Idaho, Graduate Center at Boise, 800 Park Blvd., Boise, U.S.A., wilam@ieee.org

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 error backpropagation algorithm. It is observed that the prescribed behavior can be achieved with a simple network configuration.

1. Introduction

Neural Networks (NN) 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 NN in such a wide spectrum of applications has mainly been the ability to represent complex nonlinear mappings, learning and generalization of data together with powerful training strategies and anticipatory behavior. Furthermore, architectural diversity of NN 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 NN.

Among many strategies existing for parameter tuning, Error Backpropagation (EBP) method has been a standard approach in most applications [1]. A common problem in the control applications of NN is the unavailability of the error on the applied control signal [2]. An existing approach to obtain the error at the output of the controller is to identify the plant and to propagate the output error back through the identifier until the controller outputs are reached [3]. When the output error is obtained, the controller parameters can be tuned by EBP technique so that a specified task is fulfilled. A practical drawback is the increase in the computational burden due to the identification process. In this paper, we derive the error measure for Discrete Time Sliding Mode Control (DTSMC) task for a class of uncertain nonlinear systems.

The design of sliding mode controllers is a well-developed framework especially for systems represented 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 [4]. In the literature, various techniques towards the integration of sliding mode control with NN have been presented. The design issues for conventional Discrete Time Sliding Mode Control (DTSMC) have later been addressed in [5], which scrutinize the design of DTSMC with particular emphasis on reaching 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 Discrete Time Sliding Mode Control (DTSMC) is presented in [6], in which the sufficient conditions for convergence are discussed. Pieper et al [7] 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 [8] 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 [9]. In [10], Misawa analyzes the construction of DTSMC under the presence of unmatched uncertainties. Aside from the approaches analyzing the state space representations, a number of studies have demonstrated that NN can successfully be used in DTSMC systems [11-13].

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 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 *k* 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.

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

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 $\underline{g}(k)$ is a vector function of time and the sign of it is known. The system above can compactly be written as $\underline{x}_{k+1} = \underline{f}_k + \underline{g}_k u_k$. According to Figure 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

$$s_k = \underline{\alpha}^T \underline{e}_k \tag{2}$$

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

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

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

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

If the values of the vector functions $\underline{f}_k$ and $\underline{g}_k$ were known explicitly, the application of this sequence to the system of (1) would result in $s_{k+1} = Q(s_k)$, where Q must satisfy the condition below to ensure reaching [5-6,8-9].

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

If the condition above 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 $|s_k| < \varepsilon$, where ε is some positive number. In the literature, this phenomenon is called quasi-sliding mode, or equivalently pseudo-sliding mode [5,9,13]. 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 $|s_k| < \varepsilon$.

2.1. Calculation of the Task Specific Controller Output Error

Define the task as the DTSMC of a plant of the form given in (1), whose ultimate behavior is to be enforced towards what is prescribed by $s_{k+1} = Q(s_k)$. Consider Figure 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 of it, the designer is enforced 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 $\underline{f}_k$ and $\underline{g}_k$ of (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 desired DTSMC can be formulated if the dynamics of the system described by (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 [4]. When the control sequence in (4) is applied to the system of (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 $\underline{f}_k$ and $\underline{g}_k$ 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 (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, $\underline{r}_k = [r_{1k} \quad r_{2k} \quad \cdots \quad r_{nk}]^T$ 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 (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 $\underline{e}_k \equiv \underline{0}$ 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 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 (6).

Fact 2.4: If the target control sequence formulated in (4) were applied to the system of (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 is again the unavailability of the functional forms of $\underline{f}_k$ and $\underline{g}_k$. 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 (4) is rewritten by using (6), one gets

$$u_k = -\left(\underline{\alpha}^T \underline{g}_k\right)^{-1}\left(\underline{\alpha}^T \underline{\Delta f}_k - Q(s_k)\right) + u_{dk} \qquad (7)$$

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 $\underline{f}_k$ 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.

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

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

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

The interpretation of the above 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 (9) to approach zero, this enforces $\underline{\alpha}^T \underline{\Delta f}_k - Q(s_k) = 0$ to approach zero, i.e. $s_{k+1} \to Q(s_k)$, consequently $u_k \to u_{dk}$ as k increases.

