Sliding Mode Control of Nonlinear Systems Using Gaussian Radial Basis Function Neural Networks

M. Önder Efe
Carnegie Mellon University
Electrical and Computer Engineering Department
Pittsburgh, PA 15213-3890, U.S.A.
efemond@andrew.cmu.edu

Okyay Kaynak
Bogazici University
Electrical and Electronic Engineering Department
Bebek, 80815, Istanbul, Turkey,
kaynak@boun.edu.tr

Xinghuo Yu
Central Queensland University
Faculty of Informatics and Communication
Rockhampton QLD 4702, Australia, X.Yu@cqu.edu.au

Bogdan M. Wilamowski
University of Idaho, College of Engineering
Boise, U.S.A.
wilam@ieee.org

Abstract

In this paper, a novel method for driving the dynamics of a nonlinear system to a sliding mode is discussed. The approach is based on a sliding mode control methodology, i.e., the system under control is driven towards a sliding mode by tuning the parameters of the controller. In this loop, the parameters of the controller are adjusted such that a zero learning error level is reached in one dimensional phase space defined on the output of the controller. A Gaussian radial basis function neural network is used as the controller.

1. Introduction

Earliest notion of Sliding Mode Control (SMC) strategy was constructed on a second order system in the late 1960s by Emelyanov [1]. The work stipulated that a special line could be defined on the phase plane, such that any initial state vector can be driven towards the plane and then be maintained on it, resulting in the error dynamics being forced towards the origin. Since then, the theory has greatly been improved and the sliding line has taken the form of a multidimensional surface, called the *sliding surface* around which a switching control action takes place.

In Variable Structure Control, the existence of observation noise constitutes a prime difficulty. This is due to the fact that the ideal sliding control requires very fast switching on the input, which cannot be provided by real actuators, and the input depends on the sign of a measured variable, which is very close to zero. This makes the control signal extremely vulnerable to measurement noise and may lead to unnecessarily large control signals. To alleviate these difficulties, several modifications to the original sliding mode control law have been proposed in the literature, some recent ones of which are based on the use of fuzzy logic [2-3] and artificial neural networks [4-5]. These methodologies provide an extensive freedom for control engineers to exploit their understanding of the problem, to deal with problems of uncertainty and imprecision.

During the last two decades, numerous contributions to VSS theory have been made. Some of them are as follows. Hung *et al* [6] has reviewed the control strategy for linear and nonlinear systems. In that study, the switching schemes, putting the differential equations into canonical forms and generating simple SMC strategies are considered in detail. In [7] and [8], the applications of SMC scheme to robotic manipulators are studied and the quality of the scheme is discussed from the point of robustness. The performance of SMC scheme is proven to be satisfactory in the face of external disturbances and uncertainties in the system model representation. Young *et al* [9] presents another systematic examination of SMC approach. In this reference, the practical aspects of SMC design are assessed for both continuous time and discrete time cases and a special consideration is given to the finite switching frequency, limited bandwidth actuators and parasitic dynamics, all leading to what is known as *chattering*. In [10], the design of discrete time SMC with particular emphasis on the unmatched uncertainties is elaborated.

Some studies on the use of SMC strategy are devoted to the dynamic adaptation of the parameters of a flexible model such that the error on the output of the model tends to zero in finite time [11-12]. The first results obtained by Sira-Ramirez *et al* have concentrated on the inverse dynamics identification of a Kapitsa pendulum by assuming constant bounds for uncertainties. Yu *et al* [13] extend the results of [12] by introducing adaptive uncertainty bound dynamics and focus on the same example as the application. The major drawback in both of the approaches is the fact that the dynamic adaptation mechanism needs the error on the output of the model. If the model is to be used as a controller, this fact constitutes a difficulty because the use of the approaches proposed in [12] and [13] for control applications requires the error on the control signal to be

0-7803-7044-9/01/$10.00 ©2001 IEEE

applied, which is unavailable. The second drawback of the dynamic uncertainty bound adaptation strategy in [13] is the existence of noise on the measured variables. The approach developed presented in this reference requires the integration of the absolute value of the error signal observed on the outputs. When the error signal is close to zero, it clearly leads to the integration of the absolute value of the noise signal, which causes a regular increment on the bound value and leads to instability in the long run.

