

Identifying Three Linear Systems Using Only Single Neural Model

Şeref SAĞIROĞLU
Gazi University, Engineering Faculty,
Computer Engineering Department, 06570, Maltepe, Ankara, Turkey

ABSTRACT

This paper introduces a new approach based on artificial neural networks (ANNs) to identify a number of linear dynamic systems using single neural model. The structure of single neural model is capable of dealing with up-to three systems. Single neural model is trained by the back propagation with momentum learning algorithm. Total nine systems from first to third orders have been used to validate the approach presented in this work. The results have shown that single neural model is capable to identify not only one system but also two and three different systems very successfully. The new identification approach presented in this work provides simplicity, accuracy and compactness. This might bring new aspects to system identification, modelling and control applications.

Keywords: Neural network; Linear system; Multi-system identification.

Üç Doğrusal Sistemin Tek Bir Nöro Model ile Kimliklendirilmesi

ÖZET

Bu makalede, tek bir yapay sinir ağ (YSA) modeli kullanılarak birden çok lineer dinamik sistemlerin kimliklendirilmesine yönelik yeni bir yaklaşım sunulmuştur. Momentumlu geri yayılım öğrenme algoritması ile eğitilen tek YSA modeli, üç farklı doğrusal sistemin kimliklendirilmesi yeteneğine sahiptir. Farklı sistem derecesine sahip 9 doğrusal sistem, sunulan yaklaşımın başarısını doğrulamak için kullanılmaktadır. Elde edilen sonuçlar, sunulan yaklaşım ile tek bir YSA modelinin sadece tek bir sistemin kimliklendirlmesi değil aynı zamanda iki ve üç farklı doğrusal sistemin, yüksek başarımla sistemleri kimliklendirme yeteneğine sahip olduğunu göstermiştir. Bu yaklaşımın, kimliklendirmede basitlik, doğruluk ve kompakt bir yapı sunmakta olup sistem kimliklendirme, modelleme ve kontrol uygulamalarına yeni bir bakış açısı sunacağı değerlendirilmektedir.

Anahtar Kelimeler: Yapay sinir ağı, Doğrusal sistem; Çoklu-sistem kimliklendirme.

1 INTRODUCTION (GİRİŞ)

Using artificial neural networks (ANNs) in dynamic system identification is a well-known and still popular topic because of their advantages including their capacity for learning the behavior of a system without much a *priori* knowledge about it, their high degree of robustness, simplicity and their ability to generalize systems [1]-[12].

Feed-forward neural networks have been successfully applied to system identification [1]-[8]. These networks are versatile and relatively simple to apply. They achieve identification task with sufficient input neurons accepting outputs from preceding time steps immediately. Core studies were introduced by Narendra and Parthasarathy for efficient dynamic system identification and control with neural networks [1,3]. They presented four models to represent parallel and series-parallel models for static and dynamic backpropagation algorithms. These studies were based on identifying systems with single neural model and mostly referred in most system identification and control works. Sagiroglu introduced a neural model capable of dealing with systems of up to a given maximum order [8]. In his study, one neural model performed identification task up-to fifth order linear system without knowing the both orders and parameters of the systems. Better results were achieved in the studies [8]-[12] but the complexity in design and application was increased. However, a number of approaches have been introduced to improve single model efficiency [9]-[12]. In [9]-[12], systems were identified and controlled with the use of multiple models with the help of using artificial neural networks. These models have been applied for solving chaotic time series, adaptive systems and complex control systems. In [12], two different neural network models were combined to identify one linear dynamic system.

Sagiroglu and Kalinli [13] have been recently presented an approach to simplify system identification. In that work, a single neural identifier was capable of identifying two linear systems. This paper is an

* *Sorumlu Yazar (Corresponding Author)*
e-posta: ss@gazi.edu.tr
Digital Object Identifier (DOI) : 10.2339/2012.15.4, 191-198

extension of [13] and introduces a simpler neural identifier which identifies not only two but also three linear systems having same or different orders with single neural identifier. Kalinli and Sagiroglu [14,15] have also successfully tested this approach for linear and nonlinear system identifications using recurrent neural networks.

