

Radial Basis Function Neural Networks in Variable Structure Control of a Class of Biochemical Processes

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

Bogdan M. Wilamowski
University of Idaho
Graduate Center at Boise
800 Park Blvd., Boise, ID 83712
U.S.A.
wilam@ieee.org

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

Abstract – Biochemical processes often display a complicated dynamic behavior, the detailed understanding of which frequently constitutes a barrier between the theoretical foundations and practical implementations. One way of handling the complexity is to use intelligent approaches in the design of controllers. This paper presents an analytic approach to design controllers based on Radial Basis Function Neural Networks (RBFNN) with particular emphasis on the extraction of the error measure to be used in parameter tuning. The simulation studies stipulate that the control system exhibits a highly robust behavior against disturbances and sharp changes in the command signal. The most important contribution of the paper is that the method presented does not require the analytical details describing the plant dynamics available.

I. INTRODUCTION

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 [1-3]. The overall structure is called RBFNN, which possesses the distributed information processing capability.

The problem of parameter tuning in RBFNN has extensively been studied in the literature. Some important ones of which are the gradient descent technique (Error Backpropagation) [4], Levenberg-Marquardt algorithm [5], and hybrid methods such as Variable Structure Systems (VSS) theory based learning strategies [6-8]. At a first glance, what a reader notices is the fact that the application of above mentioned approaches for tuning the parameters of a

controller require the target value of the control signal, which is unavailable by the nature of the problem. Therefore, the extraction of the error on the control signal can be done either by identifying the plant under control or by assuming the plant under control is known and the Jacobian can be evaluated. Clearly the former increases the computational cost and the latter restricts the domain of applicability. A qualitative analysis of extracting the equivalent control error is discussed in [9], which considers the control of a 3-dof anthropoid robot with large initial conditions and unknown plant dynamics.

In this paper, an analytic approach towards the calculation of the error at the output of the neurocontroller is presented for a class of biochemical systems. In the next section, RBFNN are introduced, the third section presents a brief overview of variable structure control. The fourth section is devoted to the computation of the error at the output of the controller and in the fifth section the parameter tuning law is given. The sixth section presents the results obtained with the proposed technique. The concluding remarks are presented at the end of the paper.

II. RADIAL BASIS FUNCTION NEURAL NETWORKS

In the literature, RBFNN are generally considered as a smooth transition between Fuzzy Inference Systems (FIS) and Neural Networks (NN). Structurally, a RBFNN is composed of receptive units (neurons) which act as the operators providing the information about the class to which the input signal belongs. If the aggregation method, number of receptive units in the hidden layer and the constant terms are equal to those of a FIS, then there exists a functional equivalence between RBFNN and FIS [10]. Although the architectural view of a RBFNN is very similar to that of an ordinary feedforward neural network illustrated in Fig. 1, the hidden neurons of a RBFNN possess basis functions to characterize the partitions of the input space. Each neuron in the hidden layer provides a degree of membership value for the input pattern with respect to the basis vector of the receptive unit itself. The output layer is comprised of linear

neurons. NN interpretation makes RBFNN useful in incorporating the mathematical tractability, especially in the sense of propagating the error back through the network, while the FIS interpretation enables the incorporation of the expert knowledge into the training procedure. The latter is of particular importance in assigning the initial value of the network's adjustable parameter vector to a vector that is to be sought iteratively. Expectedly, this results in faster convergence in parameter space.

Referring to Fig. 1, mathematically, $o_i = \prod_{j=1}^m \Psi_{ij}(u_j)$ and a common choice for the hidden layer activation function is the Gaussian curve described as $\Psi_{ij}(u) = \exp\{-(u_j - c_{ij})^2 / \sigma_{ij}^2\}$, where c_{ij} and σ_{ij} stand for the center and the variance of the i^{th} neuron's activation function qualifying the j^{th} input variable. The output of the network is evaluated through the inner product of the adjustable weight vector denoted by $\underline{\phi}$ and the vector of hidden layer outputs, i.e. $\tau = \underline{\phi}^T \underline{o}$. Clearly the adjustable parameter set of the structure is composed of $\{\underline{c}, \underline{\sigma}, \underline{\phi}\}$ triplet. In this paper, only $\underline{\phi}$ parameters are tuned.

