Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

A Novel Computationally Intelligent Architecture for Identification and Control of Nonlinear Systems

[1]M. Önder Efe, [2]Okyay Kaynak and [3]Imre J. Rudas
[1,2]Mechatronics Research and Application Center, Bogaziçi University, Bebek, 80815, Istanbul, Turkey
[3]Dept. of Information Technology, Banki Donat Polytechnic, H-1081, Budapest, Nepszinhaz u.8, Hungary

Abstract - In this study, a novel method for identification and control of nonlinear systems is developed. The method proposed realizes the dynamics of a system by employing the Runge-Kutta method at the upper level. The intermediate level of the strategy constructs the architecture utilizing an adaptive neuro fuzzy inference system. The overall system is able to imitate the behavior of a complex dynamic system with a few rules or to control the system with high accuracy. The proposed method has been applied to a two degrees of freedom direct drive SCARA robot.

1. Introduction

Identification and control of nonlinear systems have been studied by many researchers and we have, during the last decade, witnessed distinguished solutions for specific problems. The approaches integrating intelligence and smart numerical analysis are especially worthy of attention in this respect as they have resulted in hybrid architectures capable of achieving high performance. The success of intelligent control systems is commonly attributed to constituents of the methodology providing intelligence. Neural Networks (NN) and Fuzzy Inference Systems (FIS) are two of these constituents leading to the fusion of verbal processing of data together with small scale brain-like activity provided by artificial neuron models. Various architectures of NN and FIS have been studied for years. In [1], Narendra and Parthasarathy have shown that NN architectures can easily be used for system identification and control purposes. In their award winning paper, the system nonlinearities have been assumed to be represented by NN models. This has resulted in an easier construction of control signals and led to a robust closed loop system in the qualitative sense. On the other hand, fuzzy systems have been shown to be able to substitute the human controller with their property of representing the actions on the basis of linguistic variables. In [2] and [3], various methods have been taken into consideration for identification of a direct drive and an anthropoid robotic manipulator having two and three degrees of freedom respectively.

In this study, Adaptive Neuro Fuzzy Inference Systems (ANFIS) have been chosen as the core of the approach. A computationally intelligent architecture is achieved by integrating the ANFIS structure with the Runge-Kutta method. The original form of ANFIS architecture has explained analytically in [4] and has drawn a great interest due to its extensive design flexibility. The mathematical analysis of ANFIS architecture clearly implies that many FIS models can be realized by ANFIS architecture by appropriately setting the parameters. The upper level of the architecture introduced in this paper employs the Runge-Kutta method, which is a powerful way of solving the behavior of systems modeled by ordinary differential equations. In [5], the method is combined with radial basis function neural networks. Wang and Lin [5], has shown that the method is highly sucessful in estimating the behavior of a system in the long run. In [2] and [3], Efe and Kaynak have combined the method with ordinary Feedforward Neural Networks (FNN) and realized the identification of both a two dof direct drive SCARA and a three degrees of freedom anthropoid robotic manipulator with on-line tuning of the neural network stage parameters. This paper replaces the ordinary FNN architecture with ANFIS and uses the resulting architecture for the identification and control of a two degrees of freedon direct drive SCARA manipulator. The parameter update mechanism operates on-line.

The organization of this paper is as follows. The next section describes the plant to be identified and controlled. The third section introduces ANFIS structure which functions as a single stage of the proposed architecture. The fourth section explains how ANFIS architecture could be embedded into the Runge-Kutta method. The evaluation of the equivalent output error from plant outputs to controller outputs is briefly explained, the equivalent stage errors for ANFIS structure are evaluated and the update mechanism is derived. The fifth section discusses the simulation results and finally the sixth section makes the concluding remarks on the basis of the results obtained.

2. Two DOF Direct Drive SCARA Robot Dynamics

Robotic manipulators are appropriate candidates for performance evaluation of computationally intelligent identification and control methods because the coupled nonlinear equations and ambiguities on the friction related dynamics inevitably require the use of flexible control architectures. This necessity becomes more apparent if high tracking precision is sought. The general form of the dynamics of the manipulator considered in this paper is given by (1) and the nominal values of the parameters are summarized in Table 1 in standard units. The physical view of the manipulator is illustrated in Fig. 1.

$$M(\theta)\theta'' + V(\theta,\theta') = \tau(t) - f \qquad (1)$$

By assuming the angular positions and angular velocities as state variables of the system, a total of four 1^{st} order differential equations are obtained. The state varying inertia matrix and coriolis terms are given in (2) and (3) respectively.

