Design and Performance of Solar Tracking System with Fuzzy Logic Controller

M. A. Usta[1], Ö. Akyazı[2] and İ. H. Altaş[3]

[1]Karadeniz Technical University, Trabzon/Turkey, mausta@ktu.edu.tr
[2]Karadeniz Technical University, Sürmene Abdullah Kanca MYO, Trabzon/Turkey, oakyazi@ktu.edu.tr
[3]Karadeniz Technical University, Trabzon/Turkey, ihaltas@ktu.edu.tr

***Abstract*—A solar tracking system is designed to optimize the operation of solar energy receivers. More solar energy is collected by the end of the day if solar receivers are installed with a tracker system. In this paper, a solar tracking system is modeled using Matlab/Simulink and a fuzzy logic control is designed for the control of this system. The generated controller was combined with the solar tracking system and the control was realized with the fuzzy logic controller in the Matlab/Simulink environment. At the same time, PI control is applied to the system and the results obtained with PI control were compared with the results of fuzzy logic.**

***Keywords*—Fuzzy Logic Controller, Solar Tracking System, Matlab/Simulink GUI Simulation, Permanent Magnet DC Motor (PMDC).**

I. INTRODUCTION

Consumption of electrical energy in the world is constantly growing. Most of used and produced electrical energy is obtained by combustion of fossil fuels or by nuclear processes. Thermal power plants and nuclear power plants are natural polluters of environment. Alternative energy sources that we are surrounded with, on the other hand are pure ecological energy sources [1]. The main alternative energy sources include solar energy, geothermal energy, wind energy, wave energy, bio-energy and hydrogen technologies. One of particularly important sources of energy is solar energy. Solar energy can play a very important role in providing most of the heating, cooling and electricity needs of the world and also has the potential to solve our environmental problems. The sun is infinite and clean energy source and it sends to earth about 10,000 times as much energy of the world's energy consumption. So, solar energy has attracted a great interest during the last two decades. Converting solar radiation energy into electrical energy and usage of this method of production of the necessary energy becomes an important condition for further development and progress of the planet [1].

The daily and seasonal variation of solar radiation is a significant problem in solar energy. Direct usage of solar energy is limited with sunny hours so direct usage of solar energy is not continuous. Therefore, the utilization of solar energy has the limitation of practicality due to high cost and low efficiency. To solve these problems, many researchers have studied about effectively practical methods of solar energy [2]. One of methods is solar tracking system. In this method, the direction of solar panel changes according to sunlight position during the course of a day. Thus, solar tracking systems are designed in different ways with the aim to utilize a high rate from solar radiation. Solar tracking systems provide an annual basis up to 35% increase in efficiency of energy production. Efficiency values vary according to seasons and especially in the winter months the efficiency of the solar tracking systems are higher than the efficiency of the fixed systems.

Many studies have been done on the control methods to fulfill the expectations in many systems including solar tracking systems. One of these studies has focused on fuzzy logic-based control. A fuzzy logic-based control scheme can be designed using the accumulated experience about the system behavior rather than the availability of the mathematical model [3]. In other words, machines can be provided to give decisions like humans by using fuzzy logic and fuzzy cluster operations. With the use of fuzzy logic approach increases system performance, simplifies application and reduces financial expenses. In this case, system performance is optimized perfectly and a control can be obtained more effective and responsive.

In this paper, a fuzzy logic control is designed to achieve the required tracking. Conventional control system is also applied to the system and both the control performance are analyzed. The simulation was realized using Matlab/Simulink dynamic system simulation software.

II. MODEL

For optimal efficiency, solar panels should be perpendicular to sunlight where the illumination is strongest. However, since the direction of sunlight changes during the course of a day, and from season to season, a high-performance solar tracking system can maximize usage of the panels [4].

