PERGAMON

Mechatronics 9 (1999) 287–300

MECHATRONICS

A comparative study of neural network structures in identification of nonlinear systems

M.O. Efe, O. Kaynak*

Mechatronics Research and Application Center, Bogaziçi University, Bebek, 80815 Istanbul, Turkey

Received 30 June 1998; accepted 7 August 1998

Abstract

This paper investigates the identification of nonlinear systems by neural networks. As the identification methods, Feedforward Neural Networks (FNN), Radial Basis Function Neural Networks (RBFNN), Runge–Kutta Neural Networks (RKNN) and Adaptive Neuro Fuzzy Inference Systems (ANFIS) based identification mechanisms are studied and their performances are comparatively evaluated on a three degrees of freedom anthropomorphic robotic manipulator.

1. Introduction

System identification has become an important area of study because of the increasing need to estimate the behavior of a system with partially known dynamics. Especially in the areas of control, pattern recognition and even in the realm of stock markets, the system of interest needs to be known to some extent. A common property of real life systems is the fact that they have multiple variables, some of which are subjected to stochastic disturbances. Since a system may have a complicated dynamic behavior, the varying environmental changes make the identification process much-more difficult than the cases in which those changes are modeled deterministically. In the latter, the identification is performed at the cost of losing the reliability and preciseness. Therefore, deciding on what the future behavior of a system will be and performing an adaptive estimation for this purpose are formidable problems in real life systems.

Soft computing is a practical alternative for solving computationally complex and mathematically intractable problems. The reason behind this is the fact that through the use of soft-computing methodologies, one can easily combine the natural system

* Corresponding author. Tel.: 90 212 287 2475; fax: 90 212 287 2465; e-mail: kaynak@boun.edu.tr

0957-4158/99/$ - see front matter
PII: S0957-4158(98)00047-6

dynamics and an intelligent machine. In this perspective, the intelligence comes about through the utilization of an expert's knowledge and massively parallel and adaptive data processing architecture of the computationally intelligent approach. The most popular constituents of the soft-computing methodologies are the neural networks and fuzzy logic. Neural networks provide the mathematical power of the brain whereas the fuzzy logic based mechanisms employ the verbal power. The latter allows the linguistic manipulation of input-state-output data. The most interesting applications propose an appropriate combination of these two approaches resulting in a hybrid system that both operates on linguistic descriptions of the variables and the numeric values through a parallel and fault tolerant architecture.

The mapping properties of artificial neural networks have been analyzed by many researchers. Hornik [1], and Funahashi [2] have shown that as long as the hidden layer comprises a sufficient number of nonlinear neurons, a function can be realized with a desired degree of accuracy. This proof is followed by the study of Narendra and Parthasarathy [3]. In their pioneering work, they have debated how useful artificial neural networks can be for identification and control purposes. Their paper dwells on the realization of an unknown nonlinearity by artificial neural networks. The training is performed by the error backpropagation algorithm [4]. On the other hand, a novel approach has been presented in [5] where the neural network realizes the behavior of a set of ordinary differential equations utilizing the Runge–Kutta algorithm. The method is proved to be successful in predicting the future behavior accurately. In [6], Radial Basis Function Neural Networks are explained with their functional equivalence to Fuzzy Inference Systems (FIS). In the same reference, the details of Adaptive Neuro Fuzzy Inference Systems (ANFIS) structure can be found, proposed as a core neuro-fuzzy model that can incorporate human expertise as well as adapt itself through repeated learning. This architecture has revealed a high performance in many applications. In this work, a number of structures are considered for the identification of nonlinear systems. A three degrees of freedom anthropomorphic robotic manipulator is considered as a test-case.

The identification procedure followed in all methods considered is as depicted in Fig. 1. An excitation input is applied to both the robot (nonlinear) system and the identifier. The output error is then used to update the parameters of the identifier. The excitation input cannot be selected randomly for any real system. Therefore, in the simulations, the manipulator is kept under an external control loop while the identifier performance is being tested.

In the next section the system under observation is introduced, the following section dwells on FNN based identification schemes. Next, RBFNN approach is derived. The fifth section describes the application of the Runge–Kutta neural network methodology for the identification of robotic manipulators. In the sixth section ANFIS structure is elaborated. Lastly the conclusions are presented.