This paper is organized as follows: The second section presents analytic representation of the GRBFNN structure. The third section gives the definitions and the formulation of the problem. The following section introduces the equivalency constraints on the sliding control performance for the plant and sliding mode learning performance for the controller. The fifth section describes the dynamic model of the plant and the results of the simulations. Conclusions constitute the last part of the paper.

2. Gaussian Radial Basis Function Neural Networks

The fundamental operation in most of the neural network models existing in the literature is the evaluation of a dot product of an input vector and a parameter vector, and to pass the evaluated quantity through a nonlinear activation function. The yield of the described process is the output of the neuron. However, another class of neural networks dwells on the evaluation of the neuron output by combining the values of some appropriately defined basis functions. The networks using basis functions constitute several number of hidden neurons, the activation level of which depend on the distance between the input vector and a prototype vector [14-16].

GRBFNN constitute a special class of these structures. A hidden neuron in a GRBFNN structure uses a Gaussian nonlinearity as the activation function described in (1). In this definition, i indexes the neuron order in the hidden layer while j is for ordering the input vector entries, which runs up to m. The prototype vector is comprised of the c_{ij} variables, which characterize the centers of the Gaussian functions. The variable σ_{ij} determines how the function (μ_{ij}) spreads over the domain of its input space (u_j). The output of i^{th} neuron in the structure is evaluated through the use of (2) and is denoted by w^i. The overall output of the structure depicted is evaluated by a weighted sum of the responses of the neurons contained in the hidden layer and is described by (3).

$$\mu_{ij}\left(u_j\right) = \exp\left\{-\left(\frac{u_j - c_{ij}}{\sigma_{ij}}\right)^2\right\} \qquad (1)$$

$$w^i = \prod_{j=1}^{m} \mu_{ij}\left(u_j\right) \qquad (2)$$

$$\tau = \sum_{i=1}^{h} y^i w^i = \underline{y}^T \underline{w} \qquad (3)$$

In above, y^i denotes the weight determining the effect of i^{th} hidden neuron output on the overall network response τ.

What makes the use of GRBFNN is attractive for control engineering applications is that the hidden neurons provide a degree of similarity between the prototype vectors and the input vector. Therefore, the designer can relatively easily envisage the necessary action for local regions of the input space.

The applications of GRBFNN for the identification and control purposes are discussed in [17-19] and those considering the image/pattern recognition are presented in [15].

3. Definitions and the Formulation of the Problem

Consider the described GRBFNN structure, which is to be used as the controller. The adjustable parameter vector and the vector exciting the adjustable parameters are described in (4) and (5) respectively. The input output relation of the GRBFNN controller is as described in (3).

$$\underline{y} = \left[y^1 \quad y^2 \quad \cdots \quad y^h\right]^T \qquad (4)$$

$$\underline{w} = \left[w^1 \quad w^2 \quad \cdots \quad w^h\right]^T \qquad (5)$$

where, h is the number of hidden neurons contained in the hidden layer. The structure is assumed to operate in an ordinary feedback loop as illustrated in Fig. 1. The definitions of the sliding surface $s_p\left(e,\dot{e}\right)$ and that of zero learning error level $s_c\left(\tau,\tau_d\right)$, are given in (6) and (7) respectively.

$$s_p\left(e,\dot{e}\right) = \dot{e} + \lambda e \qquad (6)$$

where, λ is the parameter determining the slope of the sliding surface.

$$s_c\left(\tau,\tau_d\right) = \tau - \tau_d \qquad (7)$$

where τ_d is the desired output of the controller and is unknown.