0-7803-5180-0-5/99 $10.00 © 1999 IEEE

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

$$V(\theta,\theta')=\begin{bmatrix} -\theta_2'(2\theta_1'+\theta_2')p_3\sin(\theta_2) \\ \theta_1'^2 p_3\sin(\theta_2) \end{bmatrix} \quad (3)$$

where $p_1 = 2.0857$, $p_2 = 0.1168$, $p_3 = 0.1630$.

Table 1. Manipulator Parameters

Motor 1 Rotor Inertia	0.267	I1	Payload Mass	0.00	Mp
Arm 1 Inertia	0.334	I2	Arm 1 length	0.359	L1
Motor 2 Rotor Inertia	0.0075	I3	Arm 2 length	0.24	L2
Motor2 Stator Inertia	0.040	I3C	Arm 1 CG	0.136	L3
Arm 2 inertia	0.063	I4	Arm 2 CG	0.102	L4
Payload Inertia	0.000	IP	Axis 1 Friction	5.3	F1
Motor 1 Mass	73.0	M1	Axis 2 Friction	1.1	F2
Arm 1 Mass	9.78	M2	Torque Limit 1	245.0	
Motor 2 Mass	14.0	M3	Torque Limit 2	39.2	
Arm 2_Mass	4.45	M4			

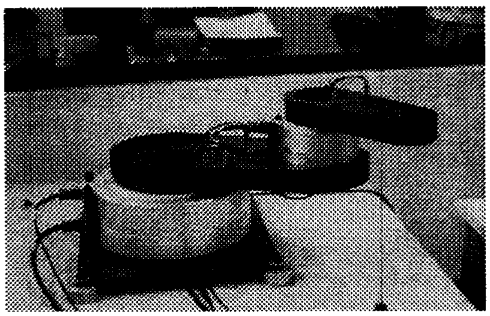

Figure 1. Physical View of the Direct Drive SCARA

3. Adaptive Neuro Fuzzy Inference Systems

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 mechanisms 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 five 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. The resulting vector is defuzzified by utilizing the first order Sugeno model. The procedure is briefly formulated in (4) through (8) for the ANFIS architecture illustrated in Fig. 2, with the construction of a simple rule base being as follows:

IF x is A_1 and y is B_1 THEN $f_1=p_1x+q_1y+r_1$
IF x is A_2 and y is B_2 THEN $f_2=p_2x+q_2y+r_2$

$$w_1=\mu_{A_1}(x)\mu_{B_1}(y) \quad (4)$$

$$w_2=\mu_{A_2}(x)\mu_{B_2}(y) \quad (5)$$

$$\underline{w}_1=\frac{w_1}{w_1+w_2} \quad (6)$$

$$\underline{w}_2=\frac{w_2}{w_1+w_2} \quad (7)$$

$$f=\underline{w}_1(p_1x+q_1y+r_1)+\underline{w}_2(p_2x+q_2y+r_2) \quad (8)$$

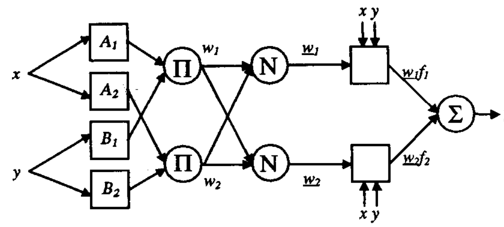

Figure 2. ANFIS Architecture

The ANFIS output is clearly a linear function of the adjustable defuzzifier parameters. At the adjustment of $[\underline{p}\ \underline{q}\ \underline{r}]^T$ vector, gradient descent method is applied. For the identification of two dof manipulator considered in this paper, the fuzzifier possesses six inputs, the rule base contains only five rules and the defuzzifier has four outputs. In the case of the proposed structure being used as a controller, it uses the same number of inputs and rules but two outputs.