2. Plant model

A robotic manipulator is a proper candidate for the evaluation of the performance of identification mechanisms stated before. The main reason for this is the fact that

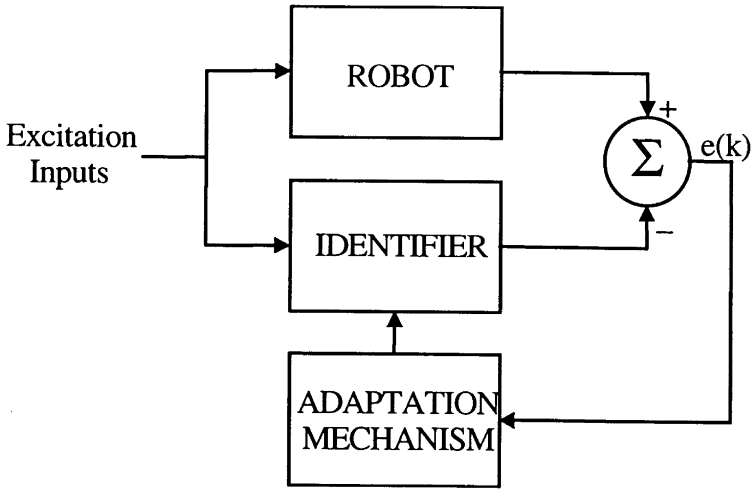

Fig. 1. Identification of a robotic manipulator.

the dynamics are highly involved, being comprised of coupled nonlinear differential equations. Furthermore, the dynamics are subjected to gravity effects. The general form of the manipulator dynamics is given by Eq. (1) and the nominal values of the parameters are summarized in Table 1 where the values are in standard units. By adopting the angular positions and angular velocities as state variables of the system, a total of six first order differential equations are formulated. The physical structure of the manipulator is illustrated in Fig. 2.

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=u \tag{1}$$

Table 1
Manipulator parameters

Link 1 length	0.5	l_1
Link 2 length	0.4	l_2
Link 3 length	0.4	l_3
Link 1 mass	4	m_1
Link 2 mass	3	m_2
Link 3 mass	3	m_3
Distance link 1 CG-joint 1	0.2	l_{c1}
Distance link 2 CG-joint 2	0.2	l_{c2}
Cylindrical link radius	0.05	R
ith Cylindrical link inertial parameter	$E_1=m_1R^2/2$, $E_i=m_il^2_i/12$	E_i
ith Cylindrical link inertial parameter	$A_i=m_iR^2/2$	A_i
ith Cylindrical link inertial parameter	$I_i=m_il^2_i/12$ for $i=2, 3$	I_i

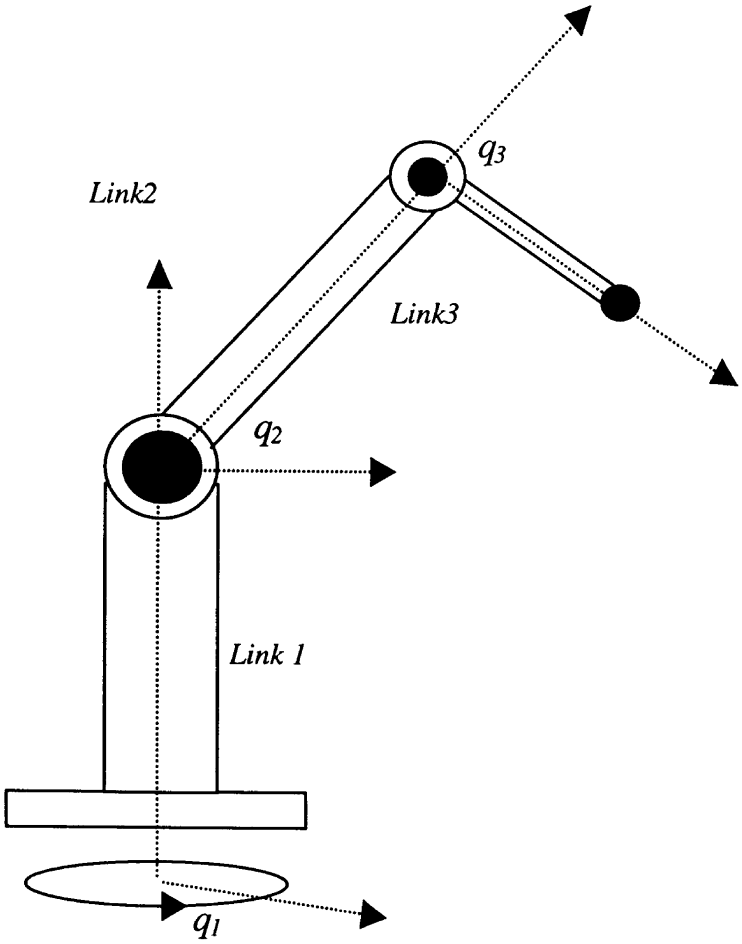

Fig. 2. Architecture of 3 DOF anthropoid robot manipulator.