In order not to be in conflict with the physical reality, the designer must impose the following inequalities, the truth of which state that the parameters of the controller, the time derivative of the input signal and the time derivative of the desired output of the controller remain bounded.

$$\left\| \underline{y} \right\| = \sqrt{\sum_{i=1}^{h} \left(y^{i}\right)^{2}} \le B_{y} \tag{8}$$

$$\left\| \underline{\dot{w}} \right\| \le B_{\dot{w}} \tag{9}$$

$$\left\| \dot{t}_{d} \right\| \le B_{\dot{t}_{d}} \tag{10}$$

Theorem 1. For a multi input single output flexible structure, whose output is a linear function of the adjustable parameters, the adaptation mechanism as described in (11) enforces the parameters to values resulting in zero learning error level in one dimensional phase space, whose argument is defined by (7).

$$\underline{\dot{y}} = -\frac{\underline{w}}{\underline{w}^{T}\,\underline{w}}\, k \operatorname{sgn}\left(s_{c}\right) \tag{11}$$

where, *k* is a sufficiently large constant satisfying (12).

$$k > B_{y} B_{\dot{w}} + B_{\dot{t}_{d}} \tag{12}$$

The adaptation mechanism in (11) drives an arbitrary initial value of s_c to zero in finite time denoted by t_h satisfying the inequality in (13).

$$t_{h} \le \frac{\left| s_{c}(0) \right|}{k - B_{y} B_{\dot{w}} - B_{\dot{t}_{d}}} \tag{13}$$

Proof 1: See Sira-Ramirez et al [12]

The main problem in applying the design presented is the unavailability of the desired value of the control signal (t_d). If this quantity is not available, one cannot construct s_c and the approach cannot be used for control purposes. In the next section, the relation between the s_p of (6) and s_c of (7) is analyzed.

4. SMC for Learning

Consider the sliding line s_p and the zero learning error level s_c described by (6) and (7) respectively. The relation between these two quantities is assumed to be as given in (14).

$$s_{c} = \Psi\left(s_{p}\right) \tag{14}$$

Qualitatively, if the value of s_p tends to zero, this means that s_c goes to zero. Physically, the system achieves a perfect tracking because the controller produces the desired control inputs or vice versa. Conversely, as the value of s_p increases in magnitude, which means that the error vector is getting away from the origin, the same sort of a divergent behavior in s_c is observed or vice versa. In this section, three conditions that Ψ must satisfy are discussed.

4.1. Region Condition

It should be clear that as the control input approaches the desired value for the current conditions, the state tracking error vector of the plant is driven towards the sliding manifold. In other words, the desired control signal drives the state tracking error to the sliding manifold. In (15), these two statements are clarified.

$$\lim_{t \to t_{d}} s_{p} = 0 \Leftrightarrow \lim_{s_{p} \to 0} t = t_{d} \tag{15}$$

The two equivalent limits and their consequences can be rewritten as given in (16) and (17) by utilizing s_p and s_c.

$$\lim_{s_{c} \to 0} s_{p} = 0 \Rightarrow \left\{ \dot{e} \to -l\, e \Rightarrow \begin{cases} e \to 0 \\ \dot{e} \to 0 \end{cases} \right. \tag{16}$$

$$\lim_{s_{p} \to 0} s_{c} = 0 \Rightarrow \left\{ t \to t_{d} \right. \tag{17}$$

The statements above require the following condition on Ψ.

$$\Psi(0) = 0 \tag{18}$$

Furthermore, the relation Ψ must use the first and the third quadrants of the s_p-$\Psi(s_p)$ coordinate system.

$$\Psi\left(s_{p}\right) = \begin{cases} positive & s_{p} > 0 \\ zero & s_{p} = 0 \\ negative & s_{p} < 0 \end{cases} \tag{19}$$

4.2. Compatibility Condition

In order to measure the tracking performance of the control system; define the Lyapunov function in (20). The realization performance of the controller is defined as $V_c=0.5s_c^2$. If one selects a Ψ relation such that a simultaneous minimization is achieved, then this selection can be considered as a suitable candidate.

$$V_{p} = \frac{1}{2} s_{p}^{2} \tag{20}$$

4.3. Invertibility Condition

If the family of lines described by $s_p=h$ ($h>0$) are drawn for varying values of h, the tracking error vector will fall into one of these subsets of the phase space at each instant of time. However, each one of the members of this family corresponds to a different situation entailing different s_c values. Therefore the relation Ψ must be invertible. In other words, $\exists\, s_p \in \Re$ for $\forall\, s_c \in \Re$.