4. Integration of Runge-Kutta Method and ANFIS Architecture

Runge-Kutta method is a powerful way of solving the behavior of a dynamic system if the system is characterized by ordinary differential equations. In [5], the proposed method is applied to several problems and it is seen that the method is successful in estimating the system states given long enough time. It should be emphasized that the ANFIS architecture realizes the changing rates of the system states instead of the $[\underline{x}(k),\underline{\tau}(k)]\Rightarrow[\underline{x}(k+1)]$ mapping that is aimed at with conventional neural identifiers. As known, the first order discretization brings large approximation errors whereas Runge-Kutta realizes a much better solution. Therefore, the integration of Runge-Kutta and ANFIS is expected to bring about a much better performance than those obtained in [2], [3] and [5]. Wang and Lin [5] have studied this concept with radial basis function neural networks (RBFNN) and trained the architecture for observed data whereas an on-line approach is adopted in this paper. The proposed architecture is illustrated in Fig. 3. In this figure, h denotes the Runge-Kutta integration stepsize, which has been set to 2.5 ms for all simulations.

Robot dynamics of (1) can be stated more compactly as in (9). The vector function f is realized by the ANFIS structures depicted in Fig. 3. For a fourth order Runge-Kutta approximation, the overall scheme is comprised of an ANFIS block being repeatedly connected four times with the shown stage gains. The update mechanism is based on the error backpropagation. The derivation for ANFIS based identification scheme is given in (10) through (14) where N represents the ANFIS stages of the architecture and ϕ is a generic parameter of ANFIS.

$$\underline{x}'=\underline{f}(\underline{x},\underline{\tau}) \quad (9)$$

$$\underline{x}(i+1)=\underline{x}(i)+\frac{h}{6}\left(\underline{k}_0+2\underline{k}_1+2\underline{k}_2+\underline{k}_3\right) \quad (10)$$

$$\underline{k}_0 = N(\underline{x};\phi) \qquad = N(\underline{x}_0;\phi) \tag{11}$$

$$\underline{k}_1 = N(\underline{x}+\frac{1}{2}h\underline{k}_0;\phi) = N(\underline{x}_1;\phi) \tag{12}$$

$$\underline{k}_2 = N(\underline{x}+\frac{1}{2}h\underline{k}_1;\phi) = N(\underline{x}_2;\phi) \tag{13}$$

$$\underline{k}_3 = N(\underline{x}+h\underline{k}_2;\phi) \ = N(\underline{x}_3;\phi) \tag{14}$$

Figure 3 clarifies how the error backpropagation rule is applied. There are two paths to be considered in this propagation. The first is the direct connection to the output summation, the other is through the ANFIS stages of the architecture. Therefore, each derivation, except the first one, will concern two terms. The rule is summarized below for the fourth order Runge-Kutta approximation.

$$\frac{\partial \underline{k}_0}{\partial \phi} = \frac{\partial N(\underline{x}_0;\phi)}{\partial \phi} \tag{15}$$

$$\frac{\partial \underline{k}_1}{\partial \phi} = \frac{\partial \underline{k}_1}{\partial \underline{x}_1}\frac{\partial \underline{x}_1}{\partial \underline{k}_0}\frac{\partial \underline{k}_0}{\partial \phi} + \frac{\partial \underline{k}_1}{\partial \phi} \tag{16}$$

$$\frac{\partial \underline{k}_2}{\partial \phi} = \frac{\partial \underline{k}_2}{\partial \underline{x}_2}\frac{\partial \underline{x}_2}{\partial \underline{k}_1}\frac{\partial \underline{k}_1}{\partial \phi} + \frac{\partial \underline{k}_2}{\partial \phi} \tag{17}$$

$$\frac{\partial \underline{k}_3}{\partial \phi} = \frac{\partial \underline{k}_3}{\partial \underline{x}_3}\frac{\partial \underline{x}_3}{\partial \underline{k}_2}\frac{\partial \underline{k}_2}{\partial \phi} + \frac{\partial \underline{k}_3}{\partial \phi} \tag{18}$$

$$\Delta\phi(i) = \frac{\eta h}{6}\left(\underline{d}^T(i) - \underline{x}^T(i)\right)\left(\frac{\partial \underline{k}_0}{\partial \phi} + 2\frac{\partial \underline{k}_1}{\partial \phi} + 2\frac{\partial \underline{k}_2}{\partial \phi} + \frac{\partial \underline{k}_3}{\partial \phi}\right) \tag{19}$$

In (19), η represents the learning rate and $\underline{d}(i)$ represents the measured state vector of the plant at time index i. The identification scheme is illustrated in Fig. 4. In this part of the study, the robot manipulator is kept under an external control loop while the identifier performance is being tested. Simulation results obtained are shown in Figs. 7 and 8, the reference signal of both axes being as described by (20).