The paper has four main sections. Section 2 briefly introduces ANNs. Section 3 discusses the identification of linear dynamic system and the proposed approach for multi-system identification by neural networks. Section 4 describes the simulations carried out to test the proposed identification scheme and gives simulation results of linear systems ranging from one to three having different orders to be identified with single feed-forward neural model. The work presented in this article is finally concluded in Section 5.

2. ARTIFICIAL NEURAL NETWORKS (YAPAY SİNİR AĞLARI)

Artificial neural networks (ANN) generally have three layers: an input layer, an output layer and an intermediate or hidden layer. Neurons in the input layer only act as buffers for distributing the input signals x_i to neurons in the hidden layer. Each neuron j in the hidden layer sums up its input signals x_i after weighting them with the strengths of the respective connections w_{ji} from the input layer and computes its output y_j as a function f of the sum as

$$y_j = f\left(\sum w_{ji} x_i\right) \quad (1)$$

where f can be a simple threshold function or a linear function.

There have been many available learning algorithms in the literature [16,17]. Backpropagation with momentum (BPM) learning algorithm is the most commonly adopted ANN training algorithm [16]. The BPM algorithm employs a gradient descent technique to adopt the ANN weights to minimize the mean squared difference between the ANN output and the desired output. The change in weight $\Delta wji(k)$ between neurons i and j is as follows,

$$\Delta w_{ji}(k) = \eta \delta_j x_i + \alpha \Delta w_{ji}(k-1) \quad (2)$$

where η is a parameter called the learning coefficient, α is the momentum coefficient, and δ_j is a factor depending on whether neuron j is an output neuron or a hidden neuron.

For output neurons,

$$\delta_j = \frac{\partial f}{\partial net_j}(y_j - y_{net-j}) \quad (3)$$

where

$$net_j = \sum_i x_i w_{ji}$$

y_j and y_{net-j} are the target and the neural outputs for neuron j, respectively. For hidden neurons,

$$\delta_j = \frac{\partial f}{\partial net_j} \sum_q w_{qj} \delta_q \quad (4)$$

As there are no target outputs for hidden neurons in Eq.(4), the difference between the target and the actual neural output of a hidden neuron j is replaced by the weighted sum of the δ_q terms already obtained for neurons q connected to the output of j. Thus, iteratively, beginning with the output layer, the δ term is computed for neurons in all layers and weight updates determined for all connections according to Eq.(2).

Training an ANN by BPM to compute y_{net} involves presenting it sequentially with different training sets. Differences between the target output and the actual output of the ANN are back-propagated through the network to adapt its weights using Eqs.(2)-(4). The adaptation is carried out after the presentation of each set. Each training epoch is completed after all patterns in the training set have been applied to the networks.

3. LINEAR DYNAMIC SYSTEM IDENTIFICATION (DOĞRUSAL DİNAMİK SİSTEM KİMLİKLENDİRME)

Linear system identification is widely used in signal processing because of its well established theoretical foundation and inherent simplicity. This identification offers a generalized framework for implementing a linear system for prediction and control applications. Most of the concepts and ideas for nonlinearity can be also simplified in the form of linearity. However, many control systems in industry are still being solved using linear techniques. An understanding of the details involved in the linear case allows the direct extension to solve nonlinear problems.

System identification techniques are applied in many fields to predict the response of unknown systems using a given set of input-output data. ANN is a common method to identify dynamic systems. Although it is also possible to make an ANN learn the state-space representation of the system if the state variables are available [4], in the work, the inputs-outputs appeared was adopted as it is simpler to implement than the state-space approach.

As known very well, a linear time-invariant system with unknown parameters is written as:

$$y(k+1) = \sum_{i=0}^{n-1} A_i y(k-i) + \sum_{l=0}^{m-1} D_j u(k-l) \quad (5)$$

where A_i and D_l are the coefficients with unknown parameters. Identifying a linear system means obtaining the coefficients given in Eq (5).

The input-output characteristics of the system will generally change rapidly or even discontinuously when the environment changes. In linear systems, such adaptation is possible, but the slowness of adaptation may result in a large transient error. Simpler model is always desired for both to identify different environments and to control them rapidly. If single identification model is used, it will have to adapt itself to the new environment before appropriate control action can be taken.