$$M_{11} = m_2 l_{c2}^2 \cos^2(q_2) + m_3 (l_2 \cos(q_2) + l_{c3} \cos(q_2 + q_3))^2 + E_1 + A_2 \sin^2(q_2) + E_2 \cos^2(q_2) + A_3 \sin^2(q_2 + q_3) + E_3 \cos^2(q_2 + q_3) \quad (2)$$

$$M_{22} = m_2 l_{c2}^2 \sin^2(q_2) + m_3 (l_2^2 + l_{c3}^2 + 2 l_2 l_{c3} \cos(q_3)) + I_2 + I_3 \quad (3)$$

$$M_{23} = M_{32} = m_3 (l_{c3}^2 + l_{c3} l_2 \cos(q_3)) + I_3 \quad (4)$$

$$M_{33} = m_3 l_{c3} + I_3 \quad (5)$$

Equation (1) describes the dynamics and Eqs. (2)–(5) are the non-zero entries of the state varying inertia matrix M. The nonzero Cristoffel symbols (defined in [7] and [8]) are obtained as follows:

$$hc_1 = (-m_2 l_{c2}^2 + A_2 - E_2)\cos(q_2)\sin(q_2) + (A_3 - E_3)\cos(q_2+q_3)\sin(q_2+q_3) + m_3(l_2\cos(q_2) + l_{c3}\cos(q_2+q_3))(-l_2\sin(q_2) - l_{c3}\sin(q_2+q_3)) \quad (6)$$

$$hc_2 = \sin(q_2+q_3)(-m_3 l_{c3} l_2\cos(q_2) + (-m_3 l_{c3}^2 + A_3 - E_3)\cos(q_2+q_3)) \quad (7)$$

$$hc_3 = m_2 l_{c2}^2\cos(q_2)\sin(q_2) \quad (8)$$

$$hc_4 = -m_2 l_2 l_{c3}\sin(q_3) \quad (9)$$

Coriolis and centrifugal terms are formulated as

$$C(q,\dot{q}) = \begin{bmatrix} 2hc_1\dot{q}_1\dot{q}_2 + 2hc_2\dot{q}_1\dot{q}_3 \\ -hc_1\dot{q}_1^2 + 2hc_4(\dot{q}_2\dot{q}_3 + \dot{q}_3^2) + hc_3\dot{q}_2^2 \\ -hc_2\dot{q}_1^2 - hc_4\dot{q}_2^2 \end{bmatrix} \quad (10)$$

Lastly, the gravity terms are obtained as given in Eq. (11) where G represents the gravity constant.

$$g(q_1,q_2,q_3) = \begin{bmatrix} 0 \\ (m_2 l_{c2} + m_3 l_2)G\cos(q_2) + m_3 l_{c3} G\cos(q_2+q_3) \\ m_3 l_{c3} G\cos(q_2+q_3) \end{bmatrix} \quad (11)$$

2. Feedforward neural networks (FNN) for system identification

Identification procedure, in the most general sense, entails a matching between the system outputs and an identifier output. As is stated in [1] and [2], artificial neural networks can be used as universal approximators. In [3], Narendra and Parthasarathy use these networks for identification and control purposes. Based on the diagram sketched in Fig. 1, the cost function given in Eq. (12) is minimized by propagating the output error back through the neural network, which is illustrated in Fig. 3. The update equations will not be derived here, for further details, [4] must be reviewed.

$$J = \frac{1}{2}\sum_{i=1}^{N}(d_i^p - y_i^p)^2 \quad (12)$$

In (12), $y^p{}_i$ denotes the ith entry of pth pattern in neural network response, $d^p{}_i$ denotes the ith entry of pth target vector. Eqs. (13) and (14) give the delta values for the output layer and hidden layer neurons respectively.

$$\delta_j^{k+1,p} = (d_j^p - y_j^{k+1,p})\Psi(S_j^{k+1,p}) \quad (13)$$

$$\delta_j^{k+1,p} = \left(\sum_{h=1}^{\#\text{neurons}_{k+2}} \delta_h^{k+2,p} w_{jh}^{k+1}\right)\Psi'(S_j^{k+1,p}) \quad (14)$$

In Eqs. (13) and (14), S_j denotes the net summation of the jth neuron in the $(k+1)$th