III. A BRIEF OVERVIEW OF VARIABLE STRUCTURE CONTROL

Consider the feedback loop illustrated in Fig. 2, in which a subscript d denotes the desired value of the relevant quantity. Furthermore, it is shown in the figure that if a supervisor provides the desired controller outputs, one might evaluate the error on the control signal denoted by \underline{s}_c . The plant shown in the figure is assumed to have the structure described in (1), in which $\underline{\theta}$ and $\underline{\tau}$ are the $(r_1 + r_2 + \dots + r_n) \times 1$ -dimensional state vector and $n \times 1$ -dimensional input vector.

$$\theta_i^{(r_i)} = f_{p_i}(\underline{\theta}) + \sum_{j=1}^n d_{ij} \tau_j \quad i = 1, 2, \dots, n \quad (1)$$

The system of (1) with these vectors can be restated as $\dot{\underline{\theta}} = \underline{f}_{-p}(\underline{\theta}) + D \underline{\tau}$. The design problem is to enforce the behavior of the system towards the desired response, which is known but the control signal ($\underline{\tau}_d$) resulting in which is unavailable. Therefore, the solution to this problem is a search towards the synthesis of such a signal iteratively by a RBFNN controller. Assuming that the controller in Fig. 2 is composed of n individual RBFNN controllers, the i^{th} one of which is to construct the i^{th} component of input vector $\underline{\tau}$, the j^{th} entry of the error vector driving this sub-controller can be given as $e_i^{(j)} = \theta_i^{(j)} - \theta_{di}^{(j)}$. Apparently, this component is the j^{th} derivative of the relevant state component. Consider the vector of sliding surfaces for the system in (1): $\underline{s}_p(\underline{e}) = G \underline{e} = G(\underline{\theta} - \underline{\theta}_d)$. The widespread selection of the matrix G is such that the i^{th} sliding surface function has the form

$$s_{p_i}(e_i) = \left(\frac{d}{dt} + \lambda_i \right)^{r_i - 1} e_i \quad (2)$$

in which, λ_i is a strictly positive constant. Let V_p be a candidate Lyapunov function given as

$$V_p(\underline{s}_p) = \frac{1}{2} \underline{s}_p^T \underline{s}_p \quad (3)$$

If the prescribed control signal satisfies $\dot{V}_p(\underline{s}_p) = -\underline{s}_p^T \xi \text{sgn}(\underline{s}_p)$, the negative definiteness of the time derivative of the Lyapunov function in (3) is ensured. In above, ξ is a positive definite diagonal matrix of dimension $n \times n$. More explicitly, $\underline{s}_p^T \dot{\underline{s}}_p = -\underline{s}_p^T \xi \text{sgn}(\underline{s}_p)$ must hold true to drive the error vector towards the sliding hypersurface. On the other hand, the use of $\dot{\underline{s}}_p = -G \dot{\underline{\theta}}_d + G \left(\underline{f}_{-p}(\underline{\theta}) + D \underline{\tau} \right)$ leads to the following control signal:

$$\underline{\tau} = -(GD)^{-1} \left(G \underline{f}_{-p}(\underline{\theta}) - G \dot{\underline{\theta}}_d \right) - (GD)^{-1} \xi \text{sgn}(\underline{s}_p) \quad (4)$$