These three conditions clearly stipulate that the Ψ relation must be such that the horizontal axes of s_c vs. V_c and s_p vs. V_p must be mapped onto each other for simultaneous minimization of the described quadratic functions.

Theorem 2. All monotonically increasing continuous functions can serve as the Ψ relation, which satisfy the three conditions discussed in Sec. 4.1-Sec.4.3, for the establishment of an equivalency between the sliding mode control of the plant and the sliding mode learning inside the controller.

Proof 2: Stability in the Lyapunov sense requires the negative definiteness of the time derivative of the Lyapunov function in (20). Utilizing (21) leads to the time derivative in (22).

$$s_p = \Psi^{-1}(s_c) \tag{21}$$

$$\begin{aligned}
\dot{V}_p &= \dot{s}_p s_p \\
&= \left(\overset{\bullet}{\Psi^{-1}}(s_c)\right)\Psi^{-1}(s_c) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\underline{\dot{y}}^T \underline{w} + \underline{y}^T \underline{\dot{w}} - \dot{t}_d\right)\Psi^{-1}(s_c) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(-k\mathrm{sgn}(s_c) + \underline{y}^T \underline{\dot{w}} - \dot{t}_d\right)\Psi^{-1}(s_c) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\Psi^{-1}(s_c)\left(\underline{y}^T \underline{\dot{w}} - \dot{t}_d\right) - k\left(\Psi^{-1}(s_c)\mathrm{sgn}(s_c)\right)\right) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\left|\Psi^{-1}(s_c)\right|\left(B_G B_{\ddot{u}} + B_{\dot{t}_d}\right) - k\left|\Psi^{-1}(s_c)\right|\right) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left|\Psi^{-1}(s_c)\right|\left(-k + B_G B_{\ddot{u}} + B_{\dot{t}_d}\right)
\end{aligned} \tag{22}$$

Since the partial derivative $\partial\Psi^{-1}(s_c)/\partial s_c$ is positive due to the monotonically increasing behavior of Ψ, the bound parameter given in (12) enforces value of s_c to zero level, or equivalently, s_p to zero. It is straightforward to prove that a hitting occurs in finite time (See Proof 1). ÿ

5. Simulation Studies

In this study, a coupled double pendulum system, which is illustrated in Fig. 2, is used to elaborate the performance of the method discussed. The differential equations characterizing the behavior of the system are given in (23)-(26), in which the angular positions and the angular velocities define the state vector. The control inputs, which are denoted by t$_1$ and t$_2$, are provided to the relevant pendulum by the base servomotors. The model introduced in this section has been studied by Spooner and Passino [20], who discuss the decentralized adaptive control using radial basis neural networks. The parameters of the plant are given in [20], which states that as $b<l$, the two pendulums repel each other in the upright position.

$$\dot{x}_1 = x_3 \tag{23}$$

$$\dot{x}_2 = x_4 \tag{24}$$

$$\begin{aligned}
\dot{x}_3 = &\left(\frac{M_1 gr}{J_1} - \frac{k_s r^2}{4J_1}\right)\sin(x_1) + \frac{k_s r}{2J_1}(l-b) + \\
&\frac{\mathrm{t}_1}{J_1} + \frac{k_s r^2}{4J_1}\sin(x_4)
\end{aligned} \tag{25}$$

$$\begin{aligned}
\dot{x}_4 = &\left(\frac{M_2 gr}{J_2} - \frac{k_s r^2}{4J_2}\right)\sin(x_2) - \frac{k_s r}{2J_2}(l-b) + \\
&\frac{\mathrm{t}_2}{J_2} + \frac{k_s r^2}{4J_2}\sin(x_3)
\end{aligned} \tag{26}$$

where, g=9.81 m/s^2 is the gravitational acceleration constant. In the simulation studies presented, the architecture discussed in the second section is adopted with 9 hidden neurons, i.e. h=9. Based on the tracking error vector, first the value of $s_p(e,\dot{e})$ is evaluated and this quantity is passed through the Ψ function to get the value of s_c, which is used in the dynamic adjustment mechanism. In evaluating the value of the quantity s_p, the slope parameter of the switching line (l) has been set to unity for both controllers.

