Neuro-Fuzzy Approaches for Identification and Control of Nonlinear Systems

M. Onder Efe and Okyay Kaynak
Bogazici University, Electrical and Electronic Engineering Department
Mechatronics Research and Application Center
Bebek 80815, Istanbul, TURKEY

Abstract - **Neural Networks and Fuzzy Inference Systems are becoming well-recognized tools of designing an identifier/controller capable of perceiving the operating environment and imitating a human operator with high performance. The motivation behind the use of neuro-fuzzy approaches is based on the complexity of real life systems, ambiguities on sensory information or time-varying nature of the system under investigation. In this respect, neuro-fuzzy design approaches combine architectural (by neural networks) and philosophical (by fuzzy systems) aspects of an expert resulting in an artificial brain, which can be used as an identifier or a controller. It is known that the fuzzy inference systems and neural networks are universal approximators. An architecture with an appropriate learning strategy can teach any mapping to such a system with a predefined realization error bound. The most questionable quality in the use of neuro-fuzzy architectures is the stable training. This tutorial considers various neuro-fuzzy structures and gradient based training procedures. Consideration is given to stabilization of training dynamics.**

1. INTRODUCTION

Twentieth century has witnessed widespread innovations in both hardware and software design. In the first half of the century, the emphasis was mainly on the development of accurate mechanical component design, whereas in the second half, new technologies emerged together with new needs and new directions in industry. The development of fast microprocessors enabled the design and implementation of *Expert-Machine Interaction* based computation environments. Ever increasing needs brought about by the multidimensionality of the problem space and time-varying behavior of real-life physical systems further required to reduce the role of expert and to increase the role of machine. A natural consequence of this rapid growth is the emergence of the field of intelligent systems, or in other words, the field of neuro-fuzzy systems.

Artificial Neural Networks are well known with their property of representing complex nonlinear mappings. Earlier works on the mapping properties of these architectures have shown that neural networks are universal approximators [1-2]. The mathematical power of intelligence is commonly attributed to the neural systems because of their massively interconnected, fault tolerant architecture. Various architectures of neural systems are studied in the literature. Feedforward and recurrent neural networks, Gaussian radial basis function neural networks and dynamical neural networks [3] constitute typical structurally different approaches. These have successfully been applied to problems extending from control applications to image/pattern recognition problems.

Fuzzy Inference Systems are the most popular constituent of the soft computing area since they are able to represent human expertise in the form of *IF antecedent THEN consequent* statements. In this domain, the system behavior is modeled through the use of linguistic descriptions. Although the earliest work by Prof. Zadeh on fuzzy systems has not been paid as much attention as it deserved in early 1960s, since then the methodology has become a well-developed framework. The typical architectures of fuzzy inference systems are those introduced by Wang [4-5], Takagi and Sugeno [6] and Jang [7]. In [4], a fuzzy system having Gaussian membership functions, product inference rule and weighted average defuzzifier is constructed and has become the standard method in most applications. Takagi and Sugeno change the defuzzification procedure where dynamic systems are introduced as defuzzification subsystems. The potential advantage of the method is that under certain constraints, the stability of the system can be studied. Jang *et al* [6] propose an adaptive neuro fuzzy inference system, in which a polynomial is used as the defuzzifier. This structure is commonly referred to as ANFIS in the related literature. The choice concerning the order of the polynomial and the variables to be used in the defuzzifier are left to the designer.

The second emphasis in this tutorial is on the stabilization of gradient based training dynamics. The concept of stabilization concerns the parametric convergence property during the training phase. If a learning algorithm uses the sensitivity derivatives of the cost defined solely on the realization error, the parametric displacements can assume large values depending on the shape of the surface. Therefore the training algorithm must be incorporated with a guaranteeing term leading to parametric stability. This term can be constructed by the use of Variable Structure Systems (VSS) philosophy. VSS approach has first been proposed by Emelyanov [8] and its early applications have been confined to linear systems. In the case of a second order system, a sliding line is

defined on the phase plane. In the reaching mode, the system is driven towards the plane and is maintained on it during the sliding mode. The approach presented in this tutorial considers the training dynamics as the *system* to which the VSS approach is applied. The approach discussed therefore enforces the parametric displacements to those imposed by VSS based solution.

The organization of this tutorial is as follows. The second section briefly discusses the error backpropagation based training of artificial neural networks. The following section considers the same task for Gaussian radial basis function neural networks. In the third section standard fuzzy inference system model is discussed. The fifth section dwells on the adaptive neuro-fuzzy inference system architecture and its training. In the sixth section, the parameter stabilizing criteria is derived and how a mixture of realization cost minimization and parametric cost minimization is performed is explained. The seventh section is devoted to the plant model used for the simulations. The eighth section demonstrates the architectures adopted for identification and control of systems by neuro-fuzzy models. The simulation results are presented and discussed in the ninth section. Conclusions constitute the last part of the paper.

