Techno-Science

Scientific Journal of Mehmet Akif Ersoy University

www.dergipark.gov.tr/sjmakeu

Original Research Article

hosted by Turkish JournalPark ACADEMIC

Real-Time Communication between S7-1200 PLC and Matlab/Simulink and a Fuzzy Logic Temperature Humidity Control Application

Yavuz User[1*], Cumali Kara[1]

[1] *Department of Electrical ad Electronics Engineering, Akdeniz University, Antalya, Turkey*

ARTICLE INFO

Article History

Received : 28/11/2018
Revised : 17/12/2018
Accepted : 17/12/2018
Available online : 30/01/2019

Keywords

S7-1200
Real-Time Communication
Complex System
Fuzzy Logic Control
OPC

ABSTRACT

It is not much possible for controllers such as PLC (Programmable Logic Controller) and DCS (Distributed Control System) used in the industry to solve complex systems. On the contrary, Matlab/Simulink provides advantages with the solutions it offers in the complex systems. It is highly significant that PLCs have a real-time communication with Matlab to solve this kind of complex systems. For industrial automation, interoperability standard OPC (OLE for process control) plays the role of a bridge between Matlab/Simulink and PLC and enables data exchange between them. On the other hand, Fuzzy Logic provides advantage in solving complex systems. There are two inputs in this research. Temperature and relative humidity data is transferred to Matlab/Simulink via OPC by PLCs. Based on the temperature and relative humidity felt depending on the temperature, the percentage of fan speed is calculated by using Fuzzy controller in Matlab/Simulink and transferred to PLC via OPC and the required physical output connected to PLC is activated. This system shows how fast the processes which are complex in PLC can be solved with Matlab/Simulink. A complex system developed in Matlab/ Simulink can be automatically converted to any microcontroller code. The PLC used in industry can communicate with this microcontroller as well and obtain the desired output in the best way possible. Nowadays, Artificial Intelligence develops day by day and such a system gains even more importance.

1. INTRODUCTION

Generally, industrial controllers that can perform simple operations in a factory environment are used. However, these controllers cannot evaluate complex processes at the desired performance. For this reason, complex processes can be calculated by using a PC and the result can be transferred to the controller. Complex processes can be calculated in Matlab through an artificial intelligence algorithm.

MATLAB is a programming language that is capable of performing multi-paradigm numerical computation. And, it is a powerful tool that enables the development of operations such as process control, fuzzy logic, artificial neural networks, image processing, genetic algorithms in their own environment. Also, it is easier to prepare and run programs with Matlab/ Simulink solution.

PLC (Programmable Logic Controller) is a processor unit which is generally used in the industry for process controls. It is in great demand in the industry as it provides easy programming possibility, the flexibility it provides in including into the process, and it can operate under severe conditions. The Siemens product S7-1200 used in this study can be programmed with the TIA PORTAL editor via its Ethernet port.

Before OPC, producers offered special software and drivers to enable different products communicate with each other, and as a result, data exchange was ensured by complying with specific protocols.

* Corresponding Author: yuser@akdeniz.edu.tr / Tel: +905053762675

To cite this article: User, Y., Kara, C. (2019). Real-Time Communication between S7-1200 PLC and Matlab/Simulink and a Fuzzy Logic Temperature Humidity Control Application. *Techno-Science* vol. 2, no. 1 p. 7-14

e-ISSN: 2651-3722

The wide product range and the protocols varying from brand to brand made it necessary to proceed with a specific standard in this respect.
OPC technology, which is based on server-client architecture, was developed to enable seamless interoperability of different products with a single standard [1].

Figure 1 shows the structure where the systems using different communication protocols offered product-specific driver before the OPC.