To study the effects of observation noise, which is very likely to be encountered in practice, the information used by the controller is corrupted by a Gaussian distributed random noise having zero mean and variance equal to 0.33e-6. The peak magnitude of the noise signal is within ±1e-3 with probability very close to unity. The second difficulty is the nonzero positional initial conditions. In order to demonstrate the reaching mode performance of the algorithm, the first pendulum is moved to π/6 radians and the second one is moved to –π/4 radians initially. The reference trajectory used in the simulations is depicted in Fig. 3.

It should be pointed out that once the error or the rate of error comes very close to zero, the adjustment mechanism is driven solely by the outputs of the hidden neurons whose prototype vectors are close to the vector $\underline{0}_{2x1}$. The change in the outputs of these neurons is due to the noise sequence. Since the bound of perturbing signal is known, the dynamic equation of the parameter y^i can be modified so that a reduction on the unnecessary adjustment activity is obtained and the convergent behavior of the parameters can still be achieved by utilizing a sufficiently hard threshold function given by (27). The value of threshold is denoted by

n_b and has been set to 2e-3 in the simulations. The modified form of the update equation is given in (28).

$$T\left(s_p\right) = \left(1 + \exp\left(-10^5\left(\left|s_p\right| - n_b\right)\right)\right)^{-1} \tag{27}$$

$$\dot{y}^i = -\frac{w^i}{\sum_{j=1}^{h}\left(w^j\right)^2} k \operatorname{sgn}\left(\Psi\left(s_p\right)\right) T\left(s_p\right) \tag{28}$$

As the Ψ relation, the following selection is made parallel to the conditions discussed in the fourth section.

$$\Psi\left(s_p\right) = s_p \tag{29}$$

Furthermore, in order to reduce the chattering effect in the sliding mode, the function in (30) has been used instead of the sgn function in the dynamic strategy described in (28), and initially, the adjustable parameters are all set to zero.

$$\operatorname{sgn}\left(\Psi\left(s_p\right)\right) \approx \frac{\Psi\left(s_p\right)}{\left|\Psi\left(s_p\right)\right| + 0.05} \tag{30}$$

Under these conditions, phase plane motions depicted in Fig. 4 are obtained. The trend in position and velocity errors clearly stipulate that the algorithm is able to achieve precise tracking objective with a response characterized by l . Due to the space limit, the behavior of the torque signals and the evolution in the parameter space is omitted.

During the simulations, the bounds for the uncertainties denoted by *k* for both pendulums has been set to 1000. The simulation stepsize has been selected as 2.5 msec and the time required to perform the simulation has been measured as 12.32 seconds on a Pentium III-600 PC running Matlab 5.2 software, indicating that the applicability of the algorithm in real time.

6. Conclusions

In this paper, a novel method for establishing a sliding motion in the dynamics of a nonlinear plant is discussed. The method is based on the adoption of a nonlinear dynamic adjustment strategy in a GRBFNN based controller. The task is to drive the tracking error vector to the sliding manifold and keep it on the manifold forever. What makes the proposed algorithm so attractive in this sense is the fact that the sliding mode control of the plant is achieved while an equivalent regime is imposed on the controller parameters. Contrary to what is known in the field of variable structure controller design, the governing equations of the plant under control are assumed to be unknown and the lack of this knowledge is left as a difficulty alleviated by a learning controller.

As discussed throughout the paper, the problems that arise due to the uncertainties are alleviated by incorporating the robustness provided by the VSS technique into the proposed approach. A further attractiveness of the algorithm is the fact that the controller for each pendulum possesses only nine adjustable parameters. The computational requirement is not therefore excessive.