2. FEEDFORWARD NEURAL NETWORKS (FNN)

In this section, a multilayer perceptron is introduced as the flexible architecture, the parameters of which are to be updated by using the error backpropagation technique [9]. In [10], Narendra and Parthasarathy demonstrate that this structure can effectively be used for identification and control purposes. In the conventional error backpropagation method, propagating the output error, which is described by (1), back through the neural network, which is illustrated in Fig. 1, minimizes the cost function given in (2).

$$e = d - F(\phi) \tag{1}$$

$$J = \frac{1}{2} e^2 \tag{2}$$

where, d, F, and ϕ denote the target output, network response and the generic representation of an adjustable parameter respectively. Gradient rule anticipates the following change on parameter ϕ.

$$\Delta\phi = -\eta \frac{\partial J}{\partial \phi} \tag{3}$$

$$\Delta\phi = \eta e \frac{\partial F(\phi)}{\partial \phi} \tag{4}$$

Based on the derivation presented in detail in [9], the delta values for the output and hidden layer neurons are evaluated as given by (5) and (6) respectively.

$$\delta_j^{k+1,p} = (d_j^p - f_j^{k+1,p})\, \Psi'(S_j^{k+1,p}) \tag{5}$$

$$\delta_j^{k+1,p} = \left(\sum_{h=1}^{\#neurons_{k+2}} \delta_h^{k+2,p}\, w_{jh}^{k+1} \right) \Psi'(S_j^{k+1,p}) \tag{6}$$

where, $S_j^{k+1,p}$ is the net summation for the j^{th} neuron in the $(k+1)^{th}$ layer. Having evaluated the delta values during the backward pass, the weight update rule described by (7) is applied for each training pair with the backpropagated error value defined in (8).

$$\Delta w_{ij\,EBP}^k = \zeta N_{w_{ij}}^k \tag{7}$$

$$N_{w_{ij}}^k = \delta_j^{k+1,p}\, o_i^{k,p} \tag{8}$$

In (7), ζ is the constant learning rate selected from the interval (0,1).

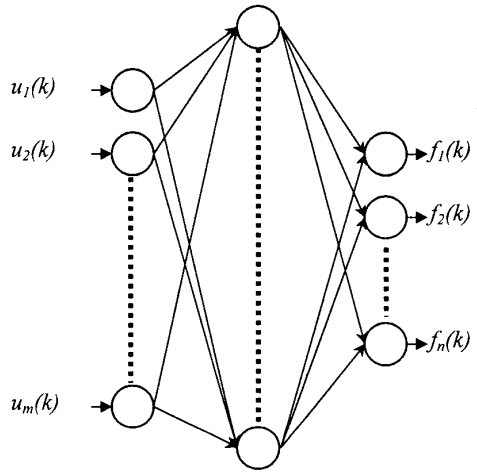

Figure 1. Architecture of a Feedforward Neural Network

In [11], the use of multilayer perceptron is considered for control of a two degrees of freedom SCARA robotic manipulator with the stabilizing learning approach discussed in the sixth section. In [12] and [13], identification of robotic manipulators are comparatively evaluated and the feedforward neural network architecture is considered as one of the identification approaches.

3. GAUSSIAN RADIAL BASIS FUNCTION NEURAL NETWORKS (RBFNN)

In the literature, RBFNN are generally considered as a smooth transition between fuzzy logic and neural networks. Structurally, RBFNN are 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 Fuzzy Inference System (FIS), then there exists a functional equivalence between RBFNN and FIS [7]. In Fig. 2, a RBFNN

structure is illustrated. 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. Neural network interpretation makes RBFNN useful in incorporating the mathematical tractability, especially in the sense of propagating the error back through the network, while the fuzzy system interpretation enables the incorporation of the expert knowledge into the training procedure. The latter may be crucial in assigning the initial value of the network's adjustable parameter vector to a vector that is close to the optimal one. This results in faster convergence in parameter space provided that the system dynamics is known [12-14].

In the RBFNN used in this work, as is given by (9), each neuron output is evaluated from the multiplication of the outputs of individual Gaussians corresponding to each input. The overall response is evaluated through multiplication of hidden layer output vector by a matrix/vector of appropriate dimensions. In fuzzy terms, this is equivalent to say that, product inference rule is used with weighted sum defuzzifier.

If the network output and hidden layer output vectors are denoted by F and $\underline{o}$ respectively, the activation level of i^{th} receptive unit (or firing strength of each rule) and the overall realization performed by the network can be described by (9) and (10) respectively.

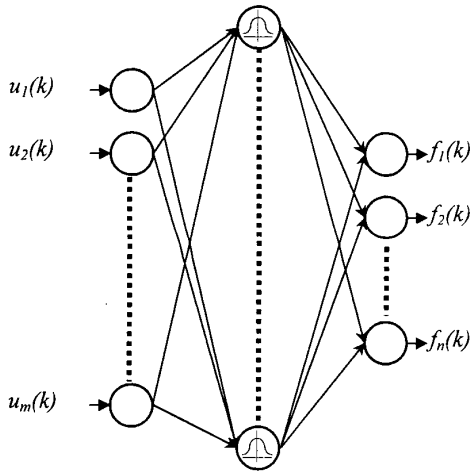

Figure 2. Architecture of a Gaussian Radial Basis Function Neural Network

$$o_i = \prod_{J=1}^{\#inputs} \exp\left\{-\left(\frac{u_j - c_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{9}$$

$$F = W_{\#outputs\,x\,\#hiddemeurons}\underline{o} \tag{10}$$

According to the error backpropagation rule, the parameter update law can be summarized as follows:

$$\underline{\delta}_{output} = \underline{d} - F \tag{11}$$

$$\underline{\delta}_{hidden} = W^T \underline{\delta}_{output} \tag{12}$$

$$\Delta W_{ij\,EBP} = \zeta N_{W_{ij}} \tag{13}$$

$$\Delta c_{ij\,EBP} = \zeta N_{c_{ij}} \tag{14}$$

$$\Delta \sigma_{ij\,EBP} = \zeta N_{\sigma_{ij}} \tag{15}$$

where,

$$N_{W_{ij}} = \delta_{output\ i} o_j \tag{16}$$

$$N_{c_{ij}} = \delta_{hidden i} o_j 2\left(\frac{u_j - c_{ij}}{\sigma_{ij}}\right)^2 \tag{17}$$

$$N_{\sigma_{ij}} = \delta_{hidden\ i} o_j 2\frac{(u_j - c_{ij})^2}{\sigma_{ij}^3} \tag{18}$$