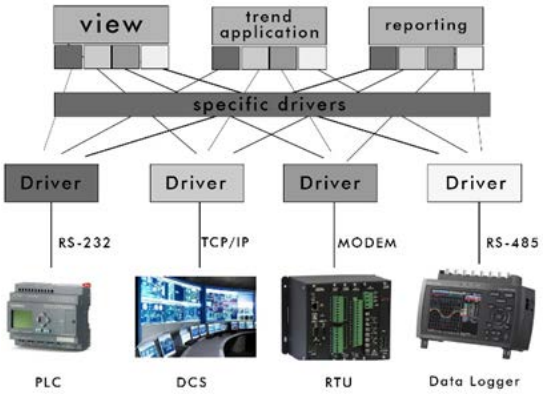

Fig. 1. *Connection diagram before OPC*

Different devices included in the system with different software applications enable communication with a single standard as long as they have the OPC driver. The communication structure after OPC is shown in Figure 2 [3] [7].

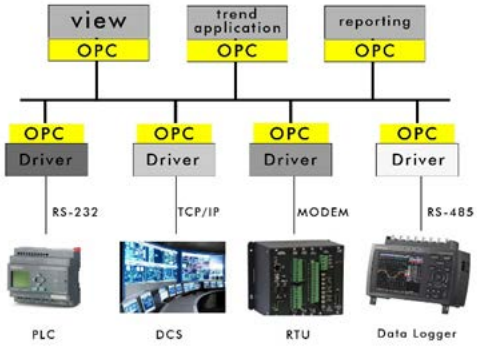

Fig. 2. *Connection diagram after OPC*

2. METHOD

2.1. Communication Between Matlab/Simulink and S7-1200 PLC Over OPC

In the OPC structure where the server and the client reside, the server carries out data transfer in line with the requests of the client. OPC client is supposed to be used on the application side to communicate with an appropriate OPC server. Also, it functions as the data receiver as clients control the communication with the servers. Figure 3 shows the structure of OPC Server.

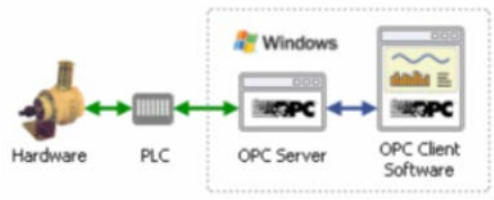

Fig. 3. *OPC Server Structure*

OPC DA (OPC Data Access) standard was established for the interaction between server and client. OPC Toolbox in Matlab is an OPC DA client application. With this application, it is possible to connect to a server that is in compliance with OPC DA

from Matlab environment and to read/write data through the server. The connection between Matlab/Simulink, PLC and OPC is shown in Figure 4.

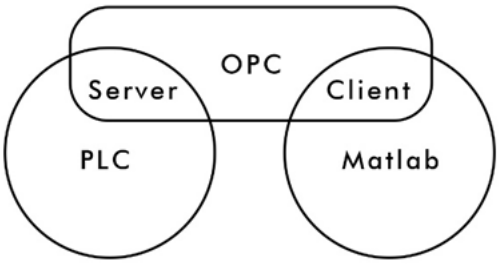

Fig. 4. *Matlab, PLC and OPC connection structure*

In Matlab / Simulink, first of all, OPC Server object is created within the OPC client program and information about the OPCItems stored on the server is obtained. Considering the information obtained, OPCGroup objects are created for OPCItems and then they are grouped. This enables the configuration of group, item and all client objects via the OPC Toolbox.

2.2. Fuzzy Logic

Classical logic is based on strict judgments. According to 0 and 1 logic, whether an object belongs to a cluster is specified with definitive expressions. In Fuzzy Logic, there is no definitive expression like 0 and 1, and instead, there are flexible expressions that we use in our daily life.

According to the classical logic, the weather is either hot or cold. In Fuzzy Logic, it is possible to have classifications such as the weather is "slightly warm", "warm", "very hot", "somewhat cold", and this gives flexibility to the fuzzy logic. The fuzzy logic stages are shown in Figure 5.