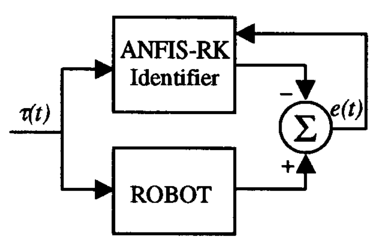

Figure 4. Identification Scheme

$$r(t) = tanh(3(t-2)) - tanh(3(t-6)) \tag{20}$$

The second part of this study analyzes the control of the manipulator. The control architecture is illustrated in Fig. 5, in which the shown controller has the structure based on an ANFIS-RK integration as depicted in Fig. 6.

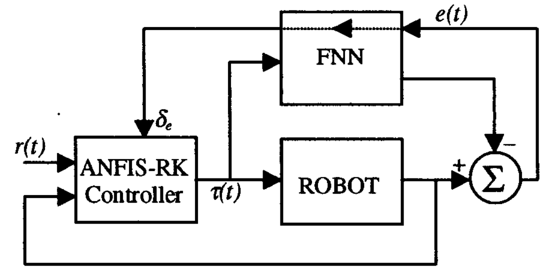

Figure 5. Control of Robotic Manipulator

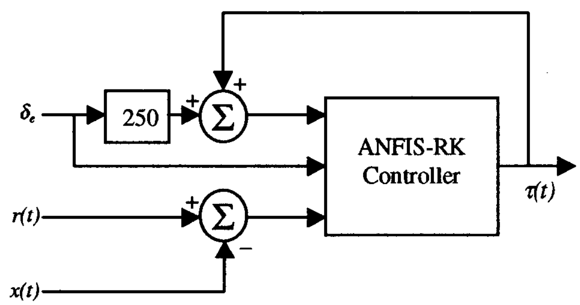

Figure 6. Controller Structure

To achieve an appropriate control signal, the error in the applied control signal must be determined. This is achieved through the utilization of a pre-trained ordinary FNN. Since the desired and actual values of the state vector are measurable quantities, the output error can be evaluated and can be propagated back through the FNN model until the controller outputs are reached as shown in Fig. 5. This is summarized as follows.

$$\delta_j^{k+1,p} = (d_j^p - x_j^{k+1,p})\Psi'(S_j^{k+1,p}) \tag{21}$$

$$\delta_j^{k+1,p} = \left(\sum_{h=1}^{\#neuron_{k+2}} \delta_h^{k+2,p} w_{jh}^{k+1}\right)\Psi'(S_j^{k+1,p}) \tag{22}$$

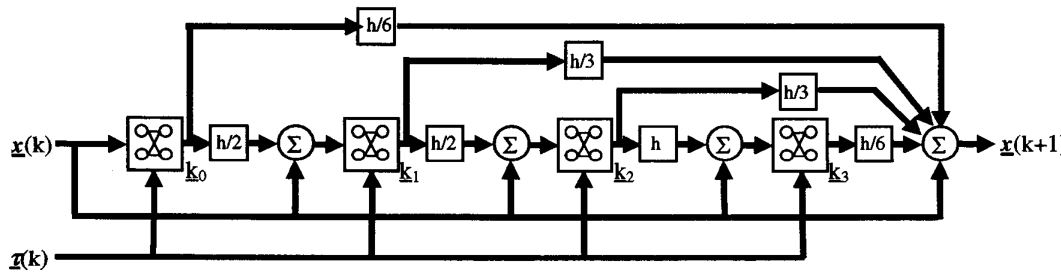

Figure 3. Runge-Kutta ANFIS Hybrid Architecture

Equation (21) describes the delta values for the output layer. The term $x_j^{k+1,p}$ denotes the j[th] entry of p[th] pattern in FNN response, d_j^p denotes the j[th] entry of p[th] target vector. In (21), delta values for the hidden layers are formulated. In this equation $S_j^{k+1,p}$, Ψ and $\delta_j^{k+1,p}$ denote the evaluated weighted sum, neuronal activation function and the delta value attached to j[th] neuron in the (k+1)[th] layer respectively. For on-line adjustment mechanism, p corresponds to the time index. For updating the parameters of controller architecture, the same sensitivity derivatives as given in (15) through (18) are used. However, in the updating rule described by (19), an equivalent error δ_e is used which is obtained by propagating the output error back through the FNN.