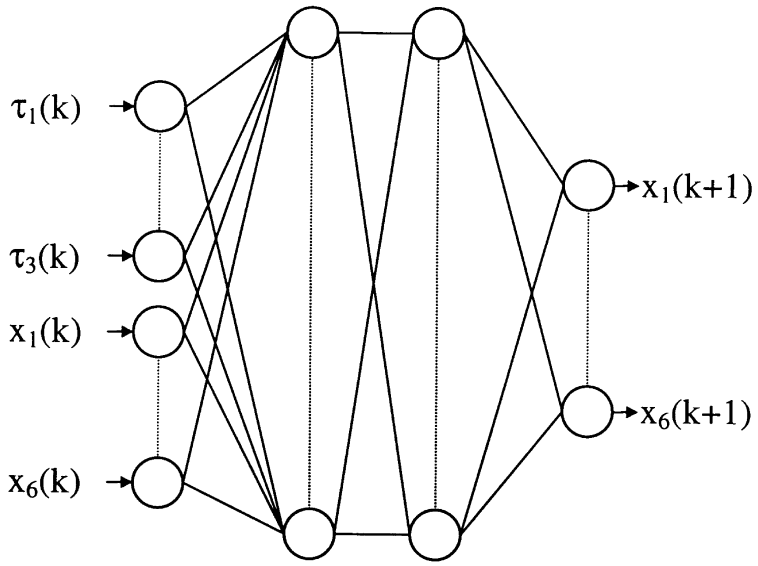

Fig. 3. FNN architecture.

layer, ψ denotes the nonlinear activation function attached to each neuron in the hidden layer. Having evaluated the delta values during the backward pass, the weight update rule given in Eq. (15) is applied for each training pair.

$$\Delta w_{ij}^{k} = \eta \delta_j^{k+1,p} o_i^{k,p} \tag{15}$$

In order to evaluate the performance of the FNN based identifier, simulations are performed on a 3-dof anthropomorphic robotic manipulator model described above. The reference trajectories used are based on trapezoidal velocity profiles, the parameters of which are selected by considering the limitations of the work space. The error ($e(k)$ in Fig. 1) waveforms for FNN architecture are given among the last group of figures of this paper as Fig. 7, where only the positional error trends are sketched.

4. Radial basis function neural networks (RBFNN) for system identification

In the literature, RBFNNs are generally considered as a smooth transition between fuzzy logic and neural networks. Structurally, RBFNNs 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 [6]. In Fig. 4, 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 combiners.

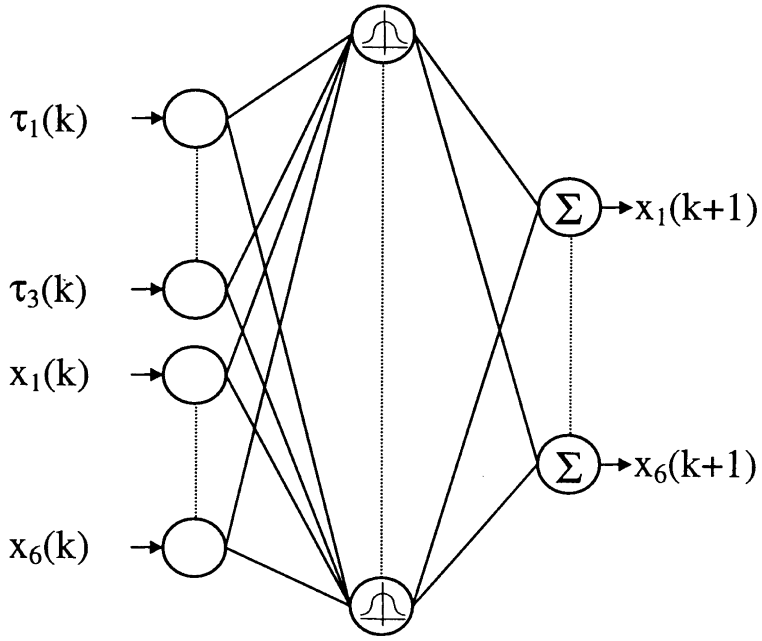

Fig. 4. RBFNN architecture.

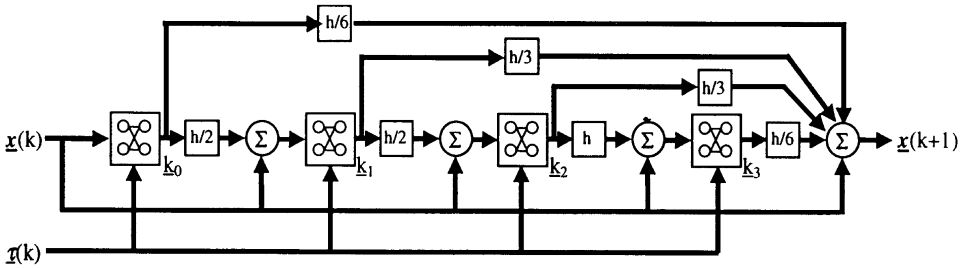

Fig. 5. Runge–Kutta neural network architecture.