This system consists of two permanent magnet dc (PMDC) motors, two directional light detecting circuits and two amplifiers to drive the motors. The two drive motors are decoupled, i.e. the rotation angle of one motor does not influence that of the other motor, reducing control problems. This implementation minimizes the system's power consumption during operation and increases efficiency and the total amount of electricity generated [4]. The light detecting circuit of design consists of four light sensitive devices, such as LDR, photodiodes or phototransistors, mounted on the solar panel and placed in an enclosure. The four light detectors are screened form each other by opaque surfaces. The sensors are configured in a way that LDR_1 and LDR_2 are used to track the sun horizontally meanwhile LDR_3 and LDR_4 allow tracking the sun vertically [5]. When one receives more light than the other, the panel is not aligned properly and an error voltage results. The error voltage is used as a command to an amplifier circuit to drive the motor and align the panel to be perpendicular to the light source beam. The following figure is shown solar tracking control architecture for one direction. If this figure is represented the vertical control architecture, the horizontal control architecture is the same as this figure.

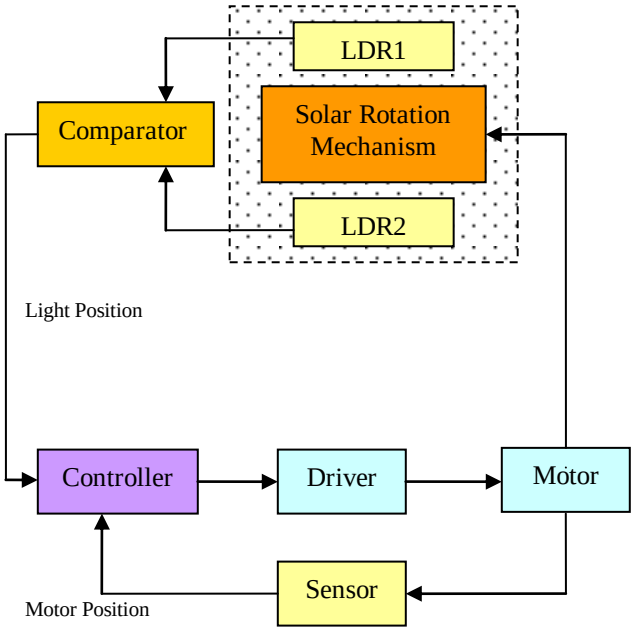


Figure 1: Solar tracking control architecture for one direction (vertical or horizontal)

A. *Photo Detecting Circuit and Amplifier*

The photo detecting circuit and amplifier provide an electrical driving force to the motor which is proportional to the rotational misalignment of the panels to the light source. The photo detecting circuit and amplifier can be considered as a single variable K, where K is a proportionality constant with units of volts per radian. The value of K represents the gain of the 'photo detector' circuit and the 'gain adjust' circuit.

B. *Permanent Magnet DC Motor*

A solar panel is actuated by two permanent magnet dc (PMDC) motors. We'll explain how to design a PMDC motor for Matlab/Simulink GUI Environment. PMDC motors are used in a wide variety of low-power applications. The field winding is a permanent magnet. Permanent magnets offer a number of useful benefits such as, they don't require external excitation, less space requirement and they are cheaper. The equivalent circuit of PMDC motor is as in Figure 2 and the equations are given by (1), (2), (3) and (4).

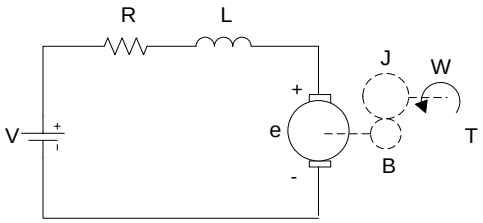


Figure 2: Equivalent circuit of a PMDC motor.

A motor coil can be modeled as a resistor in series with an inductor. When the motor has a nonzero speed, there is back emf voltage generated. The back emf voltage is proportional to the speed. According to Kirchoff's Voltage Law:

$$V(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} + E(t) \tag{1}$$

Where V is the voltage applied to the motor and

$$E(t) = K \cdot \omega(t) \tag{2}$$

The torque generated by the motor is proportional to the current. According to Newton's Second Law (rotational version):

$$T(t) - T_f(t) - T_L(t) = J \cdot \frac{d\omega(t)}{dt} \tag{3}$$

Where T_L is the load torque and

$$T(t) = K \cdot i(t) \text{ and } T_f(t) = B \cdot \omega(t) \tag{4}$$

Equations (1) and (3) describe the dynamic behavior of the motor. Equations (1), (2), (3) and (4) can be re-arranged as in (5) and (6) for construction of the block diagram.