in which the first term is the equivalent control term and the second term is the corrective control term. For the existence of the mentioned components, the matrix GD must not be rank deficient. In the literature, equivalent control is considered as the low frequency (average) component of the control signal. Because of the discontinuity on the sliding surface, the corrective term brings a high rate component [11-12]. If $\underline{e}(0) = \underline{0}$, the tracking problem can be considered as keeping \underline{e} on the sliding surface, however, for nonzero initial conditions, the strategy must enforce the state trajectories towards the sliding surface, which is ensured by the negative definiteness of the time derivative of the Lyapunov function as in (3). For the case of nonzero initial conditions, the phase until the error vector hits the sliding surface is called the reaching mode, the dynamic characteristics of the system during which is determined by the control strategy adopted. Application of the control input formulated in (5) imposes the dynamics described as $\dot{\underline{s}}_p = -\xi \text{sgn}(\underline{s}_p)$, which clearly enforces the error vector towards the sliding surface. Once the sliding surface is reached, the value of (2) becomes zero; and this enforces the error vector to move towards the origin.

Aside from the practical difficulties of conventional Sliding Mode Control (SMC) schemes, the control signal in (4) is applicable if a nominal representation of the system under control is available. In the next subsection, a method for obtaining the error on the control signal is presented for unknown systems of structure (1).

IV. CONTROL ERROR COMPUTATION

Remark 4.1: The SMC task is achievable if the dynamics of the system in (1) is totally known or if the nominal system is known with the bounds of the uncertainties. It must be noted that to satisfy the matching conditions, the disturbances and uncertainties are always assumed to enter the system through the control channels [13]. When the conventional SMC strategy is applied to the system of (1), we call the resulting behavior as the *target SMC* and the input vector leading to it as the *target control sequence* ($\underline{\tau}$), which is described in (4). If the functional form of the vector function

f_p is not known, it should be obvious that the target control sequence cannot be constructed by following the traditional SMC design approaches.

Definition 4.2: Given an uncertain plant, which has the structure described as in (1), and a command trajectory vector $\underline{\theta}_d(t)$ for $t \geq 0$, the input sequence satisfying the following vector differential equation is defined to be the *idealized control sequence* denoted by $\underline{\tau}_d$, and the vector differential equation itself is defined to be the *reference SMC model*.

$$\dot{\underline{\theta}}_d = \underline{f}_{-p}(\underline{\theta}_d) + D\underline{\tau}_d \quad (5)$$

Mathematically, the existence of such a model and the sequence means that the system of (1) perfectly follows the command trajectory vector if both the idealized control sequence is known and the initial conditions are set as $\underline{\theta}(t=0) = \underline{\theta}_d(t=0)$, more explicitly $\underline{e}(t) \equiv \underline{0}$ for $\forall t \geq 0$. Undoubtedly, such an idealized control sequence will not be a norm-bounded signal when there are step-like changes in the vector of command trajectories or when the initial errors are nonzero. It is therefore that the reference SMC model is an abstraction due to the limitations of the physical reality, but the concept of idealized control sequence should be viewed as the synthesis of the command signal $\underline{\theta}_d$ from the time solution of the differential equation set in (5).

Fact 4.3: Based on the Lyapunov stability results of the previous subsection, 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. $\lim_{t \rightarrow \infty} \underline{\tau} = \underline{\tau}_d$. However, under the assumption of the achievability of the SMC task, the difficulty here is again the unavailability of the functional form of the vector function f_p . 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 $\underline{s}_c = \underline{\tau} - \underline{\tau}_d$ and is of $n \times 1$ dimensional. If the target control sequence of (4) is rewritten by using (5), one gets

$$\begin{aligned} \underline{\tau} &= -(GD)^{-1} \left(G\underline{f}_{-p}(\underline{\theta}) - G \left(\underline{f}_{-p}(\underline{\theta}_d) + D\underline{\tau}_d \right) + \xi \operatorname{sgn}(\underline{s}_p) \right) \\ &= -(GD)^{-1} \left(G\underline{\Delta f}_{-p}(\underline{\theta}) + \xi \operatorname{sgn}(\underline{s}_p) \right) + \underline{\tau}_d \end{aligned} \quad (6)$$