4. STANDARD FUZZY SYSTEMS (SFS)

This section considers the standard fuzzy system approach introduced in [4] as the computationally intelligent architecture. The system that is considered in this study uses Gaussian membership functions as described by (19). The architecture of the system is depicted in Fig. 3.

$$\mu_{ij}(u_j) = \exp\left\{-\left(\frac{u_j - c_{ij,}}{\sigma_{ij}}\right)^2\right\} \tag{19}$$

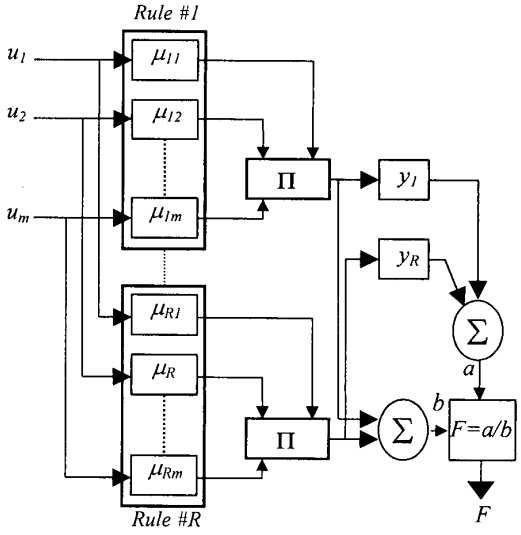

Figure 3. Architecture of Standard Fuzzy System

In (19), c_{ij} defines the center of i^{th} rule's j^{th} membership function, c_{ij} and σ_{ij} characterize the center and width of that function respectively. The structure of fuzzy system is illustrated in Fig. 3, for which the following type of a rule base structure is adopted.

IF u_1 *is* $U_{i,1}$ *AND* u_2 *is* $U_{i,2}$ *AND ... AND* u_m *is* $U_{i,m}$
THEN $F = Y_i$

In this representation, lowercase variables denote the inputs, uppercase variables stand for the fuzzy sets corresponding to the domain of each linguistic label. During the simulations the initial values of the membership functions are selected such that the region of interest is covered appropriately.

The overall realization performed by the system considered is given in (20), where weighted average defuzzifier is used with algebraic product aggregation method.

$$F = \frac{\sum_{i=1}^{\#\,Rules} Y_i \prod_{j=1}^{\#\,Inputs} \mu_{ij}(u_j)}{\sum_{i=1}^{\#\,Rules} \prod_{j=1}^{\#\,Inputs} \mu_{ij}(u_j)} = \sum_{i=1}^{\#\,Rules} Y_i w_{ni} \tag{20}$$

In (20), the vector of firing strengths denoted by w is normalized and the resulting vector is represented by w_n.

$$w_{ni} = \frac{\prod_{j=1}^{\#\,Inputs} \mu_{ij}(u_j)}{\sum_{k=1}^{\#\,Rules} \prod_{j=1}^{\#\,Inputs} \mu_{kj}(u_j)} \tag{21}$$

Based on the forward pass relations given by (20) and (21), and the cost defined in Sec. 2, relevant parameter update formulas are evaluated as follows.

$$\Delta Y_{i\,EBP} = \zeta N_{Y_{ij}} \tag{22}$$

$$\Delta c_{ij\,EBP} = \zeta N_{c_{ij}} \tag{23}$$

$$\Delta \sigma_{ij\,EBP} = \zeta N_{\sigma_{ij}} \tag{24}$$

where,

$$N_{Y_i} = e w_{ni} \tag{25}$$

$$N_{c_{ij}} = e(Y_i - F) w_{ni} 2 \frac{u_j - c_{ij}}{\sigma_{ij}^2} \tag{26}$$

$$N_{\sigma_{ij}} = e(Y_i - F) w_{ni} 2 \frac{(u_j - c_{ij})^2}{\sigma_{ij}^3} \tag{27}$$

The control applications of this architecture are discussed in [11] in detail with the approach leading to parametric stabilization, which is discussed in Sec. 6.

5. ADAPTIVE NEURO FUZZY INFERENCE SYSTEMS (ANFIS)

Adaptive Neuro-Fuzzy Inference Systems are realized by an appropriate combination of neural and fuzzy systems. This hybrid combination enables to utilize both the verbal and the numeric power of intelligent systems. As is known from the theory of fuzzy systems, different fuzzification and defuzzification strategies with different rule base structures can result in various solutions to a given task. This paper considers the ANFIS structure with first order Sugeno model containing nine rules. Gaussian membership functions with product inference rule are used at the fuzzification level. Fuzzifier outputs the firing strengths for each rule. The vector of the firing strengths is normalized and the resulting vector is defuzzified by utilizing the first order Sugeno model. The structure for two inputs and one output is illustrated in Fig. 4 and a sample rule is described below for a m-input one output ANFIS.

