EJT

INESEG

Research Article

IMITATION OF FUZZY LOGIC CONTROLLER BASED ARTIFICIAL NEURAL NETWORK, AND APPLICATION OF INVERTED PENDULUM SYSTEM CONTROL

Murat SAM[1*], M. Serhat CAN[2]

Fuzzy Logic Controllers (FLCs) are effective solutions for nonlinear and parameter variability systems, but it contains multiple mathematical operations causing the controller to react slowly. This study aims to obtain a controller that can imitate the effective control performance of the FLC, which is easy to design both in software and hardware, and has a short response time. Artificial neural networks (ANNs) provide effective solutions in system modeling. Modeling of FLC has been realized by using of ANN's learning and parallel processing capability. The design process of the FLC and the training processes of the ANN were studied in Matlab SIMULINK environment. In the study, FLC was modelled at high similarity ratio with small ANN structure. ANN results were obtained very faster than the FLC control performance. The control performances of two controllers were observed to be very close to each other. As a result, ANN model has smaller structure than FLC, which makes it possible to implement the controller easily in terms of hardware and software.

Key words*: Fuzzy logic controller, artificial neural network, fuzzy logic controller modelling.*

1. Introduction

Today, with the development of technology, some demands in our daily activities such as speed, safety and comfort are increasing. For example, while people aim to only travel between two cities in the early ages, they are now looking for solutions to how they can perform faster, safer and more comfortably travelling. To ensure a faster, safer and more comfortable travelling, advanced and often complex control systems are required.

There is a relationship between an input signal applied to a controlled system and the system's response to this input signal. This relationship, sometimes linear or nonlinear, is known as the transfer function of the system and can be formulated with a mathematical expression.

$$y = m * x + n \tag{1}$$

The Eq. (1) is an example for linear systems. Here m and n represent constant coefficients, *x* represents the input, and *y* represents the output response of the system to the input signal. It is also

[1] Department of Electrical and Electronics Engineering, Gaziosmanpasa University, Tokat, Turkey, (murat.sam0218@gop.edu.tr)

[2] Department of Electrical and Electronics Engineering, Gaziosmanpasa University, Tokat,Turkey, (mehmetserhat.can@gop.edu.tr) https://orcid.org/0000-0003-2356-9921

Received: 25 November 2019; Accepted: 24 December 2019

Doi: https://doi.org/10.36222/ejt.650617

easy to control because the mathematical models of linear control systems are simple and easy to obtain. The condition that a system to be controlled is linear is that changes in the input signal always affect the output at the same rate. The change in the x input shown in Eq. (1) will affect the y output response of the system at the same rate.

According to Newton, the universe functions systematically in a linear way like clockwork. Small changes cause small effects, and large changes cause large effects. Linearity, which is one of Newton's main concepts, contradicts the theory of complexity. Because this theory rejects linearity and predictability, it argues that predictability of complex systems is very difficult [1]. Complex systems are generally nonlinear systems and are very difficult to control. They don't have a specific equation, and some complex systems can have great effects on the output even at the slightest changes in the input [2].

Easy to design controllers such as Proportional-Integral-Derivative (PID) are sufficient for the Linear Time-Invariant (LTI) systems, while the control of non-LTI systems, whose parameters may change over time, requires more advanced control methods and algorithms. FLCs and ANNs are used for non-LTI systems. FLCs are designed using rules specific to the system-specific expert experience. ANN has inference and learning skills between system inputs and outputs. These two methods provide effective control solutions in non-LTI systems.

Fuzzy logic, which forms the basis of FLC, was proposed by Zadeh in 1965 as an extension of classical set theory. Later, the subject was dusted, thanks to the articles published by Zadeh in 1975, the first implementation studies related to FLC were carried out by Mamdani and Assilian. In 1987, Hitachi's FLC system was used and its commercial advantages were proved. Fuzzy logic uses fuzzy sets. Fuzzy sets are slightly different sets inspired by crisp sets. The difference is due to membership values. Unlike crisp sets, instead of the membership values 1 and 0 in crisp sets, it uses this range completely, which uses infinite values in real numbers. Actually, fuzzy logic is inspired from the sentences we use randomly in daily life. FLCs fulfills the task of control using an algorithm encoded with terms that do not express certainty when making decisions. In this way, the machine is able to think more humane and tries to provide a more realistic control [3].

The first theories on ANN, another of these methods, were put forward by Mc Culloch and Pitts in 1943. Later, in 1949, Hebb conducted a study to explain how the neurons of the human brain were self-learning. In 1958, Rosenblatt proposed the Perceptron ANN model based on the model proposed by McCulloch and Pitts. In the 1960s, Widrow and Hoff successfully carried out both theoretical and practical applications of ANN. Later in 1970s, even though the pace of development decreased due to practical difficulties in solving many real-world problems, the single-layer constraint was eliminated by Paul Werbos in 1974 and many restricted processes were calculated [4-6]. Although the ANN method, which also has problems, has been interrupted occasionally, in the light of the Hopfield article in 1982, many limitations have been eliminated. Even though new problems arise from these dates, new solutions have been produced. ANN is still popular today.

With the ANN structure and generalization feature, it can generate appropriate responses for untold input data and thus solve complex problems. Since ANN consists of a combination of many cells, it is important in solving complex nonlinear problems [7].

ANN architecture can contain multiple layers. The multilayer ANN architecture makes it easy to solve nonlinear problems. Multilayer structures consist of at least three layers. An input layer, an output layer must be present, and least one hidden layer between them. Connections exist only between cells on different layers. [8]. These cells also update their weights and connections according to the error rate of the returned data and form the most suitable structure. Once this structure is established, the system can properly check for new incoming data. This method is used in many different areas.

2. Related works

ANN provides effective solutions in system modeling [9]. Although FLCs are effective in many applications, both of the software and hardware design of a FLC’s presents a number of difficulties. In addition, the controller contains multiple mathematical operations causing the controller to react slowly. In the literature, there are only few studies regarded with modeling of FLCs by ANN in control applications.

Diaz and Soriano [10] reported that DC-DC converters are high-efficiency tools used to power different systems, have a nonlinear behavior and may affect the stability of differences in the main parameters. In this study, different control strategies, linear and nonlinear controllers applied to a Buck converter are examined and compared. There are three main control strategies discussed in the study. First, an optimal control-based design was used using Quadratic Performance Index, the second was a knowledge-based fuzzy control, and thirdly an ANN was proposed as a dynamic emulator of fuzzy control. Some comparisons have been made on the variation of several factory parameters about the systems created by the plant and the controller. In addition, the calculation time in the simulation was compared between the two intelligent controllers.