Finally, the simulation results presented demonstrate that the algorithm discussed is able to compensate deficiencies caused by the imperfect observations of the state variables, large initial errors and complex plant dynamics. From these points of view, the method proposed is highly promising in control engineering practice.

7. References

[1] Emelyanov, S. V. *Variable structure control systems*, Moscow, Nauka, 1967.

[2] Q. P. Ha, “Robust Sliding Mode Controller with Fuzzy Tuning,” *Electronics Letters*, Vol. 32, No. 17, pp. 1626-1628, 1996.

[3] Y. Byungkook and W. Ham, “Adaptive Fuzzy Sliding Mode Control of Nonlinear Systems”, *IEEE Trans. on Fuzzy Systems*, Vol. 6, No: 2, pp. 315-321, 1998.

[4] M. Ertugrul and O. Kaynak, “Neuro Sliding Mode Control of Robotic Manipulators,” *Mechatronics*, Vol. 10, No: 1, pp. 243-267, 2000.

[5] K. Jezernik, M. Rodic, R. Safaric and B. Curk, “Neural Network Sliding Mode Control,” *Robotica*, Vol. 15, pp. 23-30, 1997.

[6] J. Y. Hung, W. Gao and J. C. Hung, “Variable Structure Control: A Survey, *IEEE Trans. on Industrial Electronics*,” Vol. 40, No: 1, pp. 2-22, 1993.

[7] W. Gao and J. C. Hung, “Variable Structure Control of Nonlinear Systems: A New Approach,” *IEEE Trans. on Industrial Electronics*, Vol. 40, No: 1, pp. 45-55, 1993.

[8] O. Kaynak, F. Harashima and H. Hashimoto, “Variable Structure Systems Theory, as Applied to Sub-time Optimal Position Control with an Invariant Trajectory”, *Trans. IEE of Japan*, Sec. E, 104 Vol. 3/4, pp. 47-52, 1984.

[9] K. D. Young, V. I. Utkin and U. Ozguner, “A Control Engineer’s Guide to Sliding Mode Control,” *IEEE Trans. on Control Systems Technology*, Vol. 7, No: 3, pp. 328-342, 1999.

[10] E. A. Misawa, “Discrete-Time Sliding Mode Control for Nonlinear Systems With Unmatched Uncertainties and Uncertain Control Vector,” *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, Vol. 119, pp. 503-512, September 1997.

[11] G. G. Parma, B. R. Menezes, and A. P. Braga, “Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks,” *Electronics Letters*, Vol. 34, No. 1, pp. 97-98, January 1998.

[12] H. Sira-Ramirez, and E. Colina-Morles, “A Sliding Mode Strategy for Adaptive Learning in Adalines,” *IEEE Trans. on Circuits and Systems – I: Fundamental Theory and Applications*, Vol. 42, No: 12, pp. 1001-1012, 1995.

[13] X. Yu, M. Zhihong and S. M. M. Rahman, “Adaptive Sliding Mode Approach for Learning in a Feedforward Neural Network,” *Neural Computing and Applications*, No: 7, pp. 289-294, 1998.

[14] E. Hartman, J. D. Keeler and J. M. Kowalski, “Layered Neural Networks with Gaussian Hidden Units as Universal Approximations,” *Neural Computation*, Vol. 2, No: 2, pp. 210-215, 1990.

[15] Girosi F., Poggio, T. and Caprile, B., “Extension of a Theory of Networks for Approximation and Learning: Outliners and Negative Examples,” *Advances in Neural Information Processing Systems*, (Eds.) R. P Lippmann, J. E. Moody, and D. S. Touretzky, pp. 750-756, San Mateo, 1991.

[16] J. Moody and C. J. Darken, “Fast Learning Networks of Locally-Tuned Processing Units,” *Neural Computation*, Vol. 1, No. 2, pp. 281-294, 1989.