The target control sequence becomes identical to the idealized control sequence, i.e. $\underline{\tau} \equiv \underline{\tau}_d$, as long as $G\underline{\Delta f}_{-p}(\underline{\theta}) + \xi \operatorname{sgn}(\underline{s}_p) = \underline{0}$ holds true. However, this condition is of no practical importance as we do not have the analytic form of the vector function f_p . 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 $\underline{s}_c = \underline{0}$ and is the aim of the design. It is obvious that to enforce this to hold true will let us synthesize the target control sequence, which will ultimately converge to the idealized control sequence by the adaptation algorithm yet to be

discussed. Consider the time derivative of the vector of sliding surfaces

$$\begin{aligned} \dot{\underline{s}}_p(\underline{e}) &= G\dot{\underline{e}} = G(\dot{\underline{\theta}} - \dot{\underline{\theta}}_d) \\ &= G \left(\underline{f}_{-p}(\underline{\theta}) + D\underline{\tau} - \underline{f}_{-p}(\underline{\theta}_d) - D\underline{\tau}_d \right) \\ &= G \left(\underline{\Delta f}_{-p} + D(\underline{\tau} - \underline{\tau}_d) \right) = G \left(\underline{\Delta f}_{-p} + D\underline{s}_c \right) \end{aligned} \quad (7)$$

Utilizing $G\underline{\Delta f}_{-p}(\underline{\theta}) + \xi \operatorname{sgn}(\underline{s}_p) = \underline{0}$ in (7) and solving for \underline{s}_c yields the following relation:

$$\underline{s}_c = (GD)^{-1} \left(\dot{\underline{s}}_p + \xi \operatorname{sgn}(\underline{s}_p) \right) = \underline{\tau} - \underline{\tau}_d \quad (8)$$

Remark 4.4: The reader must here notice that the application of $\underline{\tau}_d$ to the system of (1) with zero initial errors will lead to $\underline{e}(t) \equiv \underline{0}$ for $\forall t \geq 0$, on the other hand, the application of $\underline{\tau}$ to the system of (1) will lead to $\underline{s}_p = \underline{0}$ for $\forall t \geq t_h$, where t_h is the hitting time, and the origin will be reached according to the dynamics of the sliding surface. Therefore, the adoption of (8) as the equivalent measure of the control error loosens $\underline{e}(t) \equiv \underline{0}$ for $\forall t \geq 0$ requirement and introduces all trajectories in the error space to tend to the sliding hypersurface, i.e. $G\underline{\Delta f}_{-p}(\underline{\theta}) + \xi \operatorname{sgn}(\underline{s}_p) = \underline{0}$ is enforced.

Consequently, the tendency of the control scheme will be to generate the target SMC sequence of (4) without requiring the analytical details of the plant.

Now consider the ordinary feedback control loop illustrated in Fig. 2, and define the following Lyapunov function, which is a measure of how well the controller performs:

$$V_c(\underline{s}_c) = \frac{1}{2} \underline{s}_c^T \underline{s}_c \quad (9)$$

Remark 4.5: An adaptation algorithm ensuring $\dot{V}_c(\underline{s}_c) < 0$ when $\underline{s}_c \neq \underline{0}$ enforces $G\underline{\Delta f}_{-p}(\underline{\theta}) + \xi \operatorname{sgn}(\underline{s}_p) = \underline{0}$ and creates the predefined sliding regime after a reaching mode lasting until the hitting time denoted by t_h , beyond which $\underline{s}_c = \underline{0}$ as the system is in the sliding regime. If $\dot{V}_c(\underline{s}_c) < 0$ when $\underline{s}_c \neq \underline{0}$, then $\lim_{t \rightarrow t_h} V_c = 0 \Leftrightarrow \lim_{t \rightarrow t_h} \|\underline{s}_c\| = 0 \Leftrightarrow \lim_{t \rightarrow t_h} \left\| \dot{\underline{s}}_p + \xi \operatorname{sgn}(\underline{s}_p) \right\| = 0$. Note that the meaning of $\underline{s}_c = \underline{0}$ is now equivalent to $\underline{s}_p = \underline{0}$ by Remark 4.4, therefore the limits above are evaluated as $t \rightarrow t_h$.