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 identification procedure. The latter may be crucial in assigning the initial value of the RBFNN parameter vector to a vector that is close to the optimal one. This results in faster convergence in parameter space if the system dynamics is known. In this paper, the parameter vector is randomly initialized.

In the RBFNN used in this work, 12 hidden neurons are used. As is given by Eq. (16), 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 of appropriate

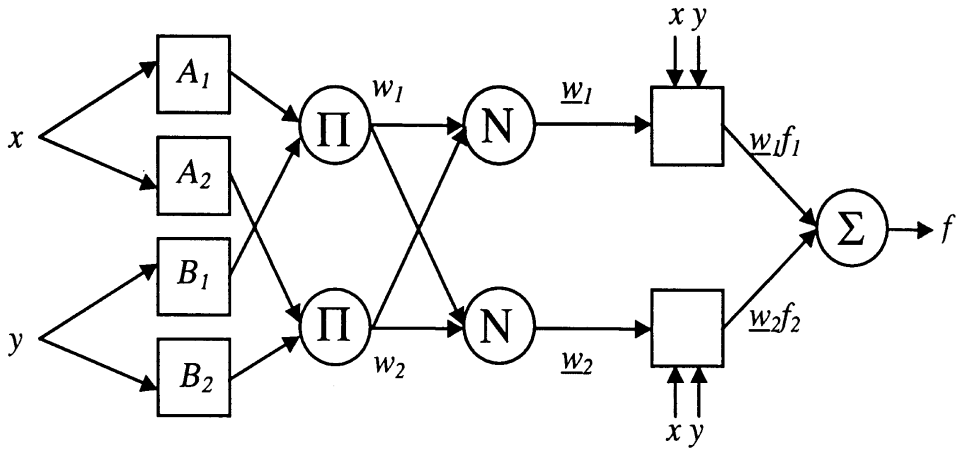

Fig. 6. Adaptive Neuro-Fuzzy Inference System (ANFIS) for two rules.

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 $\underline{y}$ and $\underline{o}$ respectively, the activation level of ith receptive unit (or firing strength of each rule) and the overall realization performed by the network can be given by Eqs. (16) and (17) respectively.

$$o_i = \prod_{j=1}^{\#\text{inputs}} \exp\left(\frac{(x_j - c_{ij})^2}{w_{ij}^2}\right) \tag{16}$$

$$\underline{y} = W_{\#\text{outputs}\, x\, \#\text{hidden neurons}}\, \underline{o} \tag{17}$$

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

$$\underline{\delta}_{\text{output}} = \underline{d} - \underline{y} \tag{18}$$

$$\underline{\delta}_{\text{hidden}} = W^T \underline{\delta}_{\text{output}} \tag{19}$$

$$W_{ij}(k+1) = W_{ij}(k) + \eta \delta_{\text{output}i} o_j \tag{20}$$

$$c_{ij}(k+1) = c_{ij}(k) + \eta \delta_{\text{hidden}\, i}\, o_j 2\left(\frac{x_j - c_{ij}(k)}{w_{ij}(k)}\right)^2 \tag{21}$$

$$w_{ij}(k+1) = w_{ij}(k) + \eta \delta_{\text{hidden}\, i}\, o_j 2\frac{(x_j - c_{ij}(k))^2}{w_{ij}(k)^3} \tag{22}$$

Simulation results for the RBFNN architecture are presented in Fig. 8. The reference

trajectories used are the same as those for the FNN structure. In fact the same command signals are used for all four approaches.

5. Runge–Kutta neural networks (RKNN) for system identification

The 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 observed that the network shows a satisfactory performance in estimating the system states in the long run. It should be emphasized that the neural network architecture realizes the changing rates of the system states instead of the $[\underline{x}(k), \underline{\tau}(k)] \Rightarrow [\underline{x}(k+1)]$ mapping. Therefore, the RKNN approach alleviates the difficulties introduced by the discretization methods. As known, the first order discretization brings large approximation errors. Wang and Lin [5] have simulated the approach with RBFNN architecture and trained the neural network for priorily observed data. This paper considers the approach with FNN with on-line tuning of the parameters. The RKNN architecture is illustrated in Fig. 5. In this figure, h denotes the Runge–Kutta integration stepsize.

