UNEC Journal of Engineering and Applied Sciences
Vol. 4, No.2, 2024, pp. 76-90
DOI: https://doi.org/10.61640/ujeas.2024.1208

SECTION: ENGINEERING SCIENCES

An experimental fuzzy logic application of position control using ST52 microcontroller

Ö.F. Özgüven, M.S. Mamiş*, A. Memmedov

Engineering Faculty, Department of Electrical and Electronics Engineering, İnönü University
44280-Malatya, Türkiye
email: *mehmet.mamis@inonu.edu.tr*

Abstract

Position control is required in many industrial applications. In this paper, several control methods; namely servo, fuzzy logic and neuro-fuzzy position controls implemented in ST52 microcontroller and laboratory test results of these methods are introduced. Fuzzystudio3.0 and Adaptive Fuzzy Modeller software packages have been used for the program development. The response of designed control systems to any desired position changes is measured, and the performance and speed for fuzzy, neuro-fuzzy and servo controllers in setting a desired position are examined. The designed software and installation steps are described and structure and listings of the control programs are presented.

***Keywords*:** position control, ST52 microcontroller, servo control, fuzzy control, neuro-fuzzy control.
PACS numbers: 02.60.−x; 02.70.−c

Received:	*Revised:*	*Accepted:*	*Published:*
31 May 2024	*10 December 2024*	*18 December 2024*	*26 December 2024*

1. Introduction

In many applications of industrial automatic control systems it is necessary to attain a controllable movement in accordance with the reference input, and it is desired output unit to maintain at the set position. The control systems may be electrical, mechanical, electro-mechanical, electro- pneumatic and electro-hydraulic etc. Position control is required in many industrial applications. Several control methods with complicated hardware have been used for position control. Recently, due to improvements in the speed and facilities in the hardware and software, microprocessors and microcontrollers have been most commonly used in the control systems [1 -4], and due to significant developments in this area, applications of microprocessors and microcontrollers have gained more importance in control engineering.

The studies carried out in previous years are listed below.

Authors and ref. num.	The content of the study carried out in the article
Muscato [5]	Using the ST52 microcontroller, the position control of a two-rotary- joint arm capable of moving in a single plane was achieved with two separate Fuzzy Logic Controllers (FLCs). A total of 72+3=75 rules were applied across both controllers. In the first FLC, 13 and 10 membership functions were used for two separate input variables, respectively, while the second FLC employed 3 membership functions for its input variable. In the control system used, the first actuator was connected to a single actuator, while the second was moved without any actuator. A 160:1 reduction gear was chosen for the arm joint.

Lu [6]	They performed the control of a DC servo motor using an FLC. A Power-14 system board containing the TMS320C14 digital signal processor was utilized. In the FLC, there are 7 different membership functions for the input-output variables error (*e*), change in error (*ce*), and control signal (*u*). The number of rules is 7x7=49.
Esmail et al. [7]	In their simulation studies, they achieved the position control of a DC motor by using PID and FLC individually, as well as by cascading PID and FLC.
Sreenatha et al. [8]	They achieved the position control of a flexible structure driven by a DC motor using an FLC. Their application employed a PC and a PC18L-PC interface card. The FLC algorithm was implemented as a software program on the computer. Seven membership functions and 49 rules were used for the variables.
Khongkoom et al. [9]	They performed the position control of a DC servo motor using an FLC. The FLC algorithm was programmed on a computer, and the control process was executed with an A/D converter and a driver for the DC motor. The motor position was determined using an encoder and a counter connected to the motor shaft.
Sapiński [10]	They controlled the reduction of oscillations in an MR damper located in the driver's seat using the ST52E420 microcontroller. Seat velocity and the relative velocity of the seat concerning the ground were used as inputs to the FLC. The current applied to the coil affecting the seat velocity was used as the fuzzy rule output variable.
Mahdi et al. [11]	They conducted a simulation in MATLAB Simulink to control the position of DC linear actuator motors for a car's throttle and brake pedals using a Fuzzy Model Reference Adaptive Control (Fuzzy MRAC). Position error and the change in position error were used as the FLC rule inputs, while the position of the DC motor was the output variable.
Hanafi et al. [12]	They performed the online position control of a DC motor using an Incremental Fuzzy Controller (IFLC). For this application, they used a computer and a RAPCON board. Simulations and application studies were conducted for error, change in error, and varying coefficients at the FLC output.
Patjoshi et al. [13]	They achieved the position control of an elevator-door system driven by a DC motor. This system was designed with a fuzzy logic-based intelligent control system using the MC9S12DP256B-NXP microcontroller. The FLC employed 5 membership functions for error and change in error, with a total of 25 rules.
Franc et al. [14]	An LPC1768 microcontroller was used, along with Keil uVision4 programming software for C programming and code compilation.

In recent years, a number of fuzzy processors, which perform fuzzyfication and defuzzyfication processes by the hardware they included, have been developed. To build up the whole control system by these processors however, a microprocessor is needed in addition. Last decades, microcontrollers with a fuzzy core have been used used, which facilitate the design procedure considerably. Microcontroller ST52 [15], developed by SGS-THOMPSON Inc., is an example for this type of the processors. Microcontroller with fuzzy processor makes the fuzzy control process easy and offers a high speed and as well as a modular control system design. Further, it is cheep and proper in physical size. Last decades also microcontrollers involving flash memory and many peripheral units have also been developed. Programming tools called Fuzzystudio3.0 and Adaptive Fuzzy Modeller are also available by ST-Microelectronics for ST52 microcontroller. With Fuzzystudio3.0 software, the program algorithm can be produced in the form of flow diagrams, which can be converted into machine language automatically. Further, it is also possible with this software package to use the program icons for conditioning special function registers (SFR). The conditions and flow diagrams are then compiled and converted into operational codes (OP- CODES). ST52 includes an EPROM inside which can be used to accumulate the generated codes. Another specification of ST52 microcontroller is that it encloses a fuzzy core, which offer many facilities for the software and hardware of the control process in addition to on board parallel and serial ports and analog to digital (A/D) converters.

The Adaptive Fuzzy Modeller is a program that can be used to produce suitable fuzzy rules and membership functions for the system using the neural networks. This tool easily allows obtaining a system model based on Fuzzy Logic data structure, starting from the sampling of a process/ function expressed in terms of input-output values pairs (patterns).

In this paper, practical implementations of servo, fuzzy logic and neuro-fuzzy position control with ST52 microcontroller are described for industrial applications. Developed algorithms are implemented using Fuzzystudio3.0 and Adaptive Fuzzy Modeller software packages. The response of the designed control system for the input position changes is measured. The performance and speed of basic servo, fuzzy, and neuro-fuzzy controllers are examined experimentally in setting a position of potentiometer at output of control system to a desired point specified by the input. Servo controls, fuzzy logic and neuro-fuzzy controls are compared in performance and speed points of view.

2. Result and discussion

2.1. Basic Operations in Fuzzy Sets

Intersection $x \wedge y = min\,(x