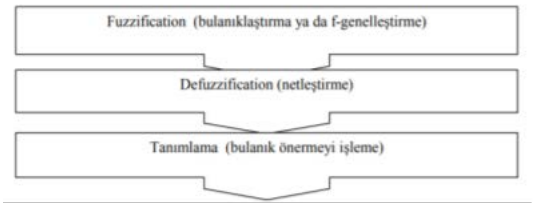

Fig. 5. *Fuzzy Logic stages*

The first and second phases refer to cluster formation and identification processes. Fuzzification is the generalization of fuzzy clusters, and at this stage, we get the correct degree of the line in question. The membership functions stored at this stage are used to obtain fuzzy input values. The information is converted into fuzzy sets and defuzzification refers to a process that transforms a fuzzy number into an integer [2].

The Methods section details the theoretical or experimental methods used. What justifies using a given method? What is special, unexpected, or different in your approach? If you use a standard or usual procedure, mention that upfront, too.

3. CREATION OF THE SYSTEM

It is ensured that the server is ready with the channel setting, system settings, PLC model, Ethernet address and tag entries to be used for OPC server configuration. The addresses and data types of the data to be transferred from Matlab to PLC and from PLC to Matlab are shown in Figure-6.

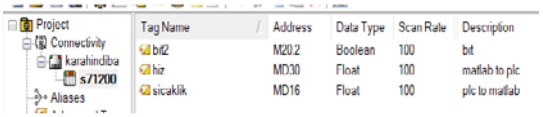

Fig. 6. *The data created in the OPC*

S7-1200 was programmed via Tia Portal editor. Analog temperature information was scaled between 0-100 (Figure-7). The scaled 'temperature' data was transferred to Matlab via OPC.

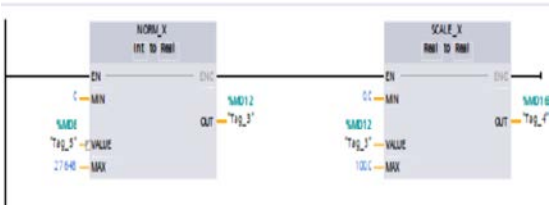

Fig. 7. *Scaled analog value*

The temperature and relative humidity information obtained from DHT22 connected to PLC were inserted into the fuzzy logic block in Matlab / Simulink through PLC, and the resulting fan speed analog value were transferred from Matlab to the PLC as a percentage. For the physical representation of the results, it was ensured that the physical outputs were activated/deactivated by using the speed information comparison commands from Matlab in TiaPortal as shown Figure 8.

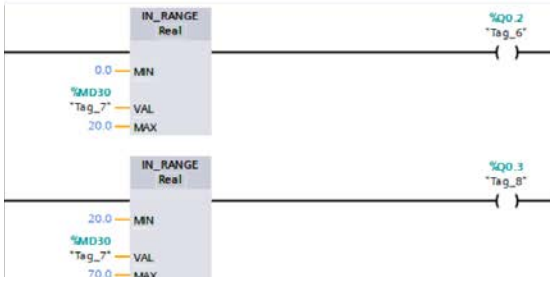

Fig. 8. *Activated physical output*

Fuzzy logic clusters were created by considering the apparent temperature table depending on the temperature and relative humidity as shown Figure 9 [4].

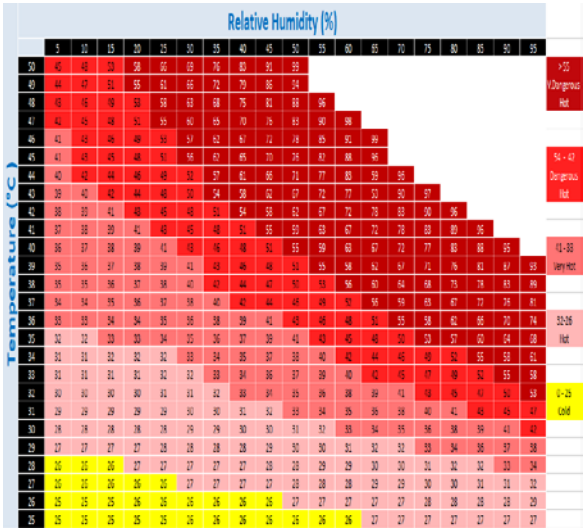

Fig. 9. *Apparent temperature table*

Figure 10 shows the parameters of the fuzzy model consisting of two input and one output parameters, and the system's modeling in MATLAB program.