Robot dynamics can be stated more compactly as given by (23). All of the neural networks appearing in Fig. 5, realize the vector function $\underline{f}$. Therefore, for fourth order Runge–Kutta approximation, the overall scheme is comprised of four times repeatedly connected neural network blocks and corresponding stage gains. The update mechanism is based on the error backpropagation. The derivation for FNN based identification scheme is given in Eqs. (24)–(33).

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{\tau}) \tag{23}$$

$$\underline{x}(i+1) = \underline{x}(i) + \frac{h}{6}(\underline{k}_0 + 2\underline{k}_1 - 2\underline{k}_2 + \underline{k}_3) \tag{24}$$

$$\underline{k}_0 = N(\underline{x}; \phi) = N(\underline{x}_0; \phi) \tag{25}$$

$$\underline{k}_1 = N(\underline{x} + \frac{1}{2}h\underline{k}_0; \phi) = N(\underline{x}_1; \phi) \tag{26}$$

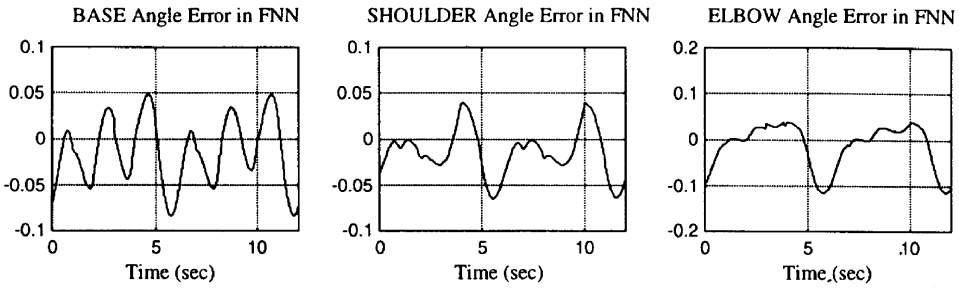

Fig. 7. Estimation errors with FNN structure.

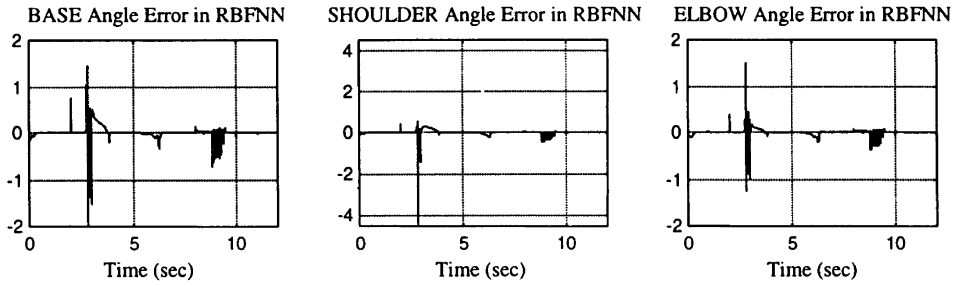

Fig. 8. Estimation errors with RBFNN structure.

$$\underline{k}_2 = N(\underline{x} + \frac{1}{2}h\underline{k}_1, \phi) = N(\underline{x}_2; \phi) \tag{27}$$

$$\underline{k}_3 = N(\underline{x} + h\underline{k}_2; \phi) = N(\underline{x}_3; \phi) \tag{28}$$

where, ϕ is a generic parameter of neural network. Fig. 5 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 FNN 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.

$$\frac{\partial \underline{k}_0}{\partial \phi} = \frac{\partial N(\underline{x}_0; \phi)}{\partial \phi} \tag{29}$$

$$\frac{\partial \underline{k}_1}{\partial \phi} = \frac{\partial \underline{k}_1}{\partial \underline{x}_1}\frac{\partial \underline{x}_1}{\partial \underline{k}_0}\frac{\partial \underline{k}_0}{\partial \phi} + \frac{\partial \underline{k}_1}{\partial \phi} \tag{30}$$

$$\frac{\partial \underline{k}_2}{\partial \phi} = \frac{\partial \underline{k}_2}{\partial \underline{x}_2}\frac{\partial \underline{x}_2}{\partial \underline{k}_1}\frac{\partial \underline{k}_1}{\partial \phi} + \frac{\partial \underline{k}_2}{\partial \phi} \tag{31}$$

$$\frac{\partial \underline{k}_3}{\partial \phi} = \frac{\partial \underline{k}_3}{\partial \underline{x}_3}\frac{\partial \underline{x}_3}{\partial \underline{k}_2}\frac{\partial \underline{k}_2}{\partial \phi} + \frac{\partial \underline{k}_3}{\partial \phi} \tag{32}$$

$$\Delta\phi(i) = \frac{\eta h}{6}(\underline{d}^T(i) - \underline{x}^T(i))\left(\frac{\partial \underline{k}_0}{\partial \phi} + 2\frac{\partial \underline{k}_1}{\partial \phi} + 2\frac{\partial \underline{k}_2}{\partial \phi} + \frac{\partial \underline{k}_3}{\partial \phi}\right) \tag{33}$$