[17] M. O. Efe and O. Kaynak, “A Comparative Study of Soft Computing Methodologies in Identification of Robotic Manipulators,” *Robotics and Autonomous Systems*, Vol. 30, No: 3, pp. 221-230, 2000.

[18] S. Seshagiri and H. K. Khalil, “Output Feedback Control of Nonlinear Systems Using RBF Neural Networks,” *IEEE Transactions on Neural Networks*,” Vol. 11, No: 1, pp. 69-79, 2000.

[19] R. M. Sanner and J.-J. E. Slotine, “Gaussian Networks for Direct Adaptive Control, *IEEE Transactions on Neural Networks*,” Vol. 3, No: 6, pp. 837-863, 1992.

[20] J. T. Spooner and K. M. Passino, “Decentralized Adadptive Control of Nonlinear Systems Using Radial Basis Neural Networks,” *IEEE Trans. on Automatic Control*, Vol. 44, No: 11, pp. 2050-2057, 1999.

Acknowledgments

This work is supported by Bogazici University Research Fund (Project no: 00A203D) and by NSF (Grant No: 9906233).

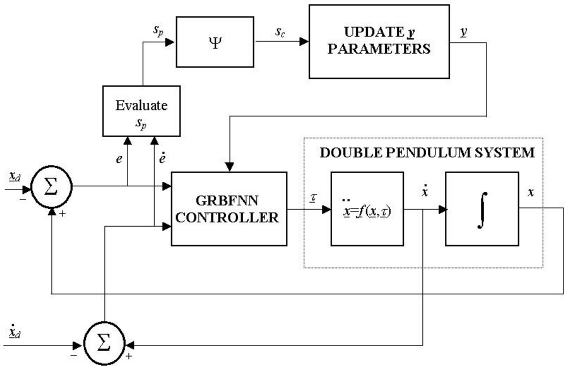

Figure 1: Structure of the control system

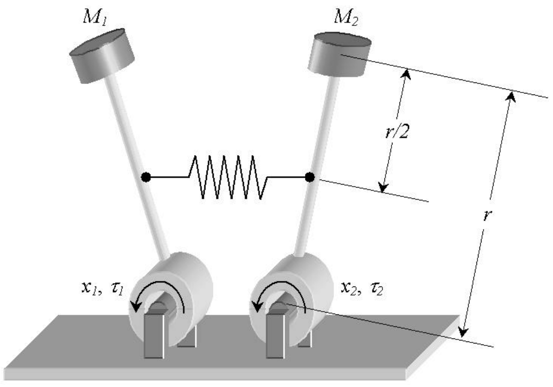

Figure2: Physical structure of the double pendulum system

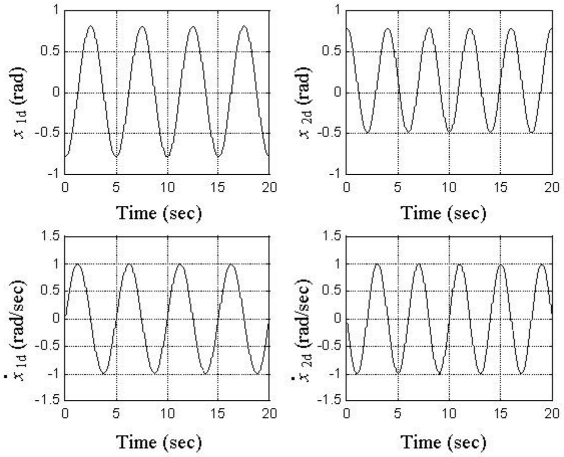

Figure 3: Reference state trajectories

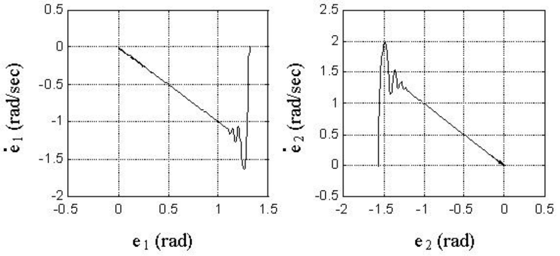

Figure 4: Behavior in the phase space