IF u_1 *is* $U_{i,1}$ *AND* u_2 *is* $U_{i,2}$ *AND ... AND* u_m *is* $U_{i,m}$
THEN $f_i = q_{i,1}u_1 + \ldots + q_{i,m}u_m + q_{i,m+1}$

In the IF part of this representation, lowercase variables denote the inputs, uppercase variables stand for the fuzzy sets corresponding to the domain of each linguistic label. The ANFIS output is clearly a linear function of the adjustable defuzzifier parameters denoted by $q_{i,j}$. The system that is considered in this study uses Gaussian membership functions as described by (28).

$$\mu_{ij}(u_j) = \exp\left\{-\left(\frac{u_j - c_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{28}$$

In above, c_{ij} and σ_{ij} characterize the center and width of i^{th} rule's j^{th} membership function respectively. The initial values of the membership functions are selected such that the region of interest is covered appropriately. The overall realization performed by the system considered is given in (29), where linear functions of input variables are used as defuzzifier with algebraic product aggregation method.

$$F = \frac{\sum_{i=1}^{\#\,Rules} f_i \prod_{j=1}^{\#\,Inputs} \mu_{ij}(u_j)}{\sum_{i=1}^{\#\,Rules} \prod_{j=1}^{\#\,Inputs} \mu_{ij}(u_j)} = \sum_{i=1}^{\#\,Rules} f_i w_{ni} \tag{29}$$

$$w_{ni} = \frac{\prod_{j=1}^{\#\,Inputs} \mu_{ij}(u_j)}{\sum_{k=1}^{\#\,Rules} \prod_{j=1}^{\#\,Inputs} \mu_{kj}(u_j)} \tag{30}$$

In (29), the vector of firing strengths is normalized and the resulting vector is represented by w_n and given by (30). With the definition given in (30), and the realization described by (29), the adjustable parameter set can be selected as follows.

$$\phi = \left\{ c_{ij}, \sigma_{ij}, q_{i,j} \right\}_{\substack{i=1,\ldots,Rules \\ j=1,\ldots,Inputs}} \tag{31}$$

The relevant backpropagated error values are given in (32) through (34).

$$N_{q_{i,j}} = \begin{cases} ew_{ni} u_j & 1 \le j \le m+1 \\ ew_{ni} & j = m+1 \end{cases} \tag{32}$$

$$N_{c_{ij}} = e(f_i - F) w_{ni} 2 \frac{u_j - c_{ij}}{\sigma_{ij}^2} \tag{33}$$

$$N_{\sigma_{ij}} = e(f_i - F) w_{ni} 2 \frac{(u_j - c_{ij})^2}{\sigma_{ij}^3} \tag{34}$$

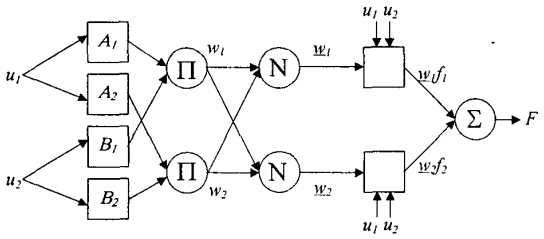

Figure 4. Architecture of ANFIS

The control applications of ANFIS architecture with stabilizing learning criteria is analyzed and discussed in [15] in detail. The examples using ANFIS architecture for identification purposes can be found in [12] and [13].

6. STABILIZING THE TRAINING DYNAMICS BY USING VSS APPROACH

In control engineering practice, stability and robustness are of crucial importance. Because of this, the implementation-oriented control engineering expert is always in pursuit of a design, which provide accuracy as well as insensitivity to environmental disturbances and structural uncertainties. At this point, it must be emphasized that these ambiguities can never be modeled accurately. When the designer tries to minimize the ambiguities by the use of a detailed model, then the design becomes so tedious that its cost increases dramatically. A suitable way of tackling with uncertainties without the use of complicated models is to introduce VSS theory based components into the system structure.

Variable Structure Control (VSC) has successfully been applied to a wide variety of systems having uncertainties in the representative system models. The philosophy of the control strategy is simple, being based on two goals. First, the system is forced towards a desired dynamics, second, the system is maintained on that differential geometry. In the literature, the former dynamics is named the reaching mode, while the latter is called the sliding mode. The control strategy borrows its name from the latter dynamic behavior, and is called *Sliding Mode Control (SMC)*.

Numerous contributions to VSS theory have been made during the last decade, some of them are as follows: Hung *et al* [16] have reviewed the control strategy for linear and nonlinear systems. In [16], the switching schemes, putting the differential equations into canonical forms and generating simple SMC strategies are considered in detail. Gao *et al* [17], apply the SMC scheme to robotic manipulators and discuss the quality of the scheme. One of the crucial points in SMC is the selection of the parameters of the sliding surface. Some studies devoted to the adaptive design of sliding surfaces have shown that the performance of control system can be refined by interfacing it with an adaptation mechanism, which regularly redesigns the sliding surface [18-19]. This eventually results in a robust control system. The performance of SMC scheme is proven to be satisfactory in the face of external disturbances and uncertainties in the system model representation. The latest studies consider this robustness property by equipping the system with computationally intelligent methods. In [20] and [21], fuzzy inference systems are proposed for SMC scheme. A standard fuzzy system is studied and the relevant robustness analyses are carried out. Particularly, the work presented in [20] emphasizes that the robustness and stability properties of soft computing based control strategies can be studied through the use SMC theory. It is shown in the paper in this way that the approach is robust i. e. it can compensate the deficiencies caused by poor modeling of plant dynamics and external disturbances.