In Eq. (33), η represents the learning rate and $\underline{d}(i)$ represents the measured state vector of the plant at time index i. Simulation results for RKNN methodology are presented in Fig. 9.

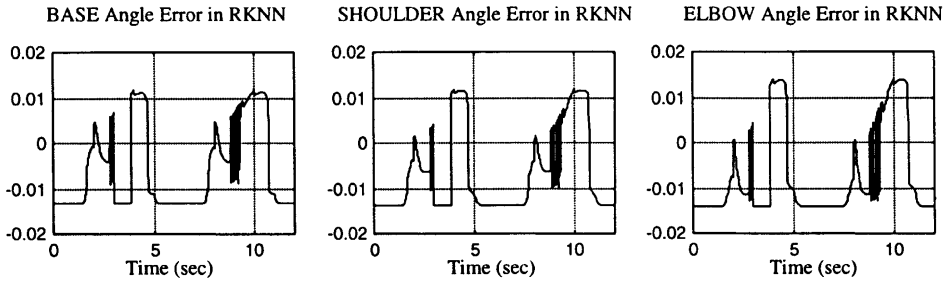

Fig. 9. Estimation errors with RKNN structure.

6. Adaptive neuro fuzzy inference systems (ANFIS) for system identification

Adaptive Neuro Fuzzy Inference Systems (ANFIS) are hybrid systems that combine the verbal power of fuzzy systems with the numeric power of neural systems. As is known from the theory of fuzzy systems, different fuzzification and defuzzification mechanisms with different rule base structures can propose various solutions to a given task. This paper considers the ANFIS structure with first order Sugeno model containing 25 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 firing strengths is normalized. The resulting vector is defuzzified by utilizing the first order Sugeno model. The procedure is explained briefly in Eqs. (34)–(38) for the ANFIS architecture illustrated in Fig. 6 for a simple rule base as follows:

$$\text{IF } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ THEN} f_1 = p_1 x + q_1 y + r_1$$

$$\text{IF } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ THEN} f_2 = p_2 x + q_2 y + r_2$$

$$w_1 = \mu_{A_1}(x)\mu_{B_1}(y) \tag{34}$$

$$w_2 = \mu_{A_2}(x)\mu_{B_2}(y) \tag{35}$$

$$\underline{w}_1 = \frac{w_1}{w_1 + w_2} \tag{36}$$

$$\underline{w}_2 = \frac{w_2}{w_1 + w_2} \tag{37}$$

$$f = \underline{w}_1(p_1 x + q_1 y + r_1) + \underline{w}_2(p_2 x + q_2 y + r_2) \tag{38}$$

The ANFIS output is clearly a linear function of the adjustable defuzzifier parameters. For the adjustment of $[\underline{p}\ \ \underline{q}\ \ \underline{r}]^T$ vector, gradient descent method is applied. Depending on the system in hand, the parameters of the membership functions can be initialized so that the convergence speed is increased.

In robotic manipulator identification problem, the fuzzifier input vector dimension is equal to the number of plant states plus the number of plant inputs, the rule base

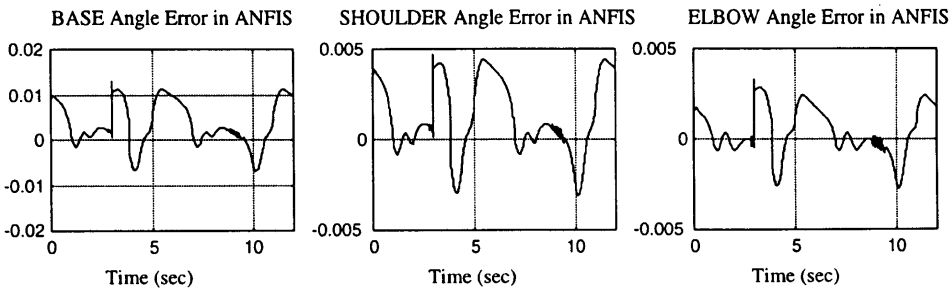

Fig. 10. Estimation errors with ANFIS structure.

contains 25 rules and the defuzzifier output vector dimension is equal to the number of plant states. The simulation results are presented in Fig. 10.