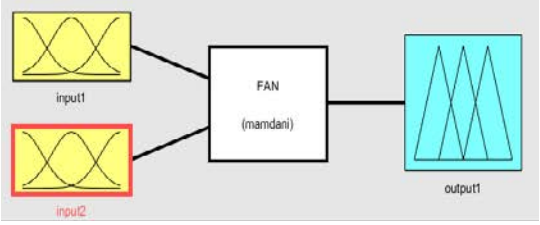

Fig. 10. *(input1=temperature , input2=humidity, output=speed)*

The graphical representation of the Input 1 "temperature" parameter in Matlab environment is given in Figure 11.

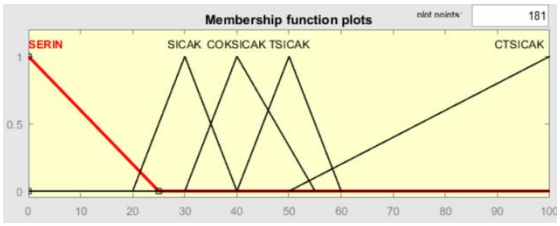

Fig. 11. *Temperature parameter*

The graphical representation of the Input 2 "humidity" parameter in Matlab environment is given in Figure 12.

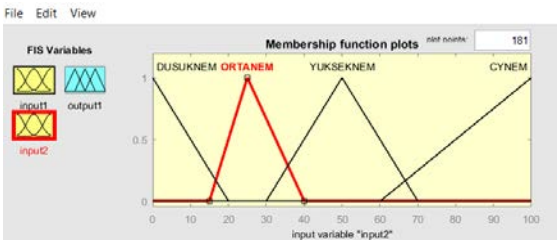

Fig. 12. *Humidity parameter*

The graphical representation of fan speed variable in Matlab environment is given in Figure 13.

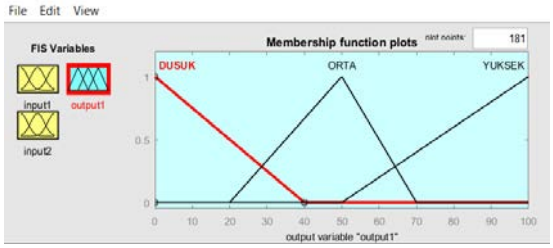

Fig. 13. *Fan parameter*

Rules were set to understand the relation between input and output parameters. The resulting database contains all the possibilities and some rules of the fuzzy model are shown in Figure 14.

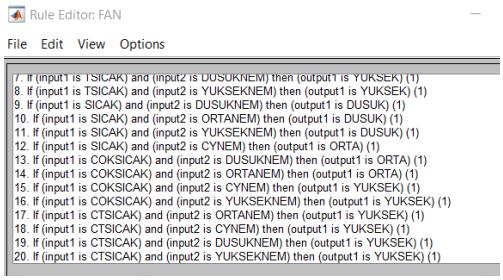

Fig. 14. Fuzzy Model rule

The impact of input variables on the output variables are given in Figure 15 based on the fuzzy rule base.