In another study, fractional order PID ($PI^{\lambda}D^{\mu}$) control was considered. This is a challenge for real-time applications, since the implementation of fractional-order operators is achieved with integer-grade transfer functions. In the present article, mathematical difficulties are mentioned in the design stage of the $PI^{\lambda}D^{\mu}$ controller and ANN design issues that will give the same performance as the $PI^{\lambda}D^{\mu}$ controller are examined [11, 12]. Önder et al., used ANN for predict of nonlinear system dynamics [13].

This study aims to imitation of the FLC by using ANN to be easy to design both in software and hardware, and has a short response time. For the imitation process ANN is used. The reason for using ANN is that it has parallel operation feature. Parallel processing enables the processing speed to be increased. This improves the control performance of the controller and allows for easy design. The design process of the FLC and the training processes of the ANN were studied in Matlab SIMULINK environment. The performance of the designed controllers has been tested on the inverted pendulum system. In the study, FLC was modelled at high similarity ratio with small ANN structure. ANN results were obtained very faster than the FLC control performance. The control performances of two controllers were observed to be very close to each other. As a result, ANN model has smaller structure than FLC, which makes it possible to implement the controller easily in terms of hardware and software.

3. Mathematical background

3.1. Fuzzy logic controller

In the classical set theory, the degree of membership of an element is as crisp as 0 and 1. On the other hand, in fuzzy set theory, a different kind of set definition has been made. In fuzzy sets, membership values can be considered as infinitely real number equivalents in the closed range [0 1] instead of crisp sets. Fuzzy logic control system is actually a structure created by using the sentences we use in our daily life, such as increase the temperature, a little slow down the cart speed. FLCs also fulfills the task of control using an algorithm encoded with terms that do not express such certainty when deciding on a case. The main goal here is to think more humane and to provide a more realistic control [3]. A FLC architecture is shown in Fig. 1.

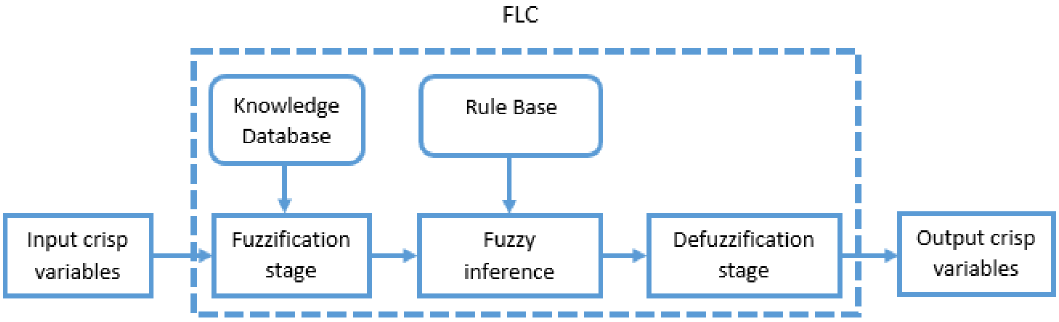


Figure 1. FLC controller structure.

In fuzzy set theory, the degree of belonging of an element to a set is realized by the membership functions. Real-life crisp quantities are passed through membership functions and fuzzy membership degrees are calculated, and these calculated membership degrees and input quantity are included in fuzzy subsets. In fuzzy sets, the degree of membership of an element is indicated by μ. The selection of membership functions on the FLC greatly affects the control performance.

Fuzzification is the process of converting crisp real-life values into fuzzy values. In any fuzzification process, quantities such as speed and temperature are called antecedent variables. These antecedent values applied to some membership functions, and so obtained fuzzy values in the closed [0 1] range. FLC designs use a varying number of membership functions for each antecedent variable. Usually an odd number such as 3, 5 and 7 membership functions are preferred. It is possible to use the same or different membership functions for input and output variables in applications. The fuzzy values from the membership functions are subjected to a number of fuzzy operations in the FLC. These are the rules for the operation of the FLC. The number of these rules is proportional to the number of membership functions. Therefore, as membership functions increase, more rules are needed. The search for the number of rules provides better control results in most cases. As a disadvantage, the increase in the number of rules increases the number of fuzzy processes and therefore makes it difficult to implement FLC both of software and hardware.

In the fuzzy inference stage, the input variable obtained by fuzzification process is applied to the fuzzy operator such as min, max by fuzzy rules. Fuzzy rules are fuzzy instructions prepared with expert experience. These instructions indicate where the fuzzified input variable is directed at the output according to the values it receives in different fuzzy subsets. The realization of a fuzzy rule is the process of subjecting it to operators in fuzzy set theory. The fuzzy result values obtained after the fuzzification and fuzzy inference processes should be converted to the actual control signal applicable to the controlled system. The process of converting fuzzy values into crisp values is called defuzzification process. There are different methods used for defuzzification. Due to the difference in the way these methods work, differences in results may occur depending on the type of study. Expert experience in FLC design may still be required here. Nowadays, the method suitable for the needs of the system in need of control can be tested by simulation and trial studies.

3.2. Artificial neural networks

ANN is a very complex nonlinear calculation tool that can model extremely complex functions. It is preferred where it is impossible or very difficult to find any given functional relationship between the sequence of inputs and the corresponding sequence of outputs. ANNs can be represented by means of an appropriately selected architecture and a successful learning result created by ANN [14]. The basic element of ANN is the cells. Cells, the main processing element of ANN, are nonlinear. In this way, it has become one of the most important tools for the solution of complex nonlinear problems and their architecture usually has parallel distributed cells. This parallel structure allows easy learning. Thanks to its generalization feature, it can produce appropriate responses for entrances not encountered. Thus, it can solve even complex problems of unknown function [7].

ANNs are composed of cells. These cells are used in groups in layers. An ANN input layer consists of at least three layers, the hidden layer and the output layer. There are no cells in the input layer. Input signals from the input layer are transmitted to the cells in the hidden layer by weights. The number of hidden layers may be multiple according to the application. The connections between the cells between the hidden layers and the cells in the output layer are carried out by means of weights. The cells in the hidden layer and the output layer are the basic information processing units within the ANN. There are three basic elements in the cell model. These are synapses, collector and activation functions. The synapses between the cells consist of directed connections, and these connections have weights. The collector is used to collect the inputs. The activation function is used to process a neuron output in a way that is necessary to update weights [15].

There are several network architectures for ANN. Structuring the neurons of a neural network is closely related to the learning algorithm used to train the network. In general, network architectures can basically define three different classes of network architecture [16].

One of these is a single layer network structure, in the simplest form it connects directly to a neuron output layer. It is the second class of a feedforward neural network. The compute nodes comprise one or more inter layers, called hidden layers, that contain inter layer cells. The function of the cells in the hidden layers is to interfere in some way between the input and the output. Adding one or more hidden layers can help generate higher-level statistics from the input of the network. A recurrent neural network separates itself from a forward neural network because it has at least one feedback loop. For example, a recurrent network may consist of a single layer of neurons that feeds the output signal of each cell to the inputs of all other neurons. There are no self-feedback loops in the network [16]. In the Fig. 2, single-layer and multilayer ANN examples are shown.

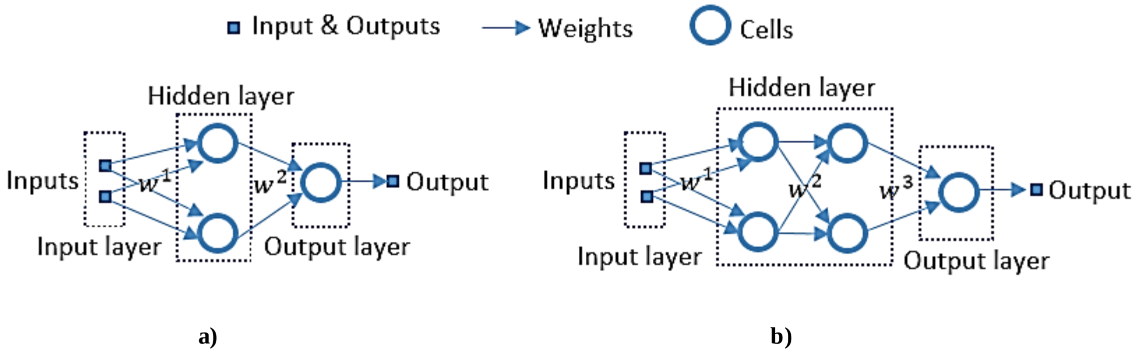


Figure 2. a) Single-layer ANN b) Multilayer ANN

3.3. Cart and inverted pendulum system

The inverted pendulum is a classical control problem which has been the subject of many articles in order to test the control performance of the designed a controller. As shown in Fig. 3, it consists of a cart that can move linearly along the *x*-axis and a rod that can move angularly in the *x-y* axis with a hinge. This system can be likened to a rod held upright in the palm. The inverted pendulum has a nonlinear transfer function and can therefore easily become unstable. A general representation of a cart-inverted pendulum free body diagram is shown in the Fig. 3.

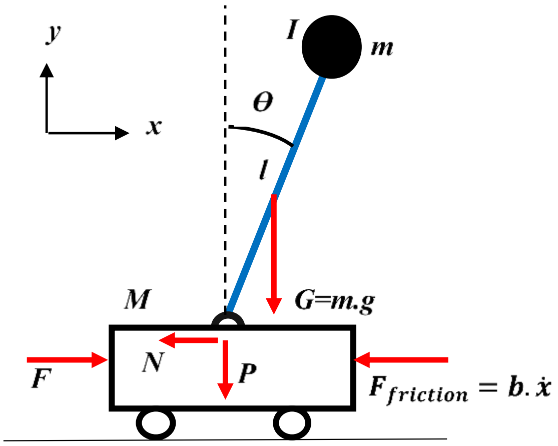


Figure 3. Cart-inverted pendulum free body diagram.

The symbols used in the Fig. 3 and in the following equations, and their values used in the simulation studies in this article are presented in Tab. 1 below.

Table 1. Symbols, their explanations and values used in the study.

Symbols	Explanations	Values	Units
M	Mass of cart	0.455	*kg*
m	Mass of pendulum	0.21	*kg*
l	Length of pendulum	0.61	*m*
g	Acceleration of gravity	9.8	m/s^2
Θ	Angle of pendulum	Variable	*rad*
G	Weight of pendulum	2.0580	*N*
F	Force	Variable	*N*
$F_{friction}$	Friction force	Neglected	*N*
b	Friction coefficient	Neglected	*N*s/m*
x	Distance on *x* axis	Variable	*m*
N	weight component on *x* axis	Variable	*N*
P	weight component on *y* axis	Variable	*N*

According to free body diagram shown in Fig. 3, and Newton's second law;

$$\ddot{x} = \frac{1}{M}\sum_{cart} F_x = \frac{1}{M}(F - N - b\dot{x}) \tag{2}$$

$$\ddot{\theta} = \frac{1}{I}\sum_{pendulum} \tau = \frac{1}{I(-Nl\cos\theta - Pl\sin\theta)} \tag{3}$$

The dynamic equations in the *x* and *y* directions of the pendulum are as follows.

$$N = \sum_{pendulum} F_x = m\ddot{x}_p \tag{4}$$

$$P = mg + \sum_{pendulum} F_y = m(\ddot{y}_p + g) \tag{5}$$

However, the *x* and *y* position coordinates of pendulum can be represented with equations as following.

$$x_p = x + l\sin\theta \tag{6}$$

$$\ddot{x}_p = \ddot{x} - l\dot{\theta}^2\sin\theta + l\ddot{\theta}\cos\theta \tag{7}$$

$$y_p = -l\cos\theta \tag{8}$$

$$\ddot{y}_p = l\dot{\theta}^2\cos\theta + l\ddot{\theta}\sin\theta \tag{9}$$

The $\ddot{x}_p$ and $\ddot{y}_p$ values in Eq. (7) and Eq. (9) can be substituted in Eq. (4) and Eq. (5). So the following equations are obtained.

$$N = m\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta \tag{10}$$

$$P = m(l\dot{\theta}^2\cos\theta + l\ddot{\theta}\sin\theta + g) \tag{11}$$

The *N* and *P* values in Eq. (10) and Eq. (11) can be substituted in Eq. (2) and Eq. (3). So the following equations are obtained.

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = F \tag{12}$$

$$Psin\theta + Ncos\theta - mgsin\theta = ml\ddot{\theta} + m\ddot{x}cos\theta \tag{13}$$

$$-Plsin\theta - Nlcos\theta = I\ddot{\theta} \tag{14}$$

$$(I + ml^2)\ddot{\theta} + mglsin\theta = -ml\ddot{x}cos\theta \tag{15}$$

In order to get rid of the nonlinearities caused by *sin* and *cos* functions in the above equations, the following approaches are made by accepting that the pendulum changes in a little ϕ range.

$$cos\theta = \cos(\pi + \phi) \approx -1 \tag{16}$$

$$sin\theta = \sin(\pi + \phi) \approx -\phi \tag{17}$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \tag{18}$$

These approaches are in Eq. (16), Eq. (17) and Eq. (18) can be substituted in Eq. (12) and Eq. (15). And also *U* is used instead of *F*. So the following equations are obtained.

$$(I+ml^2)\ddot{\phi}-mgl\phi=ml\ddot{x} \tag{19}$$

$$(M+m)\ddot{x}+b\dot{x}-ml\ddot{\phi}=u \tag{20}$$

Laplace transform of the Eq. (19) and Eq. (20) at zero initial conditions are shown below.

$$(I+ml^2)\Phi(s)s^2-mgl\Phi(s)=mlX(s)s^2 \tag{21}$$

$$(M+m)X(s)s^2+bX(s)s-ml\Phi(s)s^2=U(s) \tag{22}$$

The transfer functions of the pendulum and the cart from Eq. (21) and Eq. (22) are obtained as shown below.

$$P_{pend}(s)=\frac{\Phi(s)}{\mathrm{U(s)}}=\frac{\frac{ml}{q}s}{s^3+\frac{b(I+ml^2)}{q}s^2-\frac{(M+m)mgl}{q}s-\frac{bmgl}{q}}\quad\left[\frac{rad}{N}\right] \tag{23}$$

$$P_{cart}(s)=\frac{X(s)}{U(s)}=\frac{\frac{(I+ml^2)s^2-gml}{q}}{s^4+\frac{b(I+ml^2)}{q}s^3-\frac{(M+m)mgl}{q}s^2-\frac{bmgl}{q}s}\quad\left[\frac{m}{N}\right] \tag{24}$$

Where q value;

$$q=[(M+m)(I+ml^2)-(ml)^2] \tag{25}$$

4. Simulation studies and results

To perform simulation works, the cart-inverted pendulum model was created using Eq. (23) and Eq. (24) in Simulink. The design of the FLCs was made with Fuzzy Toolbox. The ANN design for the modelling of the FLC was made using Matlab’s Neural Network Toolbox.

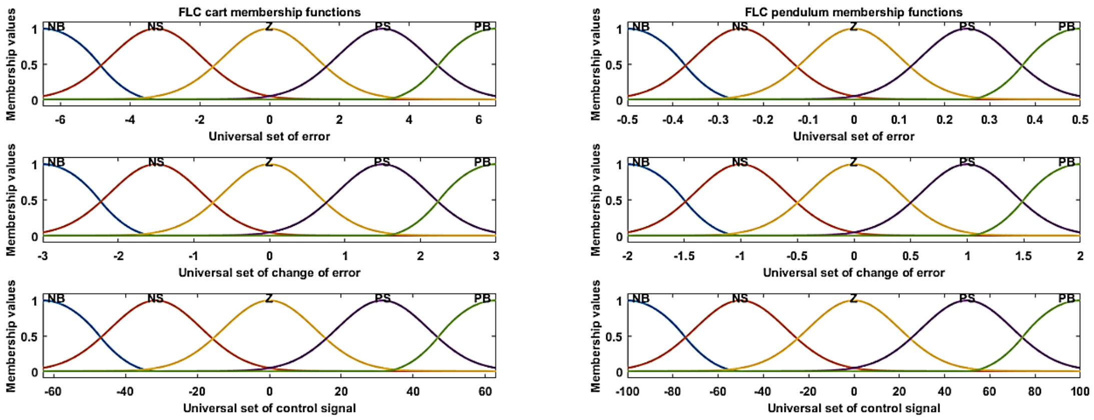
4.1. FLC design

In both FLCs used to control the cart-inverted pendulum, the error (e) and the change of the error (ce) were used as input variables. However, the same input membership functions and the same output membership functions were used in both FLCs. The only difference is the universal set range of variables. The universal set of input and output variables is shown in Tab. 2.

Table 2. Fuzzy subset ranges of variables on the universal set.

Cart	**NB**	**NS**	**Z**	**PS**	**PB**
e	[-6.5 -3.25]	[1.3 -3.25]	[1.3 0]	[1.3 3.25]	[3.25 6.5]
ce	[-3 -1.5]	[0.6 -1.5]	[0.6 0]	[0.6 1.5]	[1.5 3]
u	[-63 -31.5]	[12.6 -31.5]	[12.6 0]	[12.6 31.5]	[31.5 63]
Pendulum	**NB**	**NS**	**Z**	**PS**	**PB**
e	[-0.5 -0.25]	[0.1 -0.25]	[0.1 0]	[0.1 0.25]	[0.25 0.5]
ce	[-2 -1]	[0.4 -1]	[0.4 0]	[0.4 1]	[1 2]
u	[-100 -50]	[20 -50]	[20 0]	[20 50]	[50 100]

The membership functions used are shown in Fig. 4 and Tab. 2, a total of 5 gauss membership functions are used: Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Big (PB).



a) **b)**

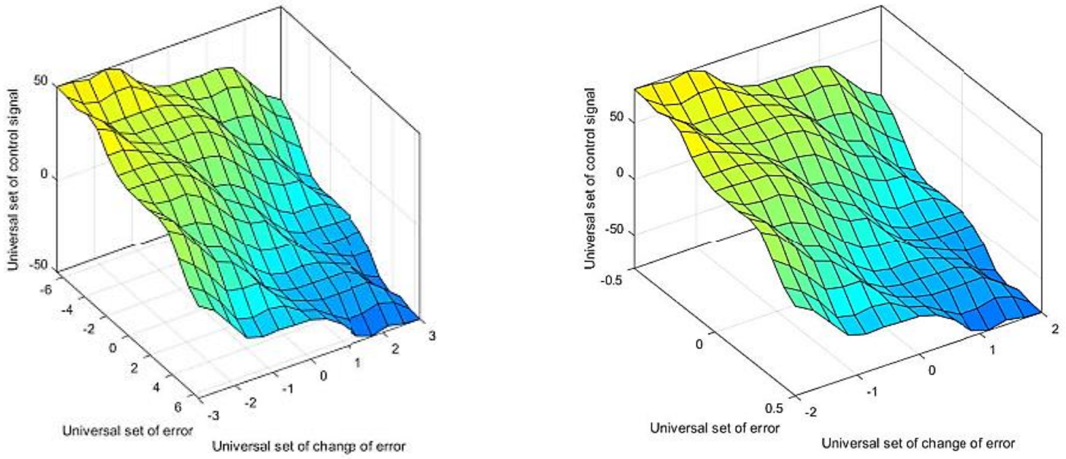
Figure 4. a) Membership functions of cart. b) Membership functions of pendulum.

The rules established according to the membership functions in Fig. 4 are shown in Tab. 3. As can be seen from Tab. 3, there are a total of 25 rules in FLCs.

Table 3. Rule table of FLCs

		e				
		NB	**NS**	**Z**	**PS**	**PB**
ce	**NB**	PB	PB	PS	PS	Z
	NS	PB	PS	PS	Z	NS
	Z	PS	PS	Z	NS	NS
	PS	PS	Z	NS	NS	NB
	PB	Z	NS	NS	NB	NB

The relationship between the input variables and the output variables according to the rules in Tab. 3 is shown in Fig. 5.



a) **b)**

Figure 5. Representation of FLC's relationships between input and output variables a) Cart b) Pendulum.

The exact design used for FLC training is shown in Fig. 6. As shown in Fig. 6, the position controls of the cart and inverted pendulum are realized as closed loop control structure with two separate FLC units. For FLC_cart, the error of the cart position error and the change of the error of the cart position are used as the input signal, while the change of the error of the pendulum angle and the error of the pendulum angle is used as the input signal for FLC_pend. The outputs of these two FLCs were summed, and this final check mark was intended to move the cart to the desired position and to keep the pendulum upright without falling.

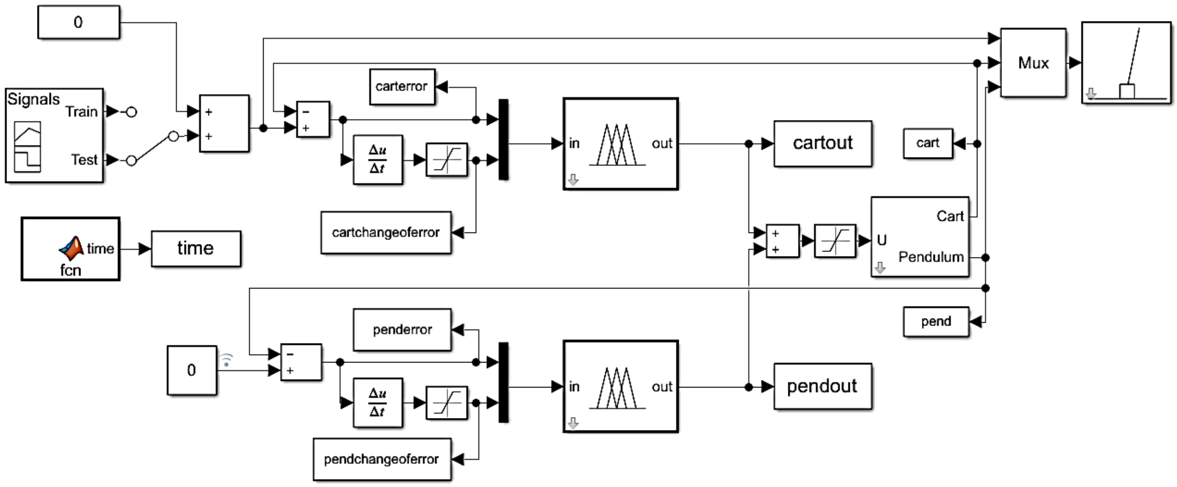


Figure 6. The exact design used for FLC training.

4.2. Training of ANN

On the FLC design made on Matlab Simulink, different reference values were given for cart and control was performed. During the control process, input variables (e and ce) and output variables (u) of FLC_cart and FLC_pendulum were transferred from Simulink to the Matlab workspace with the "to workspace" block. Thus, ANN's training set was created. e and ce values are taken as the input variables of the training set and u is taken as the output variable of the training set. The same architecture was used for both FLC_cart and FLC_pend ANNs. Feed forward network structure was used including each ANN, 2 variable input layer, two 5-cell hidden layers and one variable output layer. “tansig” function was used as the activation function in the hidden layers and “purelin” function was used as the activation function in the output layer.

The success rate of the training was measured according to the MSE performance. The training was done with the Levenberg-Marquardt algorithm called “trainlm” in the Matlab. The Levenberg-Marquardt algorithm update derived from steepest descent and Newton algorithms was given in Eq. (26). [17]

$$\Delta w = (J^T J + \mu I)^{-1} J^T e \tag{26}$$

Where w is the weight vector, I is the unit matrix, μ is the combination coefficient, J represents the Jacobian matrix in dimension *(PxM)xN*, e represents the error vector in dimension *(PxM)x1*. And P is the number of training samples, M is the number of outputs, N is the number of weights. μ is an adjustable variable. If this parameter is too big, the method behaves like a steepest descent method, if it is too small the method behaves like a Newton method. An adaptive structure for this parameter was given in Eq. (27). [17]

$$\mu(n) = \begin{cases} \mu(n-1)k & E(n) > E(n-1) \\ \mu(n-1)/k & E(N) \leq E(n-1) \end{cases} \tag{27}$$

In Eq. (27), k is a constant number. E shows the fitness value.

The contents of the ANN block, which ANN block placed instead of the FLC block, were shown in the Fig. 7. These contents were created by Matlab during the ANN's training and then turned into Simulink block with "gensim" function. The Feed-forward network was used as ANN structure. The ANN contains ten cells, two inputs, an output, four layers, three activation functions, collectors, and many weights. The first layer is the input and the last layer is the output. Each layer has biases and also weights coming from the previous layer.

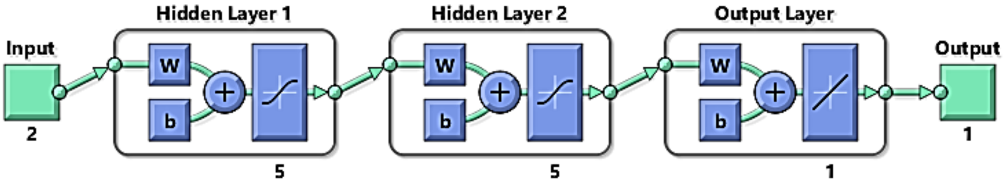


Figure 7. Ann structure and contents

The ANN training process is as follows.

Fig. 8 shows the graph obtained during ANN training. 500 iterations were made for the training. The mean square error (MSE) rate during training of cart decreased to 0.3669 at 359 iterations. MSE rate during training of pendulum decreased to 0.0078433 at 500 iterations.

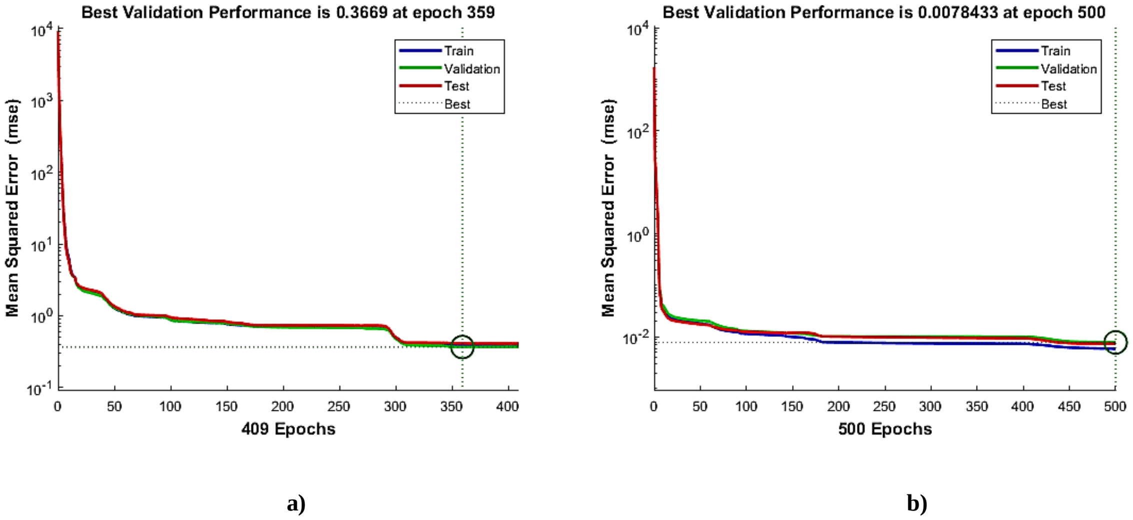
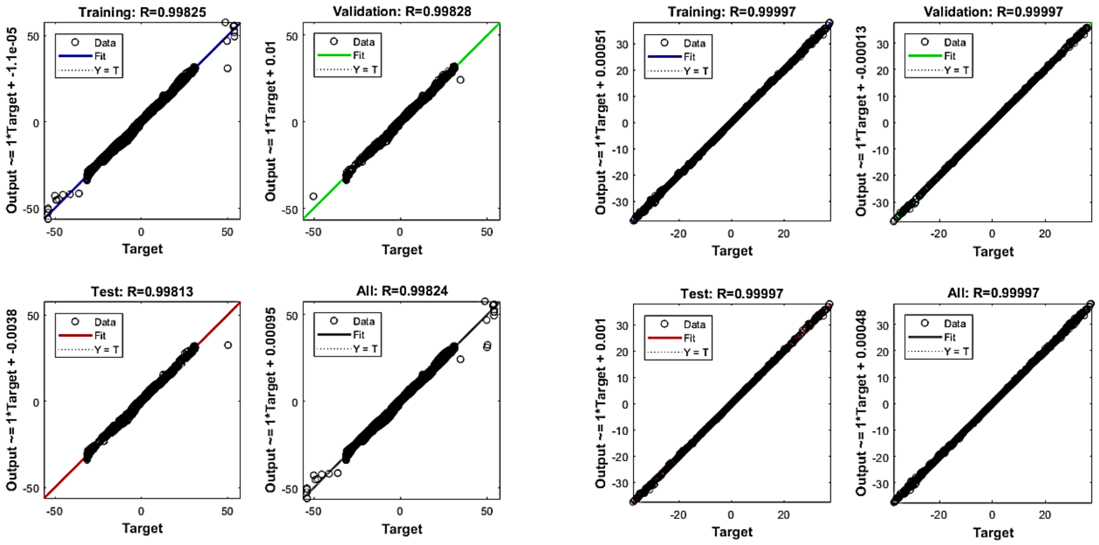


Figure 8. ANN's training performance graph. a) cart b) pendulum

The regression graph plotted during ANN training is shown in Fig. 9. Here, the collection of data on the regression curve is an indicator of a good training.



a) **b)**

Figure 9. ANNs training regression graphs. a) cart b) pendulum

Fig. 10 shows a block diagram of the cart-pendulum system. Here, the FLCs were removed and replaced with ANNs.

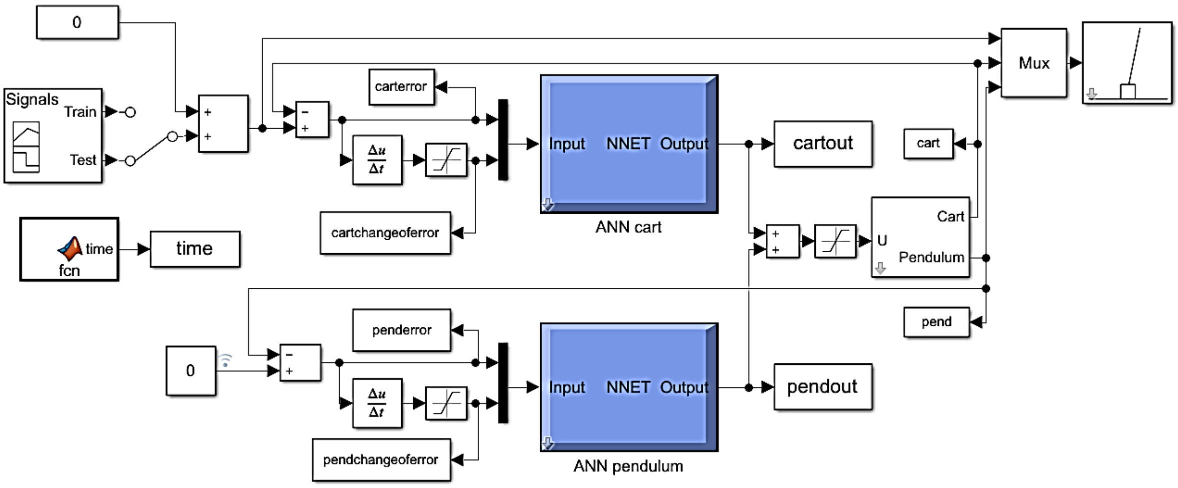


Figure 10. Control block diagram created with ANNs instead of FLCs.

The Simulink Diagram was run with the test set, which completely different from training set. The FLC and the ANN controllers were run separately and tested. And the results obtained are as follows.

Fig. 11 is a position graph of the cart and it shows the comparison between the control results of the two controllers. Here, the MSE value calculated based on the difference between the cart position controlled by the FLC and the reference position is 10.1312. Similarly, the MSE value calculated based on the difference between the cart position controlled by the ANN and the reference position is 10.1177. As can be seen, the MSE values of both controllers are very close to each other. This shows us that ANN is able to imitate FLC highly rate. Furthermore, the calculated MSE value which based on the difference between cart position control performances of these two controllers is 0.000806, which is very small.

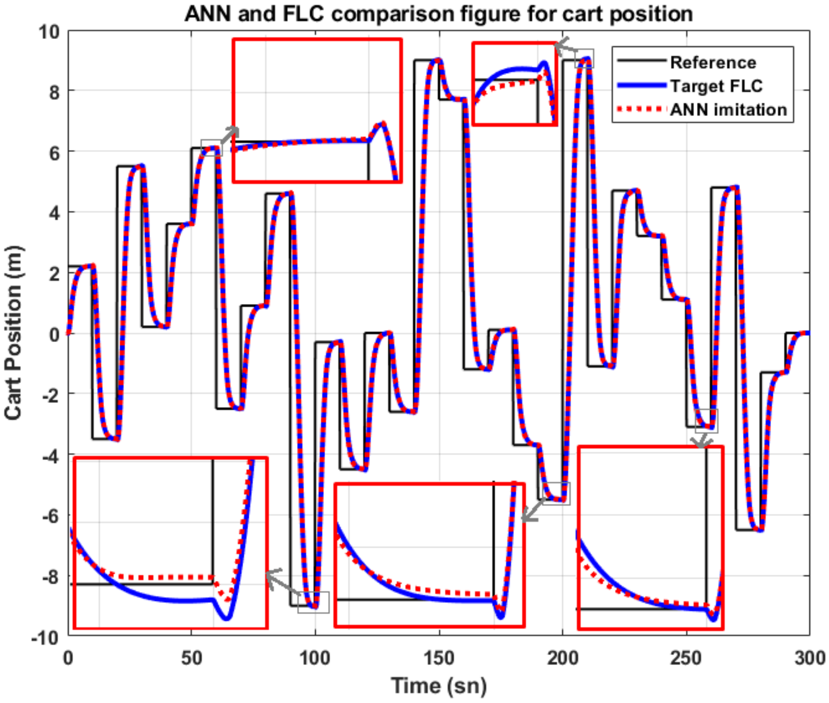


Figure 11. Position graph of the cart position controlled by ANN and FLC.

Fig. 11 is the position graph of the cart, and it shows the comparison between the control results of two controllers. In order to make this difference more clarity, the position difference graph of Fig. 12 is obtained by subtracting the position of the cart controlled by the ANN from the position of the cart controlled by the FLC. When Fig. 12 is examined, it is seen that the biggest difference between the control results of two controllers is 15 cm and this difference closes in a short time.

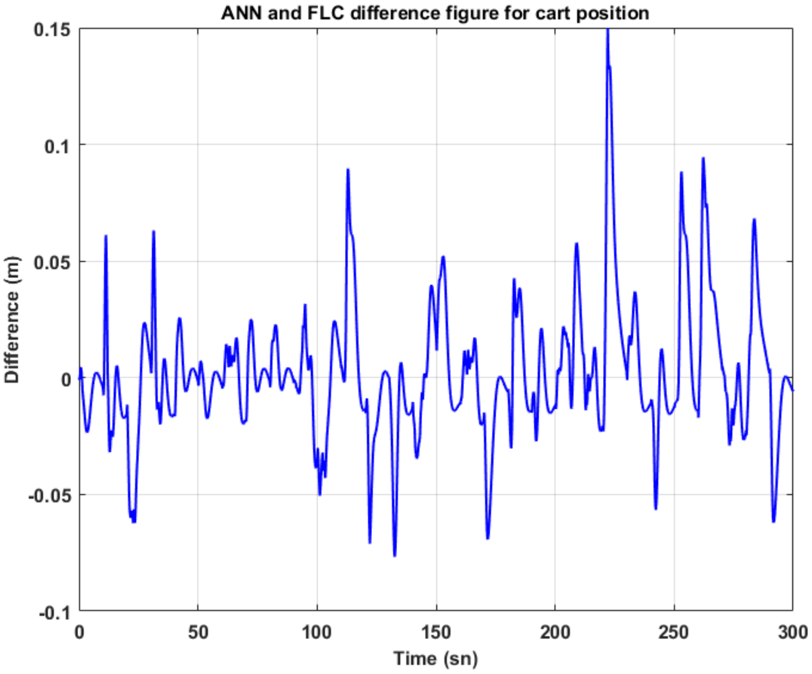


Figure 12. Difference graph of cart position control results between FLC and ANN.

Fig. 13 is an angle graph of the pendulum and it shows the comparison between the control results of FLC and ANN controllers. Here, the MSE value calculated based on the difference between the pendulum angle controlled by the FLC and the reference angle (0 radian) is 0.0032098. Similarly, the MSE value calculated based on the difference between the pendulum angle controlled by the ANN and the reference angle is 0.0031916. As can be seen, the MSE values of both controllers are very close to each other as cart position. This shows us that ANN is able to imitate FLC highly rate. Furthermore, the calculated MSE value which based on the difference between pendulum angle control performances of these two controllers is $1.595*10^{-5}$, which is very small.

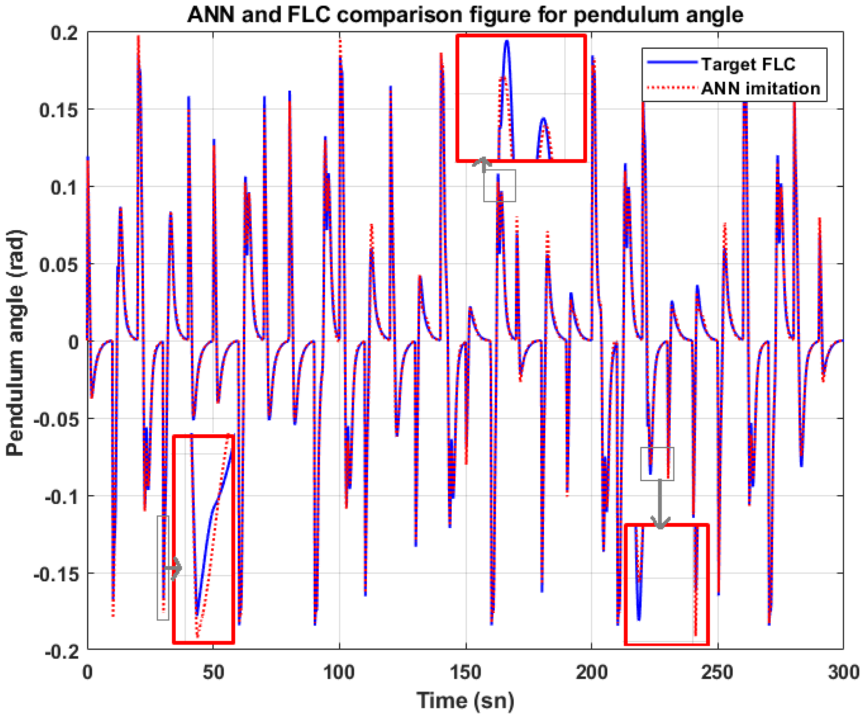


Figure 13. Position graph of the pendulum angle controlled by ANN and FLC.

Fig. 13 is the angle graph of the pendulum, and it shows the comparison between the control results of two controllers. In order to make this difference more clarity, the angle difference graph of Fig. 14 is obtained by subtracting the angle of the pendulum controlled by the ANN from the angle of the pendulum controlled by the FLC. When Fig. 14 is examined, it is seen that the biggest difference between the control results of two controllers is 0.038 radian and this difference closes in a short time.

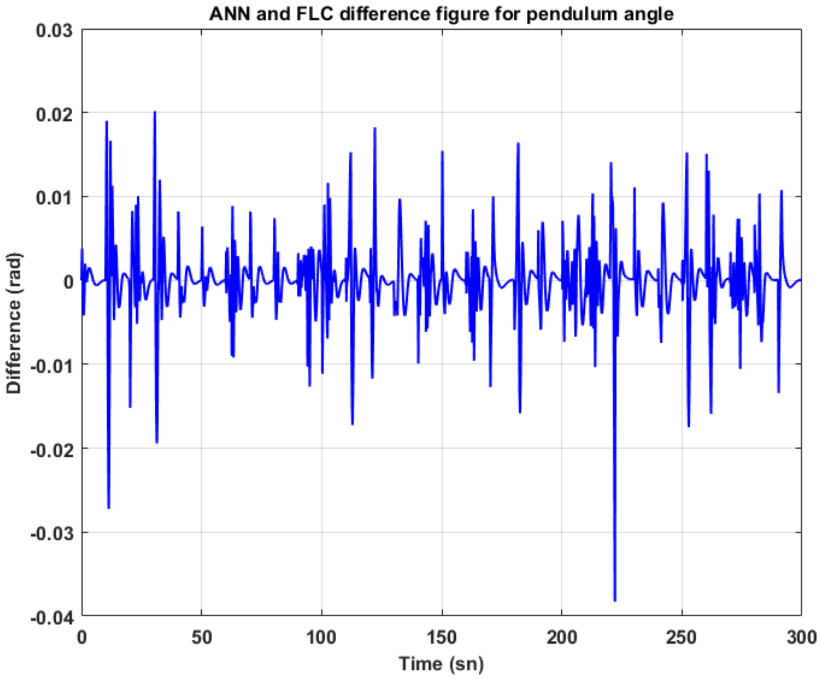


Figure 14. Difference graph of pendulum angle control results between FLC and ANN.

5. Conclusion

The ANN can be used for the imitation process of many systems, the FLC is one of them. The ANN has a parallel operation feature and so it runs very faster than the FLC. Further, the ANN design both in software and hardware is easier than the FLC design.

With the using of the data of the FLC's input and output, the trained ANN successfully imitated the FLC and managed to perform the control process under different reference values. According to the position values of the car, the MSE value of the difference between the FLC and the ANN controllers is 0.000806. For the angle values of the pendulum, the MSE value of the difference between the FLC and the ANN controllers was obtained as $1.595*10^{-5}$.

Since the operations used by ANN are easier than those used by the FLC, while a 300-second simulation was performing with 178.799 real seconds with FLC, the same simulation was performed with ANN in 2.059 real seconds. The system designed with ANN will be easier to use and design in various control applications as it is containing less complex mathematical operations. This study can be improved by using more training data, more cell numbers, more iteration or different training algorithms in the ANN training process.

References

[1] Sayğan, S. (2014). Örgüt Biliminde Karmaşıklık Teorisi. Ege Academic Review, 14(3).

[2] Ramezani, M. R., Kamyad, A. V. (2010). Approximation of general nonlinear control systems with linear time varying systems. In 2010 18th Iranian Conference on Electrical Engineering (pp. 680–685). Presented at the 2010 18th Iranian Conference on Electrical Engineering, Isfahan, Iran. https://doi.org/10.1109/IRANIANCEE.2010.5506987

[3] Altaş, İ. H. (1999). Bulanık Mantık: Bulanıklılık Kavramı. Enerji, Elektrik, Elektromekanik-3e, 62, 80–85.

[4] Pamuk, Z., Yurtay, Y., Yavuzyilmaz, O. (2015). Establishing the Potential Clients Using Artificial Neural Networks. Balkan Journal of Electrical and Computer Engineering, 3, 219–224.

[5] Yegnanarayana, B. (2009). Artificial Neural Networks. PHI Learning Pvt. Ltd.

[6] Tolon, M., Tosunoğlu, N. G. (2008). Tüketici Tatmini Verilerinin Analizi: Yapay Sinir Ağlari Ve Regresyon Analizi Karşilaştirmasi. Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 10(2), 247–259.

[7] Ergezer, H., Dikmen, M., Özdemir, E. (2003). Yapay sinir ağları ve tanıma sistemleri. PiVOLKA, 2(6), 14–17.

[8] Kilic, E., Ozbalci, U., Ozcalik, H. R. (2012). Lineer Olmayan Dinamik Sistemlerin Yapay Sinir Ağları ile Modellenmesinde MLP ve RBF Yapılarının Karşılaştırılması. ELECO2012 Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyomu,(29.11. 2012-01.12. 2012).

[9] Tasdemir, S. (2018). Artificial Neural Network Model for Prediction of Tool Tip Temperature and Analysis. International Journal of Intelligent Systems and Applications in Engineering, 6(1), 92–96. https://doi.org/10.18201/ijisae.2018637937

[10] Diaz, N. L., Soriano, J. J. (2007). Study of Two Control Strategies Based in Fuzzy Logic and Artificial Neural Network Compared with an Optimal Control Strategy Applied to a Buck Converter. In NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society (pp. 313–318). Presented at the NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society. https://doi.org/10.1109/NAFIPS.2007.383857

[11] Efe, M. Ö. (2011). Neural Network-Assisted $PI^{\lambda}D^{\mu}$ Control. Fractional Dynamics and Control pp 19-31

doi: 10.1007/978-1-4614-0457-6_2

[12] Efe, M. Ö. (2011). Neural Network Assisted Computationally Simple $PI^{\lambda}D^{\mu}$ Control of a Quadrotor UAV. IEEE Transactions On Industrial Informatics, vol. 7, no. 2

[13] Efe, M. Ö., Kaynak, O., Abadoglu, E. (1999). Neural Network Assisted Nonlinear Controller For A Bioreactor, International Journal of Robust and Nonlinear Control 9(11),799-815

doi: 10.1002/(SICI)1099-1239(199909)9:11<799::AID-RNC441>3.0.CO;2-U

[14] Marini, F., Bucci, R., Magrì, A. L., Magrì, A. D. (2008). Artificial neural networks in chemometrics: History, examples and perspectives. Microchemical Journal, 88(2), 178–185. https://doi.org/10.1016/j.microc.2007.11.008

[15] Sreelakshmi, K., Ramakanthkumar, P. (2008). Neural networks for short term wind speed prediction. World Academy of Science, Engineering and Technology, 42, 721–725.

[16] Haykin, S. S. (2009). Neural networks and learning machines/Simon Haykin. New York: Prentice Hall,.

[17] Çavuşlu, M. A., Becerikli, Y., Karakuzu, C. (2016). Levenberg-Marquardt Algoritması ile YSA Eğitiminin Donanımsal Gerçeklenmesi. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5(1).