4. PROPOSED APPROACH FOR SYSTEM IDENTIFICATION (SİSTEM KİMLİKLENDİRME İÇİN ÖNERİLEN SİSTEM)

Figure 1 shows the new approach based on ANN in diagrammatic form. In Figure 1, $u(k)$ and $y_i(k)$ represent the input to the system and the ANN and the outputs of the systems, respectively. $y_{net\_i}(k)$ also represents the outputs of the ANN. As can be seen from the figure that, three different linear systems are supposed to be identified by only one neural identifier.

The input-output relationship of linear dynamic systems up-to three or more is expected to be identified within that single identifier. This approach is named as “multi-system identification”. It should be emphasized that the three different systems having different orders or/and the same orders are to be tested in this work.

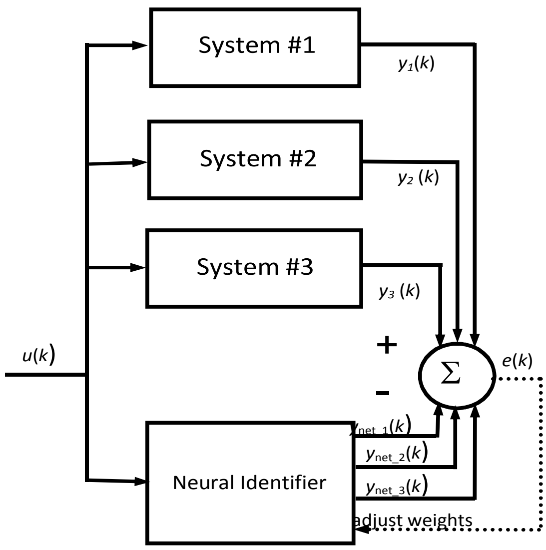


Figure 1. Single neural identifier for identifying three different systems

The approach is based on an adaptation with the tapped-delay line technique to increase its versatility and simplicity in the system identification. The only knowledge required for the systems is the numbers of the systems to be identified within one model.

Identifying three linear systems within single neural identifier means achieving the coefficients of three systems using single identifier. As opposed to the earlier studies in the literature [1-13], only one neural identifier is capable of identifying three systems having different behaviors.

Discrete form of this new approach for three systems is represented as:

$$\boldsymbol{Y}(.) = \boldsymbol{A1}y_1(.) + \boldsymbol{B}y_2(.) + \boldsymbol{C}y_3(.) + \boldsymbol{D}u(.) \quad (6)$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ represent the coefficients belonging to each of 3 different linear systems. $\boldsymbol{Y}(.)$ represents the outputs of 3 systems to be identified with single neural model. $u(.)$ is the input to 3 systems. $y_1(.)$, $y_2(.)$ and $y_3(.)$ also are the outputs of each system to be identified. As shown in Eq (6), the coefficients $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ belonging to three systems must be accurately achieved by the single identifier. The discrete representation of three linear systems becomes more complicated in comparison with the single system given in Eq (5).

A single neural identifier is supposed to achieve A_i, B_j, C_t and D_l coefficients given in Eq (7).

$$Y(.) = \sum_{i=0}^{m-1} A_i y_1(k-i) + \sum_{j=0}^{n-1} B_j y_2(k-j) + \sum_{t=0}^{o-1} C_t y_3(k-t) + \sum_{l=0}^{z-1} D_l u(k-l) \quad (7)$$

In order to test this approach, 22 different neural identifiers having different combinations of nine linear systems were trained throughout this work. Different linear systems having different behaviors were selected for proper and robust tests.

As mentioned earlier, the neural identifiers were trained by BPM learning algorithm. This algorithm adjusts the weights according to the differences between the system outputs and the neural output/s. This adjustment starts with the neurons in the output layer and proceeded backwards towards the input layer, which explains the name of the algorithm.

In order to get good performance, neural identifiers were sufficiently trained according to the criteria such as a preset iteration number or an error level set. The root mean square (*rms*) error was used to evaluate the performance of this study. The error function for measuring the performance is defined as

$$rms\ error = \left[\frac{1}{s}\sum_{j=1}^{s}\frac{1}{p}\sum_{i=1}^{p}\left[y_i(k+1) - y_{net-i}(k+1)\right]^2\right]^{1/2} \quad (8)$$

where s is the number test sequence, p is the number of systems to be identified, $y_i(k+1)$ and $y_{net\_i}(k+1)$ are the outputs of i-th system and the network at time $k+1$, respectively.