$$\frac{di(t)}{dt} = \frac{1}{L} \cdot \left(V(t) - R \cdot i(t) - K \cdot \omega(t)\right) \tag{5}$$

$$\frac{d\omega(t)}{dt} = \frac{1}{J}\cdot\left(K\cdot i(t) - T_L(t) - B\cdot\omega(t)\right) \tag{6}$$

Differential equations (5) and (6) for armature current and angular speed can be arranged as state-space equations in (7).

$$\begin{bmatrix} \dot{i}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K}{L} \\ \frac{K}{J} & -\frac{B}{J} \end{bmatrix} \cdot \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \cdot \begin{bmatrix} V(t) \\ T_L(t) \end{bmatrix} \tag{7}$$

The following figure is shown Matlab/Simulink model for PMDC motor.

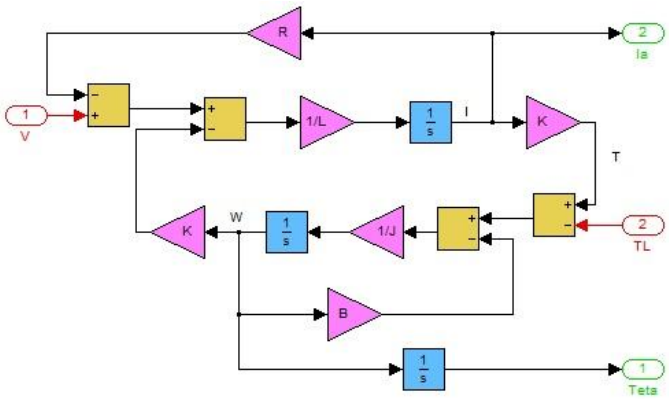


Figure 3: Matlab/Simulink model for PMDC motor

III. DESIGN PROCESS OF THE PROPOSED CONTROLLER

In most research literature, a fuzzy controller system is commonly defined as a system that emulates a human expert. In this case, the knowledge of the human operator would be put in the form of a set of fuzzy linguistic rules. These rules would produce an approximate decision in the same manner a human would do [5]. A block diagram of a fuzzy control system is shown in Figure 4.

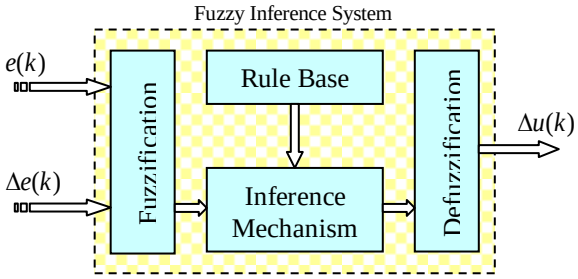


Figure 4: The basic structure of fuzzy logic based controller.

The fuzzy controller is composed of the following four elements. These are fuzzification, rule base, inference mechanism and defuzzification. A fuzzification interface converts the crisp inputs into the fuzzy membership values that are used in the rule base in order to execute the related rules so that an output can be generated [6]. A rule base consists of a data table which includes information related to the system. An inference mechanism emulates the expert's decision making in interpreting and applying knowledge about how best to control the plant [5]. A defuzzification interface converts the conclusions of the inference mechanism into the crisp inputs for the process.

The values of error (e(k)) and its change (Δe(k)) occurring during the operation of the system form the crisp inputs of the system. These two inputs defined as in (8) and (9).

$$e(k) = r(k) - y(k) \tag{8}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{9}$$

r(k), y(k) and k are expressed as the reference input, the actual output of the system and the sampling step respectively.