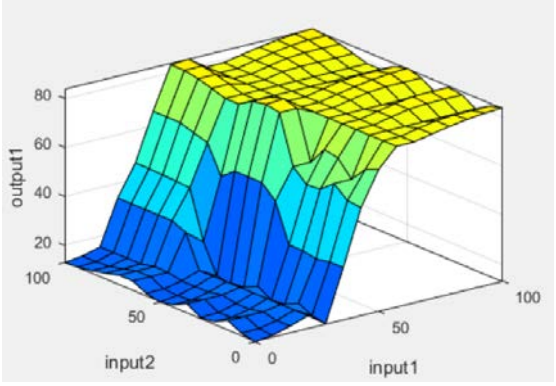

Fig. 15. *The impact of input variable on output variable*

Matlab/Simulink Block diagram is shown in Figure 16. Block descriptions are given in the following items and Figure 17 shows Matlab/Simulink OPC Configuration [5, 6].

1. In section 1, Matlab -OPC settings are made.
2. In section 2, address settings that will be read from PLC through OPC are performed.
3. The fan speed value resulting from fuzzy logic operations of the values entered in section 3 is transferred to the address associated with PLC from Matlab via OPC.
4. In section 4, the relative humidity and temperature information obtained from PLC are actively read via DHT22 with Matlab.

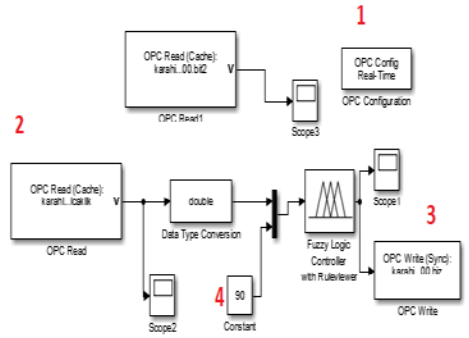

Fig. 16. Matlab/Simulink Block Diagram

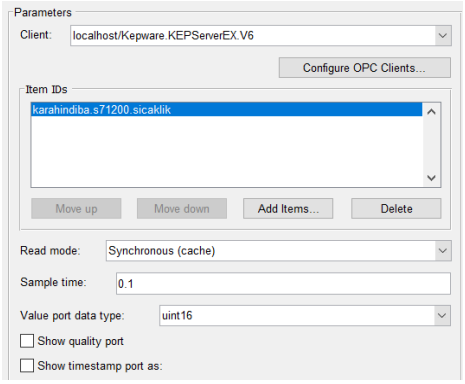

Fig. 17. Matlab/Simulink OPC Configuration

3.1. System Operation

The system gives the fan speed as percentage after the Fuzzy process on analog temperature value from PLC and relative humidity ratio entered through Matlab, and sends it to PLC. The real-time communication and control of the system are shown below. TIA PORTAL part of the system input is shown in Figure 18. OPC part of the system input is given in Figure 19.

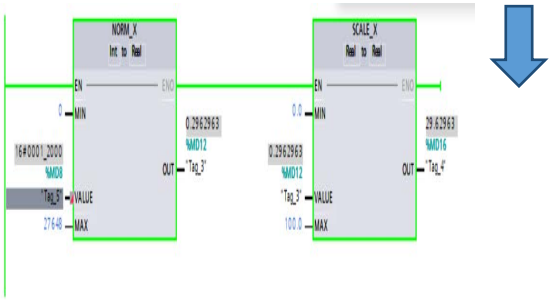

Fig. 18. System input in TIA PORTAL

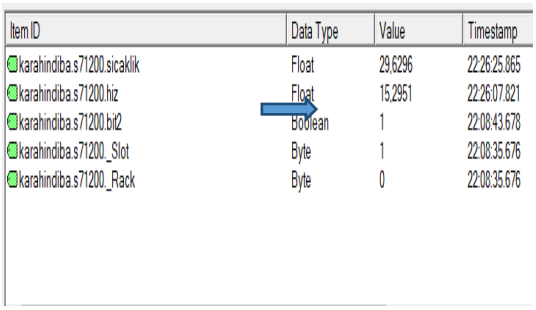

Fig.19. OPC View of system input

System input is given in Figure 20 and system output is given in Figure 21.