The next step in formulation of the scheme is the derivation of the parameter update for the controller itself. Since the plant to be controlled is a MIMO one with two inputs and four outputs, the ANFIS architecture of the controller is much more complicated than that illustrated in Fig. 2. Now, the matrix multiplying the vector of normalized firing strengths is termed WR instead of using p, q, r terms. Having this in mind, the sensitivity derivatives of a single ANFIS stage can be evaluated as follows.

$$\frac{\partial f_s}{\partial \tau_j} = \frac{\partial f_s}{\partial \tau_j} + \frac{\partial f_s}{\partial \underline{w}} \frac{\partial \underline{w}}{\partial \tau_j} \tag{23}$$

where f_s is the s[th] output and τ_j is the j[th] input of ANFIS stages. With these definitions, the s[th] output of ANFIS can be reexpressed as given by (24).

$$f_s = (WR_s \underline{\tau}_a)^T \underline{w} \tag{24}$$

In above, $\underline{\tau}_a$ denotes the augmented input. This is clear from (8) where r_j terms are multiplied by unity. Therefore, in the final evaluations, this is taken into consideration and the related matrices are diminished.

$$\frac{\partial f_s}{\partial \underline{\tau}_a} = (\underline{w}^T WR_s) \tag{25}$$

$$\frac{\partial f_s}{\partial \underline{w}} = (WR_s \underline{\tau}_a)^T \tag{26}$$

$$\frac{\partial w_k}{\partial \tau_j} = 2 \sum_{i=1}^{RULES} \left(-\frac{w_k^2}{w_k} + \delta(k,i) \frac{w_k}{w_k} \right) w_i \frac{c_{ij} - \tau_j}{\sigma_{ij}^2} \tag{27}$$

If these terms are combined as formulated by (23), each ANFIS architecture will propagate the equivalent stage errors as imposed by the architecture of Runge-Kutta method. In (27), c_{ij} and σ_{ij} are the center and width values for the i[th] rules's j[th] membership function, $\delta(k,i)$ denotes the Kronecker delta function. The controller can now be formulated as follows.

$$\underline{\tau}(i+1) = \underline{\tau}(i) + \frac{h}{6}\left(\underline{k}_0 + 2\underline{k}_1 + 2\underline{k}_2 + \underline{k}_3\right) + \lambda \delta_e \tag{28}$$

In (28), δ_e is the equivalent control error provided by FNN identifier, λ is a constant and during the simulations this has been tuned to 250. The last term acts as a correcting term. When implementing this controller, it should be emphasized that the controller inputs are composed of its previous outputs and the state tracking errors. Simulation results for the control problem are illustrated in Figs. 9 and 10 with the reference signal defined as (20).

5. Discussion of the Simulation Results

Two sets of simulation studies have been carried out in this study. In the first part, the identification of the robotic manipulator is considered. For the identification problem, the identifier possesses six inputs and four outputs. The Runge-Kutta ANFIS architecture is supposed to estimate the state vector of the plant. While the estimation process is going on, the manipulator is kept under an external control loop. As the positional reference signal, a smooth pulse, which is defined by (20), has been chosen for both axes.

The second part of the paper concerns the control of manipulator. In this part, an additional FNN identifier is used to obtain the equivalent error on applied controls. This is a widely used approach because in general, one has the observed data instead of the exact model of the plant. Therefore the training of such a FNN model for equivalent control error evaluation is a reasonable approach. This error measure is then used to adjust the parameters of ANFIS stages by appropriately propagating it back through the ANFIS structure. It must be emphasized that the ANFIS structures embedded into the Runge-Kutta method realize the same vector function. Therefore once the error backpropagation through all four stages is completed, the update rule applies in a cumulative manner.

In Fig. 7, the reference position and velocities are illustrated in the top and bottom rows respectively. Figure 8 shows the performance of Runge-Kutta ANFIS integration in identification of the manipulator model. The simulation results obtained with an FNN identifier and ANFIS-RK controller are given in Figs. 9 and 10. The signal, described by (20) and shown in Fig. 7 is used as the reference state trajectory. The state tracking errors are depicted in Fig. 9 and it is seen that the controller has good tracking capability as indicated by the bounds of tracking errors.

In the top row of Fig. 10, the produced control signals are illustrated. The bottom row shows the equivalent delta values evaluated during operation.