The model development of the FLC for Matlab/Simulink environment is described after this. The fuzzy membership functions are the basic elements of fuzzy logic controller and absolutely it must be in modeling. In many applications, triangle, trapezoid, bell, Gaussian, sigmoid and sinusoid membership function is used. In this modeling, the triangular membership function given the following equation (10) is used in both fuzzification and defuzzification stages.

$$\mu(x) = \max\left[\min\left(\frac{x - x_1}{x_2 - x_1}, \frac{x_3 - x}{x_3 - x_2}\right), 0\right] \tag{10}$$

The Simulink model of the triangular fuzzy membership function is shown in Figure 5 where x_1, x_2, and x_3 are the crisp parameters used to define the location and shape of the triangle. The input x is the crisp variable whose membership value on this triangle fuzzy subset is the output mu(x) [6].

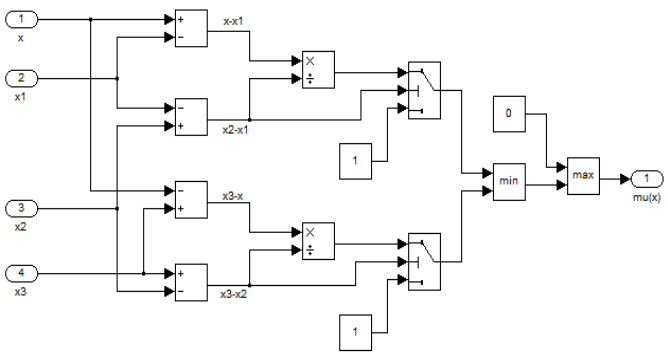


Figure 5: Simulink model of the triangular fuzzy membership function.

The crisp inputs e(k) and Δe(k) are converted to fuzzy membership values on the fuzzy subsets NB, NS, ZZ, PS and PB. The fuzzy inference mechanism has the structure as in verbal expression (11).

If e is A ***and*** Δe is B ***then*** Δu is C. (11)

A, B and C in (11) represent any one of the fuzzy subsets NB, NS, ZZ, PS and PB. The input space in (11) is the part that represented by the expression (e is A ***and*** Δe is B). The verbal connector "and" corresponds to the Boolean operator "min". The weight coefficient requiring for each rule is determined by using the Boolean operator "min" (min($\mu_A(x)$, $\mu_B(x)$)). So, we get the membership values of the each output space. The process of fuzzification of the input space with 9 rules is shown in Figure 6.

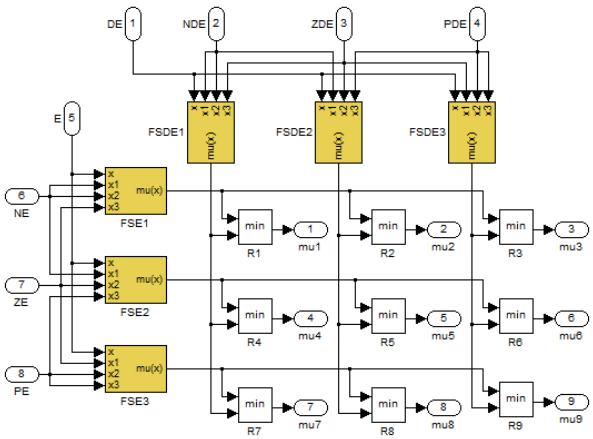


Figure 6: The Simulink model of the verbal expression.

These weight coefficients which are shown as mu1, mu2, …, mu9 are multiplied by the crisp values of each corresponding fuzzy subset in the output space Δu. These crisp values indicate the peak locations of the triangular fuzzy subsets. This multiplication process represents the products in the nominator of defuzzification method called the centre of area. Then the sum of these products is divided by the sum of membership values obtained as in Figure 7.

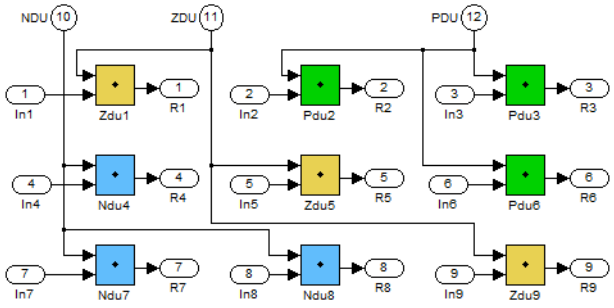


Figure 7: The Simulink model of the fuzzy rules.