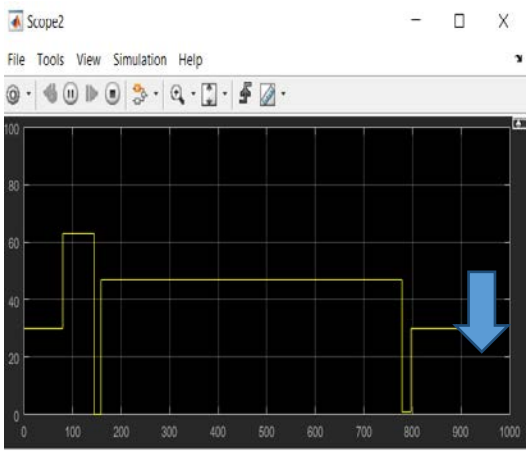

Fig. 20. System input MATLAB view

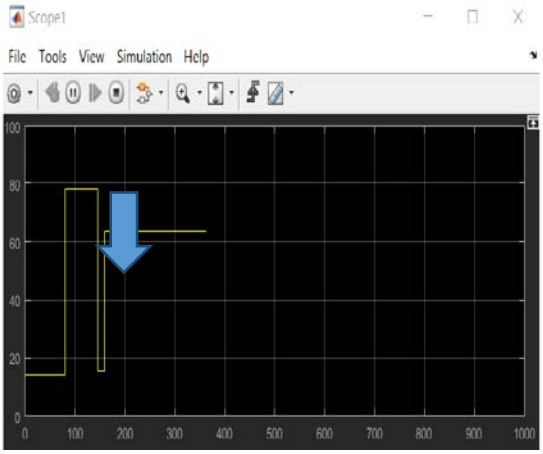

Fig. 21. MATLAB output view

OPC part of the system output is given in Figure 22. TIA PORTAL portion of the system output is given in Figure 23. Snapshot of the system's Fuzzy input/output is given in Figure 24.