Remark 2.5: Notice that the application of u_{dk} for $\forall k \geq 0$ to the system of (1) with zero initial errors will lead to $\underline{e}_k \equiv \underline{0}$ for $\forall k \geq 0$. On the other hand, the application of u_k for $\forall k \geq 0$ to the system of (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 (9) as the equivalent measure of the control error loosens $\underline{e}_k \equiv \underline{0}$ for $\forall k \geq 0$ requirement and enforces $s_{k+1} \to Q(s_k)$. Consequently, the tendency of the control scheme will be to generate the target DTSMC sequence u_k of (4).

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

$$J_{Ck} = \frac{1}{2} s_{Ck}^2 \qquad (10)$$

which instantly qualifies the similarity between u_k and u_{dk}. If the parameters of the neurocontroller are tuned such that the cost in (10) is enforced toward zero, the task implied by $s_{Ck} = 0$ is achieved. More explicitly, a system of structure (1) 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 (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. 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 of the neural network model, in this study, we consider the feedforward NN structure because of its widespread use.

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

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

In above, 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 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. Levenberg-Marquardt optimization method, Gauss-Newton algorithm or conjugate gradients [13]. In order to demonstrate the viability of the extracted error measure, we use EBP technique for parameter adjustment. According to the EBP based tuning strategy, in order to minimize the cost of (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

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

where, η is the learning rate chosen from the interval (0,1) and, J_{Ck} and s_{Ck} are defined as in (10) and (9) respectively. It is apparent that $J_{Ck} \to 0$ means $s_{Ck} \to 0$, hence $u_k \to u_{dk}$. The update rule of (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 (9), the rule in (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 $\underline{g}_k$ is unknown.

2.3. Practical Issues

The analysis presented so far has concentrated on the class of systems having the structure described by (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 system above can be viewed as the plant block in Figure 1. Consequently, the system of (1) will correspond to the sampled system inside the dashed rectangle of the figure.

i. Sampling Time: Since the design presented is based on the discrete time representation of a continuous time system, the selection of 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, discretized form of the system belongs to the class described in (1).

ii. Causality: In (9), we have postulated the error on the applied control at time *k*. However, the right hand side of (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 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.

iii. 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 $s_{k+1} = (1 - \lambda_1 T_s) s_k - \lambda_2 T_s \operatorname{sgn}(s_k)$, where $\lambda_1 > 0$, $\lambda_2 > 0$ and $1 - \lambda_1 T_s > 0$ with T_s being the sampling period [5].

iv. Enhancement of the Behavior in the Quasi-Sliding Mode: It is a well known fact that the use of 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 [14]. 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 [5]. 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 magnitude of the zigzagging during the sliding mode. For this purpose, we adopt the following approximation for the sgn(.) function.

$$\operatorname{sgn}(s_k) \cong \frac{s_k}{|s_k| + \delta} \tag{14}$$

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

3. 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 [16-17]. The continuous time dynamic equation describing the system is given in (15)-(17). Clearly, the system will be of structure (1) when discretized by using 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 $d_4(t)$=0.2sin(4πt) is the disturbance used in [16-17], 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, $|d_i(t)| \le 0.007$ with probability very close to unity. The work presented in [16] assumes that the nominal system dynamics is known and the uncertain part is comprised of what we give as the last three terms in (17). The primary difference between what we discuss and what is assumed in [16] 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=2.5msec, the switching function parameters have been selected as $\underline{\alpha} = [1 \quad 2 \quad 1]^T$ and the parameters of the reaching law are chosen as λ_1=380 and λ_2=1. The neurocontroller possesses three inputs, 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 EBP based parameter adjustment strategy has been chosen as ζ=0.01, and we set δ=0.05 for 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, (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 [16], the reference state trajectory described in (18) is 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 Figure 3, the reference state trajectories and the response of the system are illustrated together. Although the tracking performance is clear from Figure 3, the tracking errors are depicted in Figure 4, which apparently justifies the truth of 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 Figure 5. The error vector hits to the hyperplane several times and moves towards the origin along with the hyperplane. In Figure 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 Figure 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. Apparently the displacements given to the neurocontroller parameters should be convergent to maintain the stability and safety during the training. Assuming that the hidden layer contains H neurons, and defining $\Omega_H = [1 \quad 1 \quad \cdots \quad 1]^T$, which is of $H\times1$ 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 Figure 7, the quantity P is illustrated, whereas the bottom row depicts the same quantity on 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 Figure 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 (18). The results 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 neural network has been used as the controller and the parameters of it have been adjusted by utilizing 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 proved 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.

5. Acknowledgments

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

6. References

[1] Rumelhart DE, Hinton GE, and 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, M.A., 1986, pp. 318-362.