V. PARAMETER TUNING STRATEGY

If the architecture introduced in the second section is utilized for the purpose of control, without loss of generality, the output of the i^{th} controller can be restated as $\tau_i = \underline{\phi}_i^T \underline{\Omega}_i$, where $\underline{\Omega}_i$ is the vector of signals exciting the adjustable parameters denoted by $\underline{\phi}_i$ and the Lyapunov function in (9) constitutes the basis of the design.

In order not to be in conflict with the physical reality, the designer must impose $\|\underline{\phi}_i\| \leq B_{\phi_i}$, $\|\underline{\dot{\Omega}}_i\| \leq B_{\dot{\Omega}_i}$, $\|\underline{\dot{\tau}}_{id}\| \leq B_{\dot{\tau}_{id}}$, the truth of which state that the adjustable parameters of the controller, the time derivative of the signal exciting the adjustable parameter set and the time derivative of the idealized output of the controller remain bounded. Note that in Definition 4.2, we stated that there may not be a finite $B_{\dot{\tau}_{id}} \in \mathfrak{R}$ even in some realistic situations like nonzero initial errors, however, the practical meaning of imposing $\|\underline{\dot{\tau}}_{id}\| \leq B_{\dot{\tau}_{id}}$ will lead us to the construction of an approximation of the idealized control sequence and the requirement of $\underline{e}(t) \equiv 0$ for $\forall t \geq 0$ must therefore be loosened.

Theorem 5.1: For the i^{th} subsystem of the system described in (1), adopting the controller of structure $\tau_i = \underline{\phi}_i^T \underline{\Omega}_i$, the adaptation of the controller parameters as described in (10) enforces the value of the i^{th} component of control discrepancy vector (s_{ci}) to zero.

$$\dot{\underline{\phi}}_i = -\frac{\underline{\Omega}_i}{\underline{\Omega}_i^T \underline{\Omega}_i} k_i \text{sgn}(s_{ci}) \quad (10)$$

where, k_i is a sufficiently large positive constant satisfying $k_i > B_{\phi_i} B_{\dot{\Omega}_i} + B_{\dot{\tau}_{id}}$. The adaptation mechanism in (10) drives an arbitrary initial value of s_{ci} to zero in finite time denoted by t_{hi} satisfying the inequality in (11).

$$t_{hi} \leq \frac{|s_{ci}(0)|}{k_i - (B_{\phi_i} B_{\dot{\Omega}_i} + B_{\dot{\tau}_{id}})} \quad (11)$$

Proof: See Sira-Ramirez et al [6] and Efe et al [9].

An important feature of this approach is the fact that the controller parameters evolve bounded as assumed initially. The details of the bounded parametric evolution analysis can be found in [9,14].

VI. SIMULATION RESULTS

In the simulations, we consider the dynamic model of a Continuously Stirred Tank Reactor (CSTR) discussed in [15], which illustrate the SMC task with Gaussian networks and wavelet networks. The governing equations of the process dynamics are as described below, and the parameters are defined in Table 1.

$$\dot{\theta}_1 = -\theta_1 + Da(1 - \theta_1) \exp\left(\frac{\theta_2}{1 + \frac{\theta_1}{\gamma}}\right) \quad (12)$$

$$\dot{\theta}_2 = -\theta_2 + T_R Da(1 - \theta_1) \exp\left(\frac{\theta_2}{1 + \frac{\theta_1}{\gamma}}\right) - \beta(\theta_2 - \theta_c) + \eta \quad (13)$$