In all simulations, the rule base of ANFIS contains only five rules. The on-line operation and the few number of rules required make the proposed method attractive.

6. Conclusions

A novel computationally intelligent method is developed in this paper. The performance of the proposed approach has been tested on a two degrees of freedom direct drive robotic manipulator. The results obtained indicate that the proposed architecture is a good candidate both for identification and control purposes.

The main advantage of the method is the compactness of the rulebase and on-line adjustment of parameters. The only drawback is the computational cost. The work is in progress for reducing the computational requirements of the architecture.

7. Acknowledgments

This work is supported in part by a grant of Foundation for Promotion of Advanced Automation Technology.

References

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

[2] Efe, M. O. and O. Kaynak, "A Comparative Study of Soft Computing Methodologies in Identification of Robotic Manipulators," Proc. 3rd Int. Conf. on Advanced Mechatronics, ICAM'98, 3-6 August, vol. 1, pp. 21-26, Okayama, Japan, 1998.

[3] Efe, M. O. and O. Kaynak, "A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems," *Int. Journal of Mechatronics*, (accepted for publication) 1998.

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

[5] Wang, Y-J. and C-T. Lin "Runge-Kutta Neural Network for Identification of Dynamical Systems in High Accuracy," *IEEE Transactions on Neural Networks*, vol. 9., no. 2, pp. 294-307, March 1998.

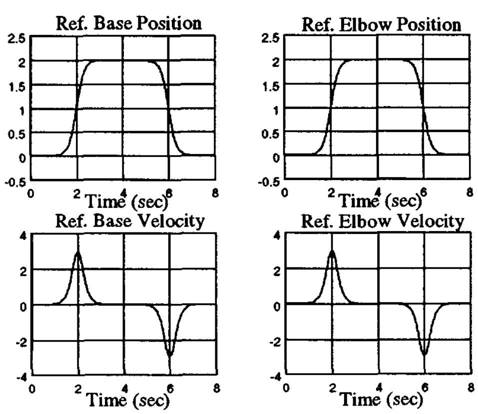

Figure 7. Reference Position and Velocity Profiles

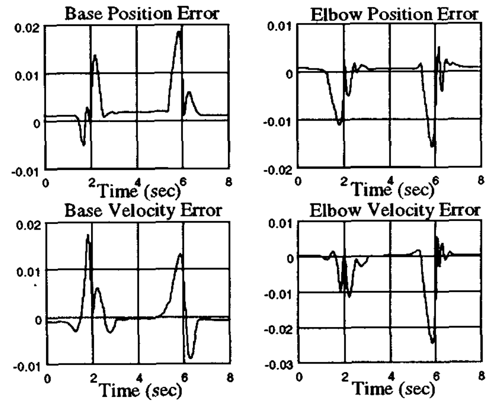

Figure 8. Estimation Errors in Identification of Robotic Manipulator

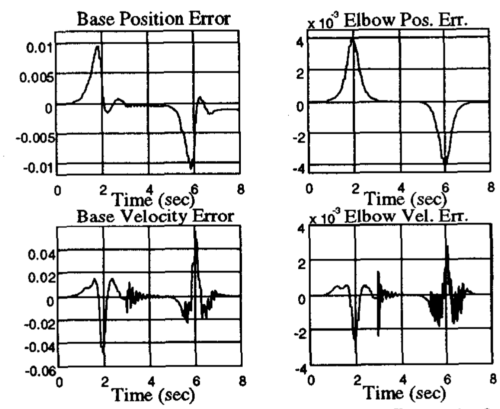

Figure 9. State Tracking Errors in Control of Robotic Manipulator

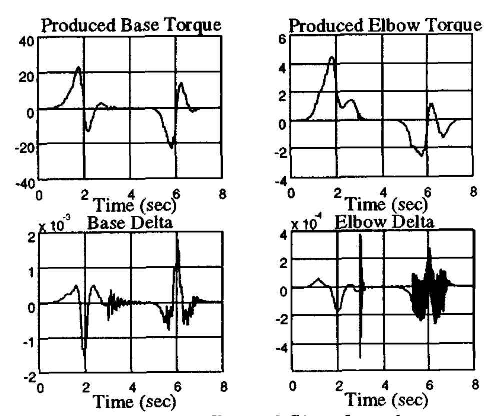

Figure 10. Produced Control Signal on the top row and Equivalent Deltas on the bottom row for Control of Robotic Manipulator