Several defuzzification methods have been applied in literature. However, the method called the center of area is widely used in fuzzy logic control applications. The center of area method is given the following equation (12).

$$\Delta U(k) = \frac{\sum_{i=1}^{n} \mu_i(\Delta u_i) \cdot \Delta u_i}{\sum_{i=1}^{n} \mu_i(\Delta u_i)} \quad (12)$$

ΔU(k) is the final crisp output. Simulation model of the center of area method is depicted in Figure 8. This is the final stage of the FLC. A general overlooked view of the FLC is given in Figure 9 where the processes from inputs e and Δe to output Δu are shown. The input data blocks to represent fuzzy membership functions for the error e, error change Δe and the controlled output change Δu are shown in Figure 9. The user is able to edit and change the parameters of the membership functions on this stage without going into the detail of the FLC.

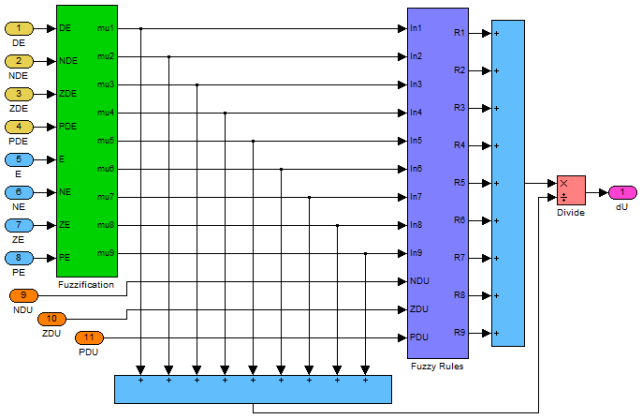


Figure 8: Fuzzy reasoning representing the process from fuzzification to defuzzification.

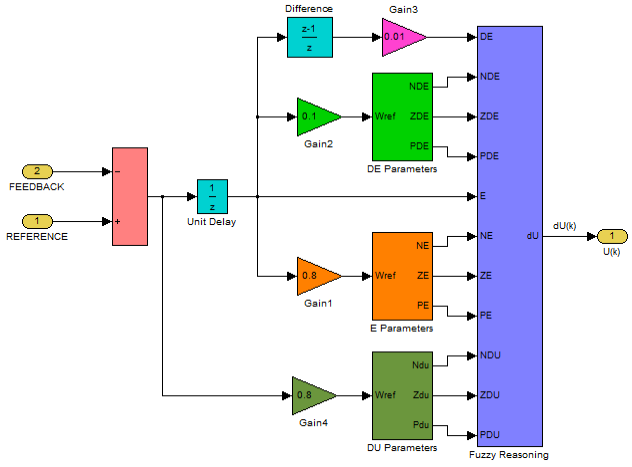


Figure 9: The input and output units of the FLC.

IV. APPLICATION AND RESULTS

Solar tracking system is simulated using both FLC and PI controllers for comparison and validation purposes. The proportional gain (K_P) set to 1.55 and the integral gain (K_I) set to 0.02. Matlab/Simulink model block diagram of this system is shown in Figure 10.

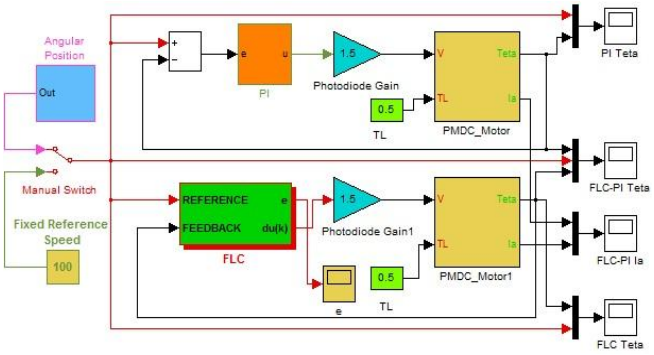


Figure 10: Matlab/Simulink model for the solar tracking system

The related simulation results are given in the following figures. The system responses from both FL and PI controllers for fixed angular position are plotted on the same graph for better comparison. Figure 11 is shown the output of the system for FL and PI controllers. It was observed that the FLC gives faster response and less overshoot than PI controller.