The control problem is to enforce the dimensionless concentration (θ_1) to follow a desired trajectory by altering the dimensionless coolant temperature (θ_c). During the control operation, the second state, which is the dimensionless temperature (θ_2), is constrained to evolve bounded in time. In [15], the nominal operating point of the CSTR system is described as $\theta_1=0.4126$, $\theta_2=3.28$ and $\theta_c=3.04$, the state values among which are used as the initial state values in this paper. According to the simulation settings given in Table 2, one should notice from the first row that the controller uses solely the noise corrupted tracking error information in synthesizing the necessary control sequence. During the simulations, we tune solely the weight parameters of the RBFNN controller and we set these parameters initially to zero, i.e. the parametric evolution starts from the origin. The noise sequences are Gaussians having zero mean. In order to compute the error given in (8), we use the following filter structure to evaluate the time derivative of \underline{s}_p . Furthermore, we set $\delta=0.25$ and $\alpha=1$.

$$H(S) = \frac{\alpha S}{Q(S)} = \frac{\alpha S}{S^2 + 2\sqrt{\alpha}|S + \alpha|} \quad (14)$$

where, S is the Laplace variable. In Fig. 3, the desired and the observed states are illustrated. In the bottom right subplot of this figure, the error is seen. At time $t=3200\text{sec}$, a step change occurs in the reference trajectory, which is also studied by Knapp et al [15], and the system successfully follows the imposed trajectory. It must be noted that since the system under control is of first order, the sliding surface of the conventional design becomes a point in the single dimensional error space, and this point corresponds to the origin. Consequently, the problem does not require a λ selection. In the top left subplot of Fig. 4, the applied control signal (θ_c) is illustrated. This subplot reveals that the control signal has a sufficiently smooth characteristic after the transient phase. The remaining three subplots in Fig. 4 depict the time evolution of the adjustable neurocontroller parameters. These subplots confirm the evolution in finite volume claim of [6,9]. The variables seen in Figs. 3 and 4 have been redrawn in Figs. 5 and 6 with the same graphical allocation but around $t=3200\text{sec}$, at which a step change occurs in the command signal. When compared to the results discussed in [15], it can be said that the settling time is not as small as in [15], but the computational simplicity, i.e. the number of neurons, and acceptable tracking accuracy make the approach presented a good candidate for control of chemical processes.

VII. CONCLUSIONS

This paper discusses a method for controlling a class of nonlinear systems by using RBFNN. The well-known problem of obtaining the error measure that is to be used in parameter tuning is analyzed. In the simulations, a CSTR is considered. The results obtained confirm the accurate tracking and bounded parameter evolution claims of the

paper. What makes the discussed technique attractive is its low computational cost. In Fig. 7, a computational complexity chart is depicted. Based on what is seen in the figure, the application presented in this paper requires 143 floating point operations (flops) for performing a forward pass through the neurocontroller for control signal evaluation and a backward pass for parameter adjustment. The filter equation in (14) is solved by 4th order Runge-Kutta approach and the result obtained is found to be quite reasonable even for average speed microcontrollers.