In this work, s was fixed to 100, p was set to 1, 2 or 3 according to the system/s to be tested for proper identification with single neural identifier.

5. SIMULATIONS AND RESULTS (BENZETİMLER VE BULGULAR)

A number of computer simulations were carried out to test the proposed identification scheme based on ANNs. Total nine systems (plants) and twenty two neural identifiers were used to test the new approach presented in this work.

The systems tested are illustrated in Table 1. The systems given in Table 1 by #1-#3, #4-#6 and #7-#9 represent the first, the second, and the third order systems, respectively

values between [-1.0,+1.0]. A step input $u(k)=1.0$, $k=0,1,...,99$, was used to test for all identifiers.

Table 1. Linear systems used in proposed identifications

System Orders	System no	Systems or plants to be identified
First	#1	$y(k+1) = 0.9*y(k)+0.1*u(k)$
	#2	$y(k+1)=0.95*y(k)+0.5*u(k)$
	#3	$y(k+1)=0.75*y(k)+0.25*u(k)$
Second	#4	$y(k+1)=1.752821*y(k)–0.818731*y(k-1)+0.011698*u(k) +0.010942*u(k-1)$
	#5	$y(k+1)=1.1953*y(k)–0.4317*y(k-1)+0.1348*u(k)+0.1017*u(k-1)$
	#6	$y(k+1)=1.7826*y(k)–0.8187*y(k-1)+0.01867*u(k)+0.01746*u(k-1)$
Third	#7	$y(k+1)=2.627771*y(k)–2.333261*y(k-1)+0.697676*y(k-2) + 0.017203*u(k)-0.030862*u(k-1)+0.014086*u(k-2)$
	#8	$y(k+1)=2.038*y(k)–1.366*y(k-1)+0.301*y(k-2) + 0.0059*u(k) -0.018*u(k-1)+0.0033*u(k-2)$
	#9	$y(k+1)=1.2*y(k)–0.5*y(k-1)+0.1*y(k-2) - 0.3*u(k)+0.4*u(k-1)-0.5*u(k-2)$

According to the systems to be identified, all the neural identifiers had 2, 3, 4, 6, 8, 9 or 12 neurons in the input layer and 1, 2 or 3 neurons in the output layer. Neural structures tested in this work also had only one hidden layer with 5 or 10 neurons. The number of neurons in the hidden layer was a test parameter.

Linear activation functions were used in all neurons so the systems to be identified were all linear.

The other test parameters were the learning coefficient, η (which determines the amount by which a connection weight is changed according to the error gradient information), the momentum coefficient, α (which governs the amount of weight change in one iteration due to the change in the immediately previous iteration), and the number of iterations in training (a training iteration is completed after all the weights have been adopted once in this work, this means after each input pattern has been presented to the neural identifiers).

The input data set was obtained by applying uniformly random inputs to systems (after seeking by the appropriate seeking factor) and observing the corresponding outputs from the systems.

At the beginning of each training process, the weights of the neural connections were initialized to small random values in the range [-0.1, 0.1]. The random input sequence $u(k)$ to the neural identifiers had

The proposed approach presented in this work was tested with different neural identifiers having various configurations. Each neural identifier was trained by BPM algorithm to identify one, two and three systems together with the same or different orders. For example, the inputs to one of 22 neural models were $u(k)$, $u(k-1)$, $u(k-2)$, $y_1(k)$, $y_1(k-1)$, $y_1(k-2)$, $y_2(k)$, $y_2(k-1)$, $y_2(k-2)$, $y_3(k)$, $y_3(k-1)$ and $y_3(k-2)$ for the three third-order systems (#7, #8 and #9). The corresponding outputs of the single neural model are $y_1(k+1)$, $y_2(k+1)$ and $y_3(k+1)$.

After proper or sufficient training of each neural identifier, the single identifiers were first tested for one, two and three first-order systems given in Table 1. The tests were then extended to three systems. Finally, each single neural identifier was examined for three systems having the same or different orders.

The performance of each neural identifier was then tested with the step function having 100 samples.

Table 2 shows the results achieved from individual and multi-system identifications obtained from the proposed approach.