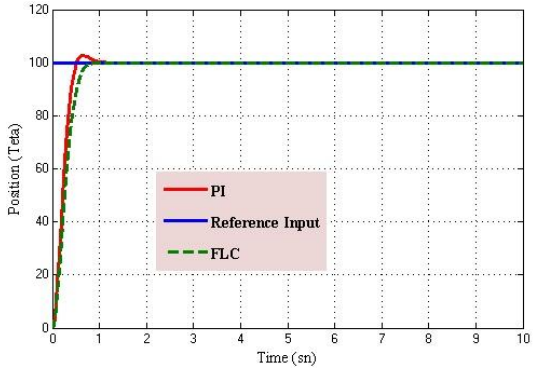


Figure 11: The output of the system for FL and PI controllers

The change of the error in fixed reference input for FL controller is shown in Figure 12. In this figure, it was shown that the error falls to zero quickly.

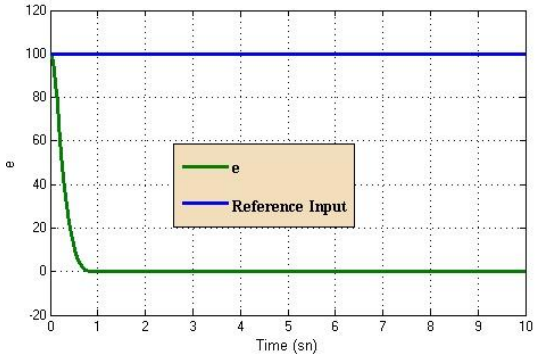


Figure 12: The change of the error in fixed reference input for FL controller

According to variable angular position applied to the system, the system responses from both FL and PI controllers are shown in Figure 13.

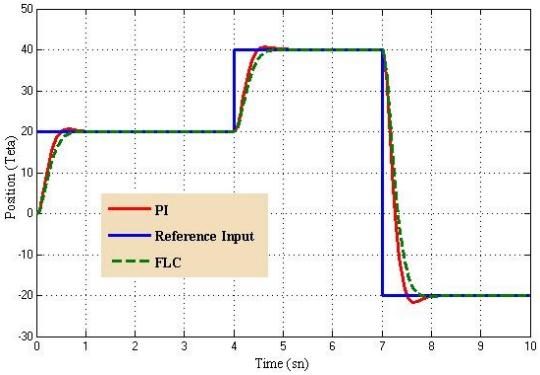


Figure 13: The outputs of the system from both FL and PI controller for variable angular position

Although there are some differences during the transient period of position, both controllers follow variable reference input. The position response from FLC seems to be slower than that of obtained with PI when step changes occur in reference position input level. If a load of 0.5 N is applied to the motor shaft, the system responses from both FL and PI controllers for variable reference input are shown in Figure 14.

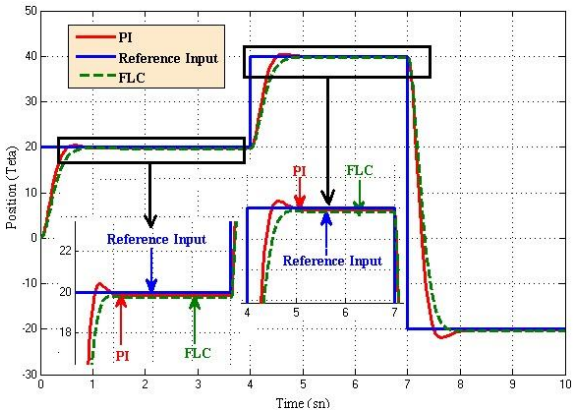


Figure 14: The output of the system from both controllers for a load of 5 N.

It is seen from Figure 14 that when a load is applied to the motor shaft, FLC gives better results than PI controller. Because, if you analyze in Figure 14, PI controller is occurred overshoot more than FLC but PI controller has less steady-state error. As the load applied to the motor shaft increases, the error of the system also increases. The change of the error in variable reference input for FL controller is shown in Figure 15.