VIII. ACKNOWLEDGMENTS

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

IX. REFERENCES

- [1] 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.
- [2] F. Girosi, T. Poggio and B. Caprile, "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.
- [3] J. Moody and C. J. Darken, "Fast Learning Networks of Locally-Tuned Processing Units," *Neural Computation*, Vol. 1, No. 2, pp. 281-294, 1989.
- [4] M. I. Jordan and D. E. Rumelhart, "Forward Models: Supervised Learning with a Distal Teacher," *Cognitive Science*, No: 16, pp. 307-354, 1992.
- [5] M. T. Hagan and M. B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Trans. on Neural Networks*, Vol. 5, No. 6, pp. 989-993, November 1994.
- [6] 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, December 1995.
- [7] 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.
- [8] M. O. Efe and O. Kaynak, "On Stabilization of Gradient Based Training Strategies for Computationally Intelligent Systems," *IEEE Trans. on Fuzzy Systems*, Vol.8, No.5, pp.564-575, October 2000.
- [9] M. O. Efe, O. Kaynak and X. Yu, "Sliding Mode Control of a Three Degrees of Freedom Anthropoid Robot by Driving the Controller Parameters to an Equivalent Regime," *Trans. of the ASME: Journal of Dynamic Systems, Measurement and Control*, Vol.122, No.4, pp.632-640, December 2000.
- [10] J.-S. R. Jang, C.-T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, PTR Prentice-Hall, 1997.
- [11] V. I. Utkin, *Sliding Modes in Control Optimization*, Springer Verlag, New York, 1992.
- [12] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall 1991.
- [13] J. Y. Hung, W. Gao and J. C. Hung, "Variable Structure Control: A survey," *IEEE Trans. on Industrial Electronics*, Vol. 40, No. 1, pp.1-9, February 1993.
- [14] X. Yu, M. Zhihong and S. M. M. Rahman, "Adaptive Sliding Mode Approach for Learning in a Feedforward Neural Network," *Neural Computing & Applications*, Vol. 7, pp. 289-294, 1998.
- [15] T. D. Knapp, H. M. Budman and G. Broderick, "Adaptive Control of a CSTR with Neural Network Model," *Journal of Process Control*, Vol. 11, pp. 53-68, 2001.