The structure of neural network identifiers, the parameters of the learning algorithm and the number of iterations for each neural identifier are also listed in Table 3 for the nine different linear systems.

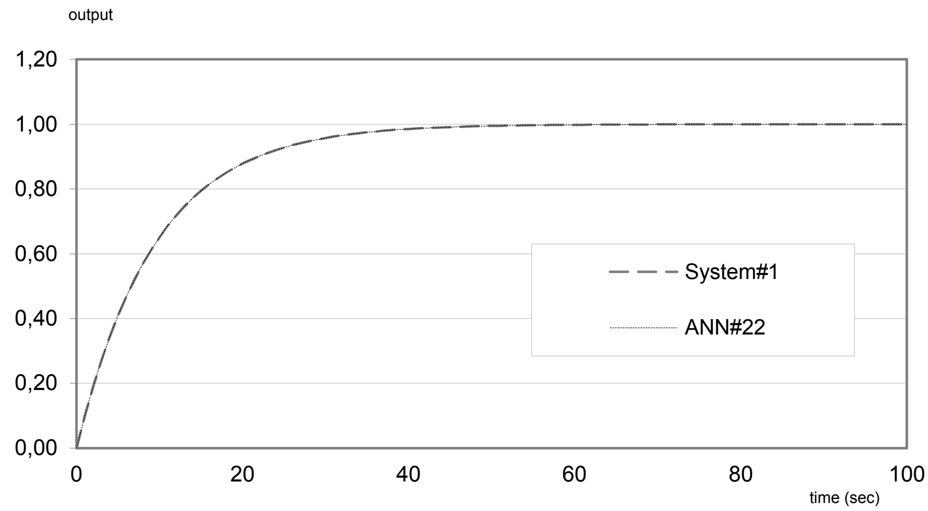
As can bee seen from Tables 1, 2 and 3, the systems can be identified with a single neural models with high accuracy and low rms errors even if the systems having different system orders.

Figure 2 demonstrates the responses of the three systems identified having the orders from first to the third by only one identifier. As can be seen from Table 2 and Figure 2, only one neural identifier is very

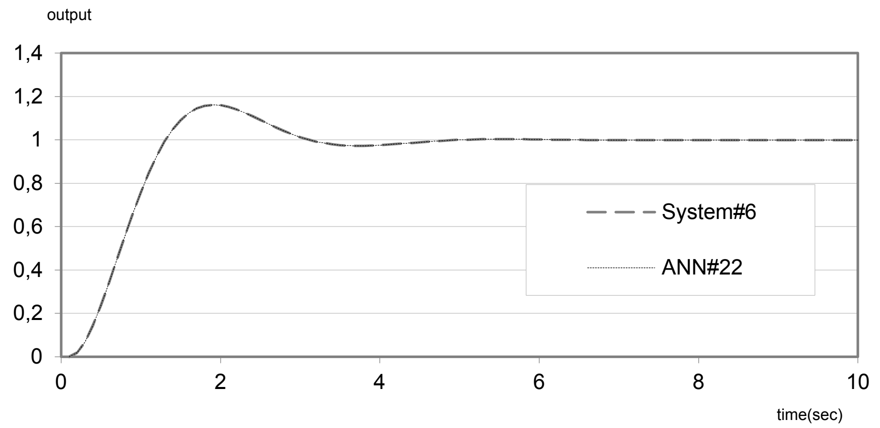
Table 2. Results obtained from single neural identifiers for multi-system identification

System order	Neural Identifiers		Neural Identifier	System no	rms error for single system in test	rms error for multi-system in test
	Inputs	Output/s				
First	2	1	ANN#1	#1	0.0006	-
	2	1	ANN#2	#2	0.0006	-
	2	1	ANN#3	#3	0.0006	-
	3	2	ANN#4	#1	0.0002	0.0003
				#2	0.0004	
	3	2	ANN#5	#2	0.0000	0.0000
				#3	0.0000	
	3	2	ANN#6	#1	0.0000	0.0000
				#3	0.0000	
	4	3	ANN#7	#1	0.0008	0.0006
				#2	0.0006	
				#3	0.0003	
Second	4	1	ANN#8	#4	0.0000	-
	4	1	ANN#9	#5	0.0000	-
	4	1	ANN#10	#6	0.0000	-
	6	2	ANN#11	#4	0.0000	0.0000
				#5	0.0000	
	6	2	ANN#12	#5	0.0000	0.0000
				#6	0.0000	
	6	2	ANN#13	#4	0.0000	0.0000
				#6	0.0000	
	8	3	ANN#14	#4	0.0021	0.0023
				#5	0.0015	
				#6	0.0032	
Third	6	1	ANN#15	#7	0.0000	-
	6	1	ANN#16	#8	0.0000	-
	6	1	ANN#17	#9	0.0000	-
	9	2	ANN#18	#7	0.0051	0.0044
				#8	0.0036	
	9	2	ANN#19	#7	0.0059	0.0030
				#9	0.0001	
	9	2	ANN#20	#8	0.0000	0.0000
				#9	0.0000	
	12	3	ANN#21	#7	0.0103	0.0070
				#8	0.0036	
				#9	0.0019	
First, Second, Third	9	3	ANN#22	#1	0.0008	0.0020
				#6	0.0004	
				#8	0.0049	

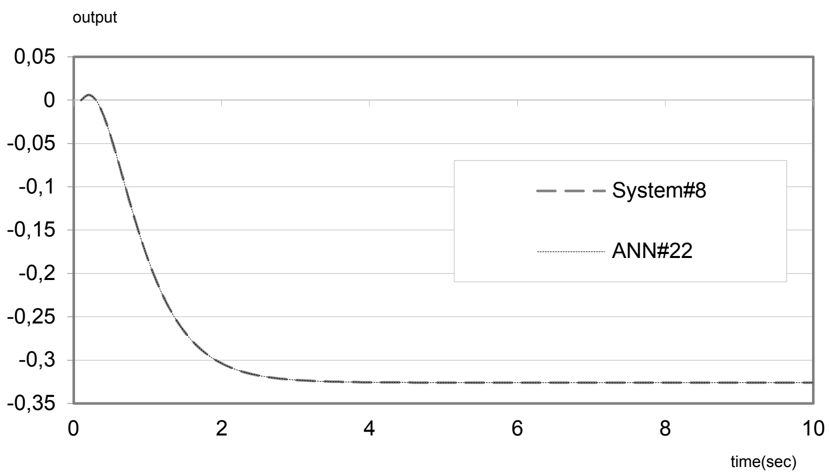
capable in successfully identifying not only one system but also two and three different systems having the same and different orders.



(a) first-order system (system#1)



(b) second-order system (system#6)



(c) third-order system (system#8)

Figure 2. Step responses of single neural identifier

When the parameters of neural identifiers are considered in Table 3, the identifiers were mostly insensitive to the learning and momentum coefficients and the number of hidden units. The learning coefficients, 0.2 and 0.3, and the momentum coefficient, 0.4, are the most suitable values for a proper multi-system identification. Increasing the number of systems to be identified with one neural identifier only requires large number of iterations.

accuracy and compactness to system identification approach.

Table 3. Default parameters for single neural identifiers

System order	System no	Neural Identifier	No of systems to be identified within single neural identifier	Neurons in hidden unit	η	α	Iteration (x1,000)
First	#1	ANN#1	1	5	0.3	0.4	10,000
	#2	ANN#2	1	5	0.3	0.4	10
	#3	ANN#3	1	5	0.3	0.4	10
	#1, #2	ANN#4	2	5	0.3	0.4	500
	#2, #3	ANN#5	2	5	0.3	0.4	500
	#1, #3	ANN#6	2	5	0.3	0.4	500
	#1, #2, #3	ANN#7	3	5	0.3	0.4	1,000
Second	#4	ANN#8	1	5	0.2	0.4	100
	#5	ANN#9	1	5	0.2	0.4	100
	#6	ANN#10	1	5	0.2	0.4	100
	#4, #5	ANN#11	2	10	0.2	0.4	1,000
	#5, #6	ANN#12	2	10	0.2	0.4	1,000
	#4, #6	ANN#13	2	10	0.2	0.4	1,000
	#4, #5, #6	ANN#14	3	10	0.2	0.4	2,000
Third	#7	ANN#15	1	10	0.2	0.4	100
	#8	ANN#16	1	10	0.2	0.4	100
	#9	ANN#17	1	10	0.2	0.4	100
	#7, #8	ANN#18	2	10	0.2	0.4	1,000
	#7, #9	ANN#19	2	10	0.2	0.4	1,000
	#8, #9	ANN#20	2	10	0.2	0.4	1,000
	#7, #8, #9	ANN#21	3	10	0.2	0.4	2,000
First, Second, Third	#1, #6, #8	ANN#22	3	10	0.2	0.4	1,000

A computer program was developed in C++ for the simulations. The program can be obtained from the author by an e-mail.

6. CONCLUSIONS (SONUÇLAR)

A new neural approach based on ANNs for multi-system identification has been successfully presented for linear systems. The approach involves using only one ANN model to learn the input-output behaviors of multi-system.

Total nine systems from the first to the third orders were used to test the approach presented in this work. The results have shown that single neural identifier performs multi-system identification with high accuracy. The identification processes for multi-system were also faster and not much sensitive to the ANN parameters except the number of iterations. The new approach presented in this work provides simplicity, If the proposed approach had been extended to other modelling, identification and control applications, it would provide simplicity and accuracy to system designers dealing with compact modelling and control applications. The author will focus on extending this approach to multi-nonlinear system identification and applying this approach to control applications for further achievements.

7. REFERENCES (KAYNAKLAR)

1) K. S. Narendra, K. Parthasarathy, Neural Networks in Dynamical Systems, SPIE, Intelligent Control and Adaptive Systems, 1196 (1989) 230-241.

2) S. Chen, S.A. Billings, P.M. Grant, Nonlinear System Identification Using Neural Networks, Int. J. Control, 51 (1990), 1191-1214.

3) K.S. Narendra, K. Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans on Neural Networks, 1 (1990), 4-27.

4) D.T. Pham, X. Liu, Neural Networks for Identification, Prediction and Control (4th edition), (Springer Verlag, 1999).

5) D.T. Pham, X. Liu, Neural Network for Discrete Dynamic System Identification, J. Syst. Eng., 1 (1991), 51-60.

6) A.B. Bulsari, H. Saxen, A Feedforward Artificial Neural Network for System Identification of a Chemical Process, J. Syst. Eng., 1 (1991), 23-25.

7) D.T. Pham, S. Sagiroglu, Three Methods of Training Multi-Layer Perceptrons to Model a Robot Sensor, Robotica, 13 (1995), 531-538.

8) S. Sagiroglu, Linear Systems Identification Using Multi Layer Perceptron, Proceedings of ISCIS XI, 2, Antalya, Turkey, 517-526, November 1996.

9) K.S. Narendra, J. Balakrishnan, M.K. Ciliz, Adaptation and Learning Using Multiple Models, Switching, and Tuning, IEEE Control Systems Magazine, 15 (1995), 37-51.

10) K. S. Narendra, J. Balakrishnan, Adaptive Control Using Multiple Models, IEEE Trans on Automatic Control, 42 (1997), 171-187.

11) R. Gundala, K.A. Hoo, M.J. Piovoso, Multiple Model Adaptive Control Design for a Multiple-Input Multiple-Output Chemical Reactor, Ind. & Eng. Chem. Res., 39 (2000), 1554-1564.

12) J.B. Galván, M.J. Pérez-Ilzarbe, Two Neural Networks for Solving the Linear System Identification Problem, Proceedings of the IEEE Int. Conf. on Neural Networks, 3226-3231, 1994.

13) S. Sagiroglu, A Kalinli, A New Approach Based On Neural Networks For Multi-System Identification IJCI Proceedings of XII. TAINN'2003, 1 (2003), 61-64.

14) A. Kalinli, S. Sagiroglu, A New Approach Based On Recurrent Neural Networks For System Identification, Lecture Notes on Computer Science (LNCS) 2869 Proceedings, (Springer-Verlag Berlin Heidelberg) 2003, 568-575, 18th Int. Symp. ISCIS 2003.

15) A. Kalinli, S. Sagiroglu, Elman Network with Embedded Memory for System Identification, Journal of Information Science and Engineering, Vol.22, 1555-1568, 2006.

16) D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing 1, (MIT Press, Cambridge, MA, 1986).

17) S. Haykin, Neural Networks A Comprehensive Foundation, (Macmillan College Publishing Comp., New York, 1994).