The objective of this section is to develop a stabilizing training procedure for neuro-fuzzy architectures, which will force the adjustable parameters to settle down to a steady state solution while minimizing an appropriate cost function defined on the realization performance. This is achieved by performing an appropriate mixture of gradient based training procedure [9] and variable structure systems based stabilizing criteria.

The observation error in (35) is used to minimize the cost function in (36) by utilizing the rule described by (37), which can be restated as given in (38).

$$e = d - F(\phi) \tag{35}$$

$$J = \frac{1}{2} e^2 \tag{36}$$

$$\Delta\phi = -\eta \frac{\partial J}{\partial \phi} \tag{37}$$

$$\Delta\phi = \eta_\phi e \frac{\partial F(\phi)}{\partial \phi} \tag{38}$$

The minimization proceeds iteratively as given in (38), for which the sensitivity derivative with respect to the generic parameter ϕ is needed. It is apparent that the method is applicable to the architectures in which the outputs are differentiable with respect to the subject of optimization.

A continuous-time dynamic model of the parameter update procedure can be constructed as described by (39).

$$\dot{\Delta\phi} = -\frac{1}{T_s}\Delta\phi + \frac{\eta_\phi}{T_s}N_\phi \tag{39}$$

The above model can easily be validated at the integer multiples of the sampling period T_s by replacing the derivative term with Euler's first order approximation. This substitution clarifies the equivalence between (38) and (39).

In (39), the evaluated parameter change, which is by error backpropagation and denoted by N_ϕ, is multiplied by a scaling factor denoted by η_ϕ, for the selection of which, a detailed analysis is presented in the subsequent discussion. In the design of variable structure controllers, one method that can be followed is the reaching law approach [16]. For the use of this theory in the stabilization of the training dynamics, let us define the switching function as in (40) and its dynamics as in (41).

$$s_\phi = \Delta\phi \tag{40}$$

$$\dot{s}_\phi = -\frac{Q_\phi}{T_s}\tanh\left(\frac{s_\phi}{\varepsilon}\right) - \frac{K_\phi}{T_s}s_\phi = \dot{\Delta\phi} \tag{41}$$

In above, Q_ϕ and K_ϕ are the gains and ε is the width of the boundary layer. Equating (39) and (41) and solving for $\Delta\phi$ yields the following:

$$\Delta\phi = \eta_\phi N_\phi + Q_\phi \tanh\left(\frac{s_\phi}{\varepsilon}\right) + K_\phi s_\phi \tag{42}$$

In the derivations presented below, a key point is the fact that the system described by (39) is also driven by η_ϕ, which is known as learning rate in the related literature. Now we demonstrate that some special selection of this quantity leads to the minimization of the magnitude of the parametric displacements. Let us define the following quantity for keeping analytic comprehensibility;

$$A_\phi = Q_\phi \tanh\left(\frac{\Delta\phi}{\varepsilon}\right) + K_\phi\Delta\phi \tag{43}$$

Now we have a model described by (39), and a solution formulated by (42). If one chooses a positive definite Lyapunov function as given by (44), the time derivative of this function must be negative definite for stability of parameter change ($\Delta\phi$) dynamics. Clearly the stability in parameter change space implies the convergence in system parameters.

$$V_\phi = \frac{1}{2}s_\phi^2 = \frac{1}{2}(\Delta\phi)^2 \tag{44}$$

$$\dot{V}_\phi = (\Delta\phi)(\dot{\Delta\phi}) \tag{45}$$

If (39) and (42) are substituted into (45), the constraint stated in (46) is obtained for stability in the Lyapunov sense.

$$\eta_\phi^2 + \frac{1}{N_\phi}(A_\phi - \Delta\phi)\eta_\phi - \frac{1}{N_\phi^2}A_\phi\Delta\phi < 0 \tag{46}$$

Equation (46) can be rewritten in a more tractable form as follows.

$$\left(\eta_\phi + \frac{1}{N_\phi}A_\phi\right)\left(\eta_\phi - \frac{1}{N_\phi}\Delta\phi\right) < 0 \tag{47}$$

Since A_ϕ and $\Delta\phi$ have the same signs, the roots of the expression (47) clearly have opposite signs. The expression on the left-hand side assumes negative values between the roots. Therefore, in order to satisfy the inequality (47), the learning rate must satisfy the constraint given by (48).

$$0 < \eta_\phi < \min\left\{\left|\frac{1}{N_\phi}\Delta\phi\right|, \left|-\frac{1}{N_\phi}A_\phi\right|\right\} \tag{48}$$

In (48), the interval of learning rate is restricted to positive values. This is due to preserve the compatibility between the MIT rule and the proposed approach. An appropriate selection of η_ϕ could be as follows:

$$\eta_\phi = \beta\min\left\{\left|\frac{1}{N_\phi}\Delta\phi\right|, \left|-\frac{1}{N_\phi}A_\phi\right|\right\},\ 0 < \beta < 1 \tag{49}$$