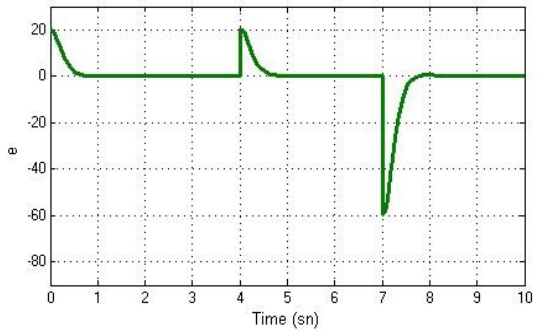


Figure 15: The change of the error in variable reference input for FL controller

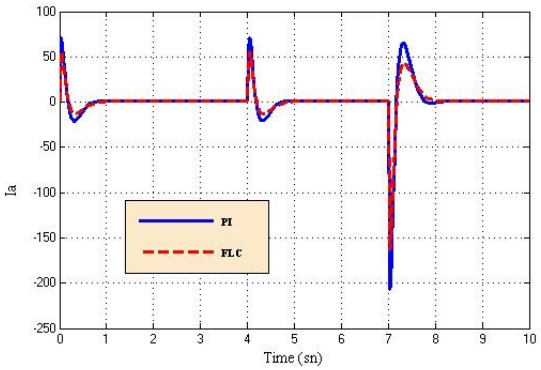


Figure 16: The motor current from both controllers for a load of 5 N.

However, each controller gives different current responses during the transients. As depicted in Figure 16, the current has less ripple magnitudes during the transients with FLC than it has with the PI controller.

V. CONCLUSION

In this paper, a solar tracking system was designed for Matlab/Simulink environment and a control method was proposed for this system. The results from the proposed FLC are compared with those obtained using PI controller. The results from simulations showed good and acceptable performances for the FLC. It was observed that both FLC and PI have almost the same settling time but the FLC show less overshoot. The most important advantage of this proposed control has less ripple magnitudes during the transient. Practically obtained results show that using such controller increases speed of the time response.

REFERENCES

[1] A. Stjepanovic, F. Softic, Z. Bundalo and S. Stjepanovic, "Solar Tracking System and Modelling of PV Module", MIPRO 2010, Opatija, Crotia

[2] J. S. Choi, D. Y. Kim, K. T. Park, C. H. Choi and D. H. Chung, "Design of Fuzzy Controller Based on PC for Solar Tracking System", International Confrerence on Smart Manufacturing Application, April, 9-11, 2008 in KINTEX, Gyeonggi-do, Korea.

[3] H. A. Yousef, "Design and Implementation of a Fuzzy Logic Computer-Controlled Sun Tracking System", Department of Electrical Engineering University of Qatar.

[4] Z. Xinhong, W. Zongxian and Y. Zhengda, "Intelligent Solar Tracking Control System Implemented on a FPGA", Institute of Electrical Engineering, Yuan Ze University.

[5] A. Louchene, A. Benmakhlouf and A. Chaghi, "Solar Tracking System with Fuzzy Reasoning Applied to Crisp Sets", Revue des Energies Renouvelables, Vol. 10, N°2, pp. 231 – 240, 2007.

[6] I. H. Altas and A. M. Sharaf, "A Generalized Direct Approach for Designing Fuzzy Logic Controllers in Matlab/Simulink GUI Environment", Accepted for publication in *International Journal of Information Technology and Intelligent Computing*, Int. J. IT&IC no.4 vol.1.

[7] N. Barsoum, "Implementation of a prototype for a traditional solar tracking system", 2009 Third UKSim European Symposium on Computer Modeling and Simulation.

[8] İ.H. Altaş, "Bulanık Mantık: Bulanık Denetim", Enerji, Elektrik, Elektromekanik-3e, Sayı 64, s:76–81,1999.

[9] İ. H. Altaş, "Bulanık Mantık Denetleyici: Matlab/Simulink Ortamı İçin Bir Modelleme", Otomasyon Dergisi, Bileşim Yayınları, Mart 2007, syf 58-62.