[2] Miller WT, Sutton RS and Werbos PJ. Neural networks for control, MIT Press, 1991.

[3] Narendra KS and Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Trans. on Neural Networks, 1990; 1:4-27.

[4] Hung JY, Gao W and Hung JC. Variable structure control: a survey. IEEE Trans. on Industrial Electronics, 1993; 40:2-22.

[5] Gao W, Wang Y and Homaifa A. Discrete-time variable structure control systems. IEEE Trans. on Industrial Electronics, 1995; 42:117-122.

[6] Sarpturk SZ, Istefanopulos Y and Kaynak O. On the stability of discrete-time sliding mode control systems. IEEE Trans. on Automatic Control, 1987; 32:930-932.

[7] Pieper JK and Surgenor BW. Optimal discrete sliding mode control with application. Proc. of the First IEEE Conf. on Control Applications, 1992; 2:916-921.

[8] Sira-Ramirez H. Non-linear discrete variable structure systems in quasi-sliding mode. Int. Journal of Control, 1991; 54:1171-1187.

[9] Chen X and Fukuda T. Computer-controlled continuous-time variable structure systems with sliding modes. Int. Journal of Control, 1997; 67:619-639.

[10] Misawa EA. Discrete-time sliding mode control for nonlinear systems with unmatched uncertainties and uncertain control vector. Trans. of the ASME, Journal of Dynamic Systems, Measurement, and Control, 1997; 119:503-512.

[11] Xu H, Sun F and Sun Z. The adaptive sliding mode control based on a fuzzy neural network for manipulators. IEEE Int. Conf. on Systems, Man and Cybernetics, 1996; 3:1942-1946.

[12] Fang Y, Chow TWS and Li XD. Use of a recurrent neural network in discrete sliding-mode control. IEE Proc. Control Theory Appl., 1999; 146:84-90.

[13] Muñoz D and Sbarbaro D. An adaptive sliding-mode controller for discrete nonlinear systems. IEEE Trans. on Industrial Electronics, 2000; 47:574-581.

[14] Haykin S. Neural networks, Macmillan College Printing Company, New Jersey, 1994.

[15] Slotine J-JE and Li W. Applied nonlinear control, Prentice-Hall, New Jersey, 1991.

[16] Roy RG and Olgac N. Robust nonlinear control via moving sliding surfaces – n-th order case. Proc. of the 36th Conference on Decision and Control, 1997, San Diego, California, U.S.A., pp. 943-948.

[17] Yilmaz C and Hürmüzlü Y. Eliminating the reaching phase from variable structure control. Trans. of the ASME, Journal of Dynamic Systems, Measurement and Control, 2000; 122:753-757.

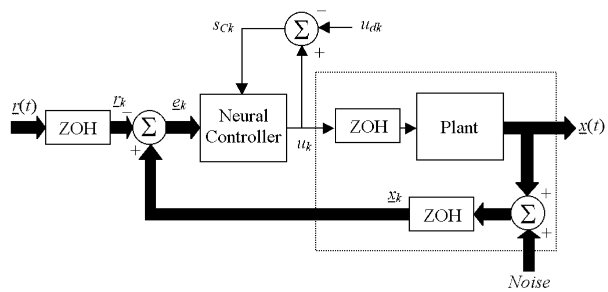

Figure 1. Structure of the Feedback Control System

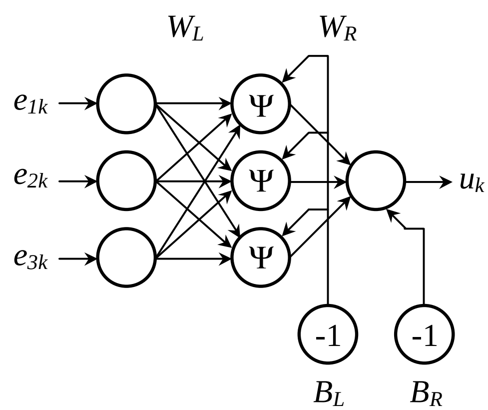

Figure 2. Structure of a Feedforward Neural Network

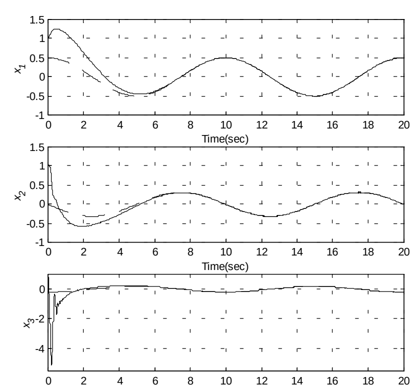

Figure 3. Reference Trajectories (Dashed) and System Response (Solid)