By substituting the learning rate formulated in (49) into the stabilizing solution given in (42), the stabilizing component $\Delta\phi_{VSS}$ of the parameter change formula is obtained as follows:

$$\Delta\phi_{VSS} = \beta\min\left(|\Delta\phi|, |A_\phi|\right)\mathrm{sgn}\left(N_\phi\right) + A_\phi \tag{50}$$

where, $\Delta\phi$ on the right-hand side is the final update value yet to be obtained. The law introduced in (50) minimizes the cost of stability, which is the Lyapunov function defined by (44). The question now reduces to the following; can the cost of realization defined by (36) be minimized by this rule? The answer is obviously not,

because the stabilizing information is derived from the displacement of parameter vector denoted by $\Delta\phi$, whereas the minimization of (36) is achieved when ϕ tends to ϕ^* (the optimal value of the parameter) regardless of what the displacement is. Therefore the rule formulated in (50) needs a final modification. In order to minimize (36), the parameter change anticipated by ordinary gradient rule, which has been reviewed in the second section, should somehow be integrated into the final form of parameter update mechanism. As introduced in the second section, error backpropagation (EBP) evaluates a parameter change as given by (51).

$$\Delta\phi_{EBP} = \zeta\, N_\phi \tag{51}$$

where, ζ is the learning rate in the conventional sense, i.e. it is chosen from the interval (0,1). In combining the laws formulated in (50) and (51), in a weighted average, the eventual form of the parameter update law is obtained as given in (52).

$$\Delta\phi = \frac{\alpha_1\Delta\phi_{VSS} + \alpha_2\Delta\phi_{EBP}}{\alpha_1 + \alpha_2} \tag{52}$$

The parameter update rule given by (52) meets the objectives of both the parametric stabilization and the cost minimization.

7. PLANT MODEL

In this study, a two degrees of freedom direct drive robotic manipulator, which is illustrated in Fig. 5, is used as the test bed. Since the dynamics of such a mechatronic system is modeled by nonlinear and coupled differential equations, precise output tracking becomes a difficult objective due to the strong interdependency between the variables involved. Furthermore, the ambiguities concerning the friction related dynamics in the plant model make the design much more complicated. Therefore the methodology adopted must use the methods of computational intelligence in some sense.

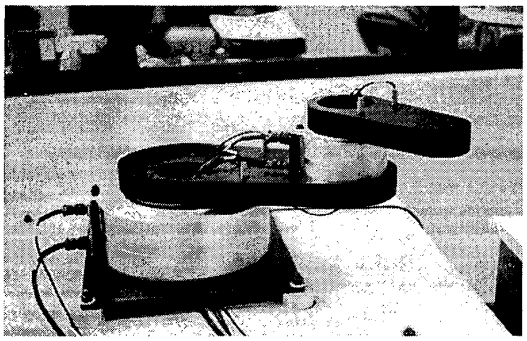

Figure 5. Physical View of the Direct Drive SCARA Robotic Manipulator

The general form of robot dynamics is described by (53) where M, V, τ and f_c stand for the state varying inertia matrix, vector of coriolis terms, applied torque inputs and friction terms respectively. The plant parameters are given in Table 1 in standard units.

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) = \tau - f_c \tag{53}$$

If the angular positions and angular velocities are defined as the state variables of the system, four coupled and first order differential equations can define the model in state space. In (54) and (55), the terms seen in (53) are given explicitly.

$$M(\theta) = \begin{bmatrix} p_1 + 2p_3\cos(\theta_2) & p_2 + p_3\cos(\theta_2) \\ p_2 + p_3\cos(\theta_2) & p_2 \end{bmatrix} \tag{54}$$

$$V(\theta,\dot{\theta}) = \begin{bmatrix} -\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2)p_3\sin(\theta_2) \\ \dot{\theta}_1^2 p_3\sin(\theta_2) \end{bmatrix} \tag{55}$$

In above, $p_1 = 2.0857$, $p_2 = 0.1168$ and $p_3 = 0.1630$. The details of the plant model are presented in [22].

Table 1. Manipulator Parameters

Motor 1 Rotor Inertia	0.2670	I1	Payload Mass	0.000	Mp
Arm 1 Inertia	0.3340	I2	Arm 1 length	0.359	L1
Motor 2 Rotor Inertia	0.0075	I3	Arm 2 length	0.240	L2
Motor 2 Stator Inertia	0.0400	I3C	Arm 1 CG	0.136	L3
Arm 2 inertia	0.0630	I4	Arm 2 CG	0.102	L4
Payload Inertia	0.0000	IP	Axis 1 Friction	5.300	F1
Motor 1 Mass	73.000	M1	Axis 2 Friction	1.100	F2
Arm 1 Mass	9.7800	M2	Torque Limit 1	245.0	
Motor 2 Mass	14.000	M3	Torque Limit 2	39.20	
Arm 2 Mass	4.4500	M4			

8. IDENTIFICATION AND CONTROL WITH NEURO FUZZY ARCHITECTURES

Identification and control of systems, in the most general sense, is a problem of realizing a multidimensional surface, which typically adopt the following realization strategies, in which x, x_d, and τ stand for the measured state, desired state and the applied inputs respectively.

a) $[\underline{x}(k), \underline{\tau}(k)] \Rightarrow [\underline{x}(k+1)]$
b) $[\underline{x}_d(k+1) - \underline{x}(k)] \Rightarrow [\underline{\tau}(k+1)]$

The former is aimed at the training of an identifier, while the latter is used in training of a controller utilizing the state tracking error vector.