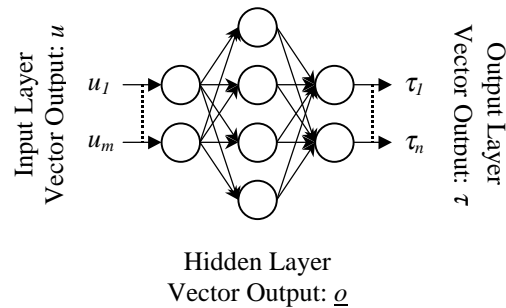


Fig. 1. Structure of a RBFNN

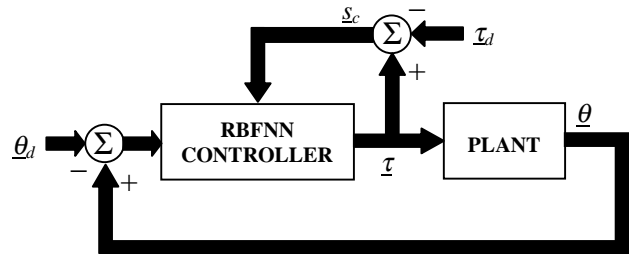


Fig. 2. Block Diagram of the control system

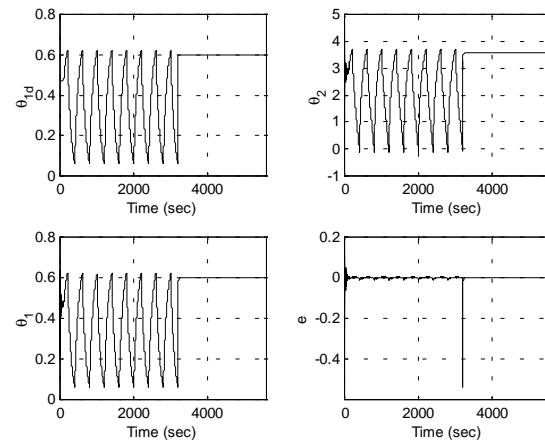


Fig. 3. Desired and observed states with error signal

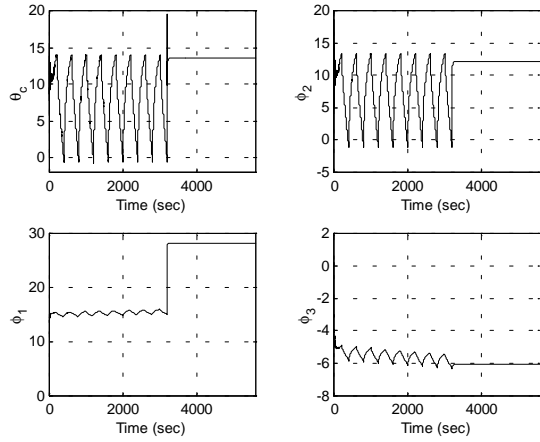


Fig. 4. Control signal and parametric evolution

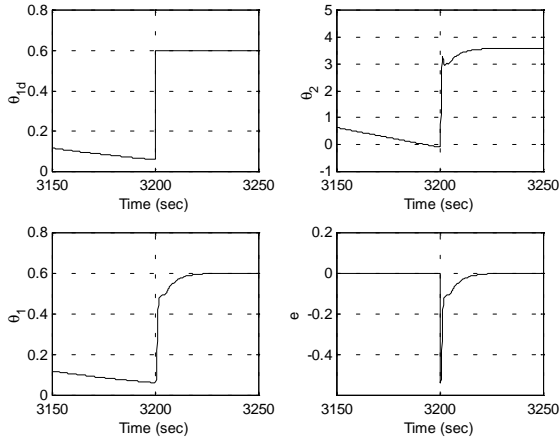


Fig. 5. Zoomed Fig. 3 around $t=3200$ sec.

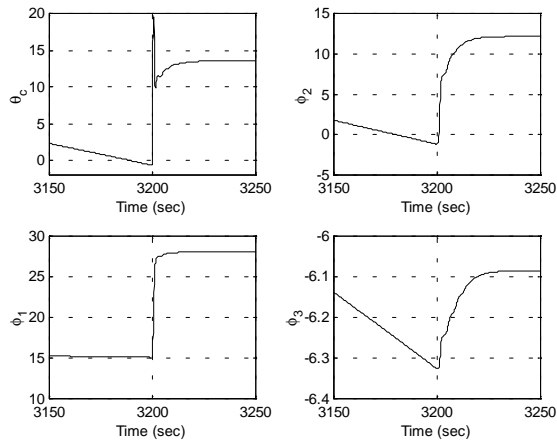


Fig. 6. Zoomed Fig. 4 around $t=3200$ sec.

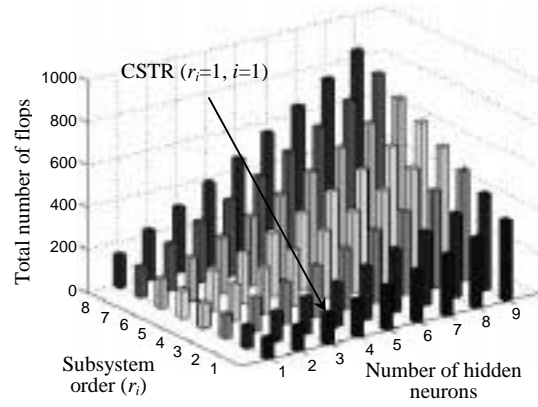


Fig. 7. Computational complexity chart

Table 1. Parameters of the CSTR Dynamics

Dimensionless concentration	θ_1	State variable
Dimensionless temperature	θ_2	State variable
Dimensionless coolant temp.	θ_c	Control input
Damkohler number	Da	0.072
Dimensionless cooling rate	β	0.3
Dimensionless activation energy	γ	20
Dimensionless heat of reaction	T_R	1
Disturbance	η	See Table 2

Table 2. Simulation Data

Controller Input Vector	$\underline{u}_i = e_i, i=1$
# of Hidden Neurons	3
Uncertainty Bounds	$k=20$
Simulation Stepsize	$T_s=0.1$ sec
Initial Errors	$e(0) = -0.0566$
Noise Variance	$7.3543e-8$
Noise Peak Value with probability ≈ 1	$1e-3$
SMC Design Matrix	$\xi = 0.1I_{1 \times 1}$
Initialization of the Basis Functions, which are kept static during the simulations	