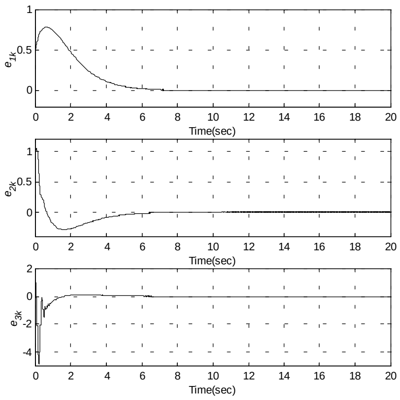

Figure 4. State Tracking Errors

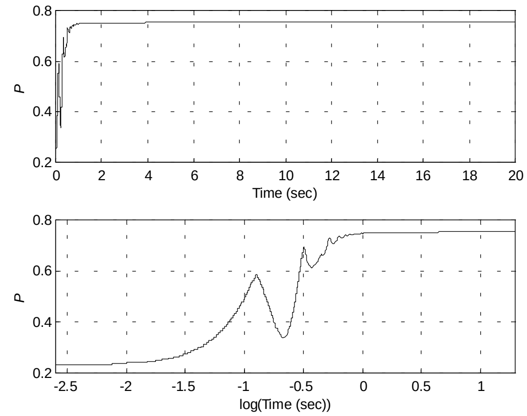

Figure 7. Behavior of the *P* Measure in Linear and Logarithmic Time Axes

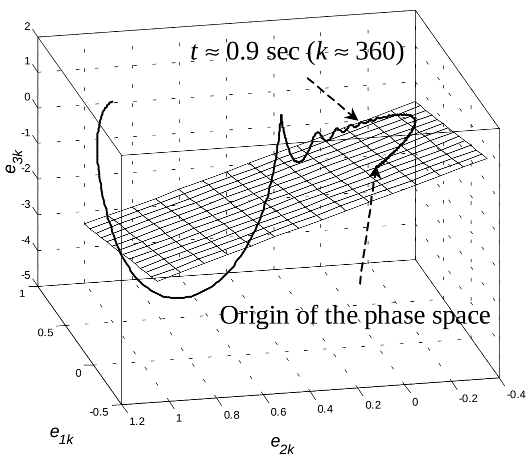

Figure 5. Behavior in the Phase Space

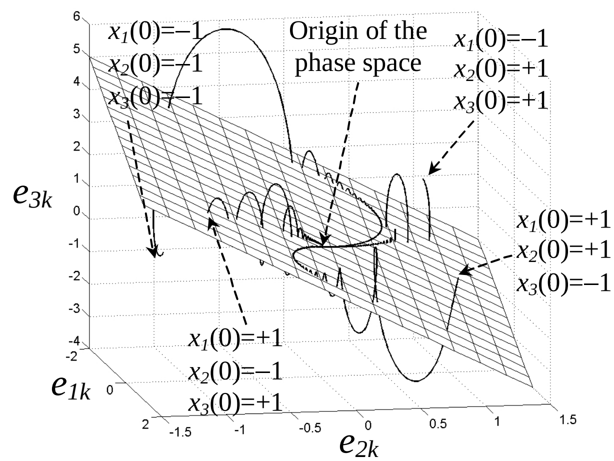

Figure 8. The Behavior in the Phase Space for Different Initial Conditions

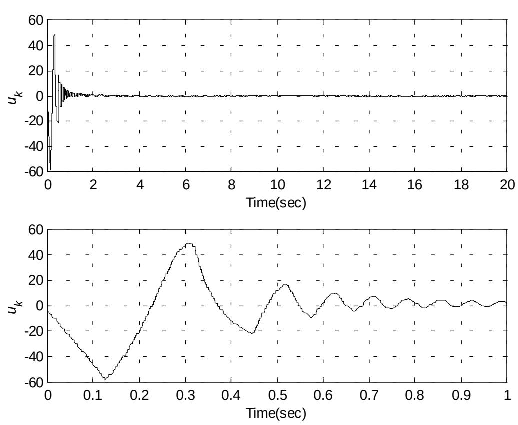

Figure 6. Applied Control Signal and its Transient Behavior