In the identification of a system, the architecture illustrated in Fig. 6 is used. For the identification of a robotic manipulator, an excitation input is applied both the system under identification and the neuro-fuzzy model, which may be an architecture discussed before. Based on the discrepancy between the responses, the parameters of the identifier are adjusted such that the objectives of the problem are met.

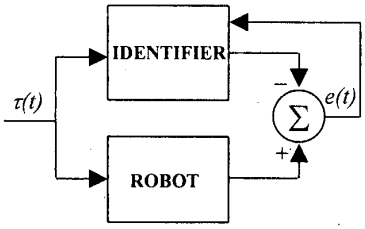

Figure 6. Identification Scheme

In Fig. 7, control of a robotic manipulator is illustrated. Based on the tracking error, controller produces the necessary torques to be applied at the next time instant while the trainer determines the strategy for the tuning of the adjustable controller parameters.

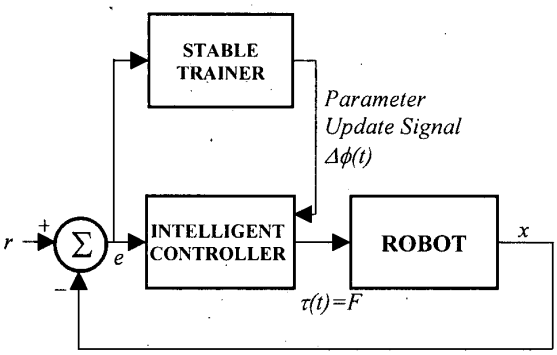

Figure 7. Control of a Robot Using the Proposed Training Method

9. SIMULATION RESULTS

The mentioned architectures and the training procedures have been applied to the manipulator dynamics considered in this tutorial. Only the results for the ANFIS architecture are presented due to the space limitations.

In the simulation studies presented, control of the manipulator by ANFIS architecture with mixed training approach is considered. The main objective is to keep the update dynamics in a stable region. This is achieved through a suitable combination of gradient based optimization technique and the strategy based on the variable structure systems approach. The control system is as illustrated in Fig. 7. The reference velocity in (56) is used in the simulations with zero initial errors.

$$\dot{\theta}_{d\,1,2} = \sin\left(\frac{2\pi t}{5}\right) \tag{56}$$

The results presented concern the tuning of all adjustable parameters of the ANFIS structure during the learning process. The state tracking errors are depicted in Fig. 8. It is evident from Fig. 8 that once a fluctuation occurs on the error or rate of error, it is dampened out by the use of VSC philosophy in the learning strategy. In the simulations discussed, the settings used are tabulated in Table 2.

Table 2. The Settings Used in the Simulations

T_s	2.5 msec.	Q	0.1
β	0.1	K	0.1
ζ	0.02	ε	1.0
α_{1i}	3.0 $\forall i$	#*Rules*	9
α_{2i}	2.0 $\forall i$	#ANFIS *Inputs*	2

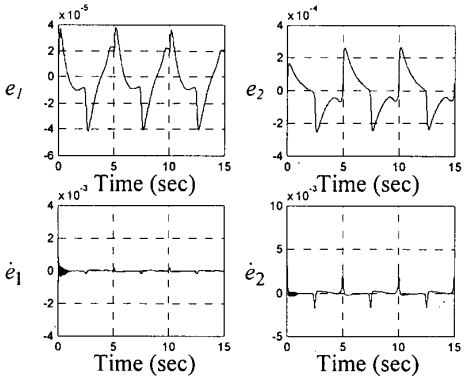

Figure 8. State Tracking Error Graph

In the training of the controllers, the squared sum of parametric changes can be defined to be the total cost of stability, which is described by (57).

$$J_s(t) = \sum_{\phi} (\Delta\phi(t))^2 \tag{57}$$

In Fig. 9, the cost of tracking described by (2) is illustrated whereas the bottom row depicts the time behavior of the parametric cost described by (57).

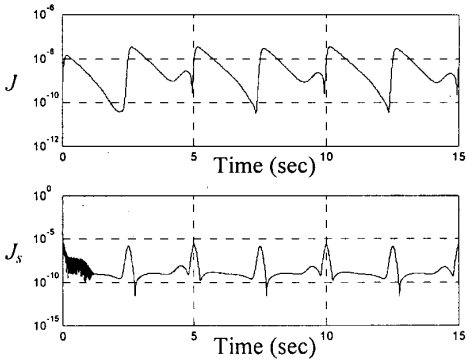

Figure 9. Time Behavior of the Realization Cost and Parametric Cost

As can be inferred from Fig. 9, both the parametric stabilization and the tracking error minimization performance of the proposed methodology is highly promising. Another remarkable property of the stabilizing algorithm presented is that it operates on-line. Therefore, the difficulties that are likely to occur in on-line learning and control are alleviated by the

robustness provided by the integration with VSS technique.