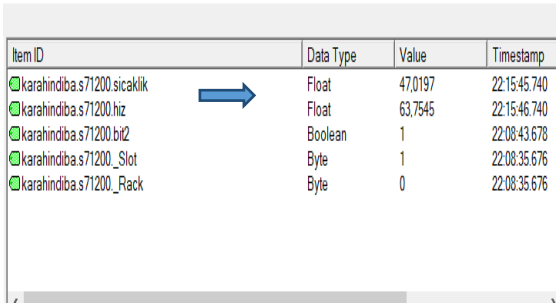

Fig. 22. *OPC View of system output*

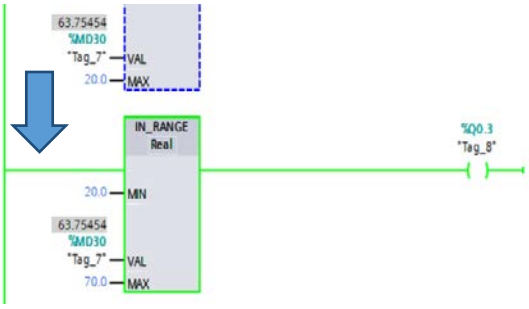

Fig. 23. *System output TIA PORTAL*

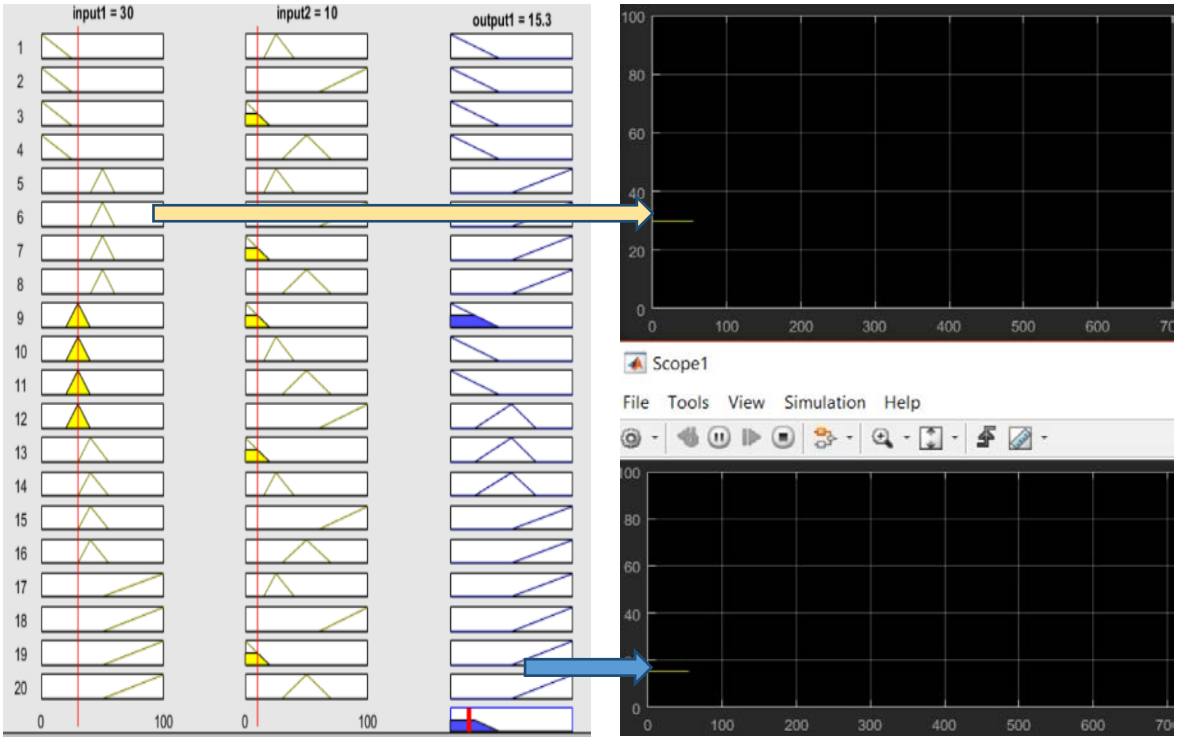

Fig. 24. *Snapshot of the inputs/outputs in Fuzzy controller*

4. CONCLUSIONS

Real-time data exchange between S7-1200 and Matlab / Simulink was ensured through OPC Server. System inputs were processed in in Fuzzy controller in Matlab / Sımulınk and then system output is obtained. The resulting system output was transferred to S7-1200 and the desired physical output is set. This study shows that a complex system is resolved in an equally strong platform in real time.

REFERENCES

[1]. Zhang Lieping, Zeng Aiqun, Zhang Yunsheng, (2007). On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology, *Chinese Control Conference*, 26-31 July , China.
[2]. Sevil, M., Elalmış, N., Görgün, H., Aydın, N. (2015). "Control of Air Conditioning with Fuzzy Logic Controller .Design for Smart Home Systems", *Sigma Journal of Engineering and Natural Sciences*.
[3]. Ünlü Y., (2007)"SÜREÇ KONTROLUNDA NESNELERİN BAĞLAŞMASI VE İLİŞKİLENDİRİLMESİ (OPC). STANDARDI VE UYGULAMASI", *Master Science Thesis-Istanbul University.*
[4]. The General Directorate Meteorology *www.mgm.gov.tr/genel/sss.aspx?s=hissedilensicaklik.*
[5]. The Matworks, (2008). *MATLAB OPC Toolbox User's Guide*.
[6]. S7-1200 Programmable controller, (2017). *System Manual*.
[7]. Tekinalp Z., Öztürk S., Kuncan M., (2013). "OPC Kullanarak Gerçek Zamanlı Haberleşen Matlab ve PLC Kontrollü Sistem

Techno-Science *Paper ID: 488741*