7. Comparison of the four approaches

In order to study the performances of the four approaches described above, a series of simulation studies has been performed and some performance measures are tabulated in Tables 2, 3, and 4. In Table 2, each column represents the error bound for a variable, e.g., B_p stands for the position error bound for base position, S_v is the velocity error bound for shoulder velocity and so on. The use of RKNN and ANFIS architectures result in a good estimation performance, as is clearly seen by the respec-

Table 2
Comparison of the tracking performances for the four approaches

	B_p	S_p	E_p	B_v	S_v	E_v	Σ
FNN	0.0844	0.0652	0.1177	0.2386	0.1720	0.0812	0.7591
RBFNN	1.9845	4.3588	1.4989	2.6527	1.2683	0.6938	12.4570
RKNN	0.0131	0.0137	0.0141	0.0139	0.0121	0.0134	0.0804
ANFIS	0.0130	0.0046	0.0033	0.0764	0.2577	0.0538	0.4088

Table 3
Comparison of pre-training times

	(hr)
FNN	0.87
RBFNN	148.60
RKNN	—
ANFIS	0.91

Table 4
Comparison of the various performance measures (H: high, L: low and M: medium)

	FNN	RBFNN	RKNN	ANFIS
Estimation performance	L	L	H	H
Pre-training time	L	H	–	L
Operational simplicity	H	H	M	M
Expert knowledge incorporation	L	M	L	H

tive position and velocity error bounds for all three links. Table 3 compares another distinguishing property of the approaches namely the need for pre-training. In this table, no figure is given for RKNN method because it operates on-line. ANFIS and FNN architectures require approximately the same amount of pre-training time. RBFNN approach requires by far the longest pre-training time. The overall assessment of the approaches is given in Table 4, where H denotes high, M denotes medium and L denotes low. The approaches are evaluated for four different comparison measures. The first row clearly recommends the use of RKNN or ANFIS. The second row accounts for the need for pre-training time. In this sense, RKNN is the best approach. On the other hand, from the viewpoint of operational simplicity, FNN and RBFNN are the simplest approaches. The last row considers the ability to incorporate expert knowledge. As is known from the theory of fuzzy systems, the philosophy of ANFIS architecture best fulfills this task because the design starts with the rules in the form of IF-THEN statements which are the best means for expressing the expert-machine interaction.

8. Conclusions

This study analyzes the performance characteristics of a number of methodologies that can be used for system identification purposes. In their assessment, the estimation performance, together with the training times are considered as the primary comparison measures. It is seen that RKNN and ANFIS perform best in estimation. On the other hand, RBFNN and FNN are the simplest approaches in the sense of computational complexity. Another important criterion is the dependency on a priori knowledge. Except the RKNN approach, all methods need a pre-training phase, which is seen to be by far the longest in the case of RBFNN.

The work, reported in this paper, indicates that ANFIS structure is a good candidate for identification purposes. Additionally, the elegant performance of the RKNN approach with on-line operation and with ordinary feedforward neural network stages instead of RBFNN based architecture is demonstrated.

The work is in progress in the direction of improving the estimation performance through the use of ANFIS and RKNN approaches in combination.

Acknowledgements

The authors acknowledge the support provided through the Bogazici University Research Fund for the project no. 97A0202.

References

[1] Hornik K. Feedforward networks are universal approximators. Neural Networks 1989;2:359–366.
[2] Funahashi K. On the approximate realization of continuous mappings by neural networks. Neural Networks 1989;2:183–192.
[3] Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1990;1(1):4–27.
[4] Jang J-SR, Sun C-T. Neuro-fuzzy modelling and control. Proceedings of IEEE 1995;83:378–406.
[5] Wang Y-J, Lin C-T. Runge–Kutta Neural Network for Identification of Dynamical Systems in High Accuracy. IEEE Transactions on Neural Networks 1998;9(2):294–307.
[6] Jang J-SR, Sun C-T, Mizutani E. In: Neuro-Fuzzy and Soft Computing. PTR Prentice Hall, 1997.
[7] Erbatur K, Vinter RB, Kaynak O. Feedback Linearization Control for a 3-DOF Flexible Joint Elbow Manipulator. Proceedings of the 1994 IEEE International Conference on Robotics and Automation, May 8–13, San Diego, U.S.A 1994;4:2979–2984.
[8] Stadler W. In: Analytical Robotics and Mechatronics. McGraw–Hill, 1995. p. 476.