10. CONCLUSIONS

In this tutorial, various neuro-fuzzy architectures and their training procedures are discussed. A novel technique for improving learning performance of neuro-fuzzy systems is presented. An approximate dynamic model of the error backpropagation procedure is constructed and variable structure systems approach is incorporated into the model of the parameter update law. In this procedure, gradient descent method is responsible for the minimization of squared error while the variable structure systems based law is responsible for the stability in the parameter space.

The conventional approaches suffer from some handicaps, such as imperfect modeling, noisy observations or time varying parameters. If the effects of these factors are transformed to the cost hypersurface, whose dimensionality is determined by the adjustable design parameters, it is evident that the surface may have directions along which the sensitivity derivatives assume large values. In these cases, gradient based optimization procedures tend to evaluate large parametric displacements, which can eventually lead to a locally divergent behavior. In control engineering practice, such a behavior constitutes a potential danger from a safety point of view. The stabilizing approach presented in this paper takes care of the instantaneous fluctuations in parameter space. Since the variable structure systems approach is well known with its robustness property, an appropriate combination of error backpropagation technique and variable structure systems can eliminate the handicaps stated above. The fluctuations that are most likely to occur in the parameter space during training are dampened out. The combination is therefore a good candidate for efficient parameter tuning.

In the application examples, identification and control of a two degrees of freedom SCARA robot are considered. In the identification examples, only the error backpropagation technique is used, whereas in the control of the manipulator, parameter stabilizing training information is mixed with the gradient based training signal. The results observed clarify two facts: Firstly, the use of computational intelligence in identification and control applications enables the designer to incorporate his/her subjective judgements with learning machines. Secondly, the training methodology can be stabilized by the robust techniques known in the conventional design framework. The algorithm presented is applicable to any neuro-fuzzy system model provided that the model output is differentiable with respect to the parameter of interest.

11. ACKNOWLEDGMENTS

This work is supported by Bogazici University Research Fund (Project No: 99A202)

12. REFERENCES

[1] Hornik, K., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, v.2, pp 359-366, 1989.

[2] Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, v.2, pp 183-192, 1989.

[3] Gupta, M. M., D. H. Rao, "Dynamical Neural Units with Applications to the Control of Unknown Nonlinear Systems," *Journal of Intelligent and Fuzzy Systems*, v.1, no.1, pp. 73-92, 1993.

[4] Wang, L. X., *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*, PTR Prentice Hall, 1994

[5] Wang, L. X., *A Course in Fuzzy Systems and Control*, PTR Prentice Hall, 1997.

[6] Takagi, T., and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," IEEE Transactions on Systems, Man, and Cybernetics, v.15, no.1, pp.116-132, 1985.

[7] Jang, J.-S. R., C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, PTR Prentice Hall, 1997.

[8] Emelyanov, S. V., *Variable Structure Control Systems*, Moscow, Nauka, 1967.

[9] Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning Internal Representations by Error Propagation," in D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing*, v.1, pp.318-362, MIT Press, Cambridge, M.A., 1986.

[10] Narendra, K. S., K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, v.1, no. 1, pp.4-27, Mar. 1990.

[11] Efe, M. O., and O. Kaynak, "Stable Training of Computationally Intelligent Systems By Using Variable Structure Systems Technique," *IEEE Transactions on Industrial Electronics* (submitted for publication).

[12] Efe, M. O., and O. Kaynak, "A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems," Mechatronics Journal, Elsevier Science Ltd., v.9, no.3, 1999, pp.287-300, United Kingdom.

[13] Efe, M. O., and O. Kaynak, "A Comparative Study of Soft Computing Methodologies in Identification of Robotic Manipulators," *International Journal of Robotics and Autonomous Systems* (submitted for publication).

[14] Efe, M. O., and O. Kaynak, "Training of Artificial Neural Networks with Variable Structure Systems Approach and Gradient Descent Integration," *IEEE Transactions on Neural Networks* (submitted for publication).

[15] Efe, M. O., A. M. Fiskiran and O. Kaynak, "Synthesis of a Parameter Stabilizing Training Criteria for Adaptive Neuro-Fuzzy Inference Systems in Motion Control, Int. Journal of Systems Science (submitted for publication).

[16] Hung, J. Y., W. Gao, J. C. Hung, "Variable

Structure Control: A Survey," *IEEE Trans. On Industrial Electronics*, v.40, no. 1, pp.2-22, Feb. 1993.

[17] Gao, W., J. C. Hung, "Variable Structure Control of Nonlinear Systems: A New Approach," *IEEE Trans. on Industrial Electronics*, v.40, no. 1, pp.45-55, Feb. 1993.

[18] Kaynak, O., 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, v.104, no.3/4, pp.47-52, 1984.

[19] Bekiroglu, N., "Adaptive Sliding Surface Design for Sliding Mode Control Systems," Ph.D. Thesis, Bogazici University, 1996.

[20] Byungkook, Y., W. Ham, "Adaptive Fuzzy Sliding Mode Control of Nonlinear Systems," *IEEE Trans. on Fuzzy Systems*, v.6, no.2, pp.315-321, May 1998.

[21] Erbatur, K., O. Kaynak, A. Sabanovic and I. Rudas, "Fuzzy Adaptive Sliding Mode Control of a Direct Drive Robot," *Robotics and Autonomous Systems*, v.19, no:2, pp.215-227, Dec. 1996.

[22] Direct Drive Manipulator R&D Package User Guide, Integrated Motions Incorporated, 704 Gillman Street, Berkeley, California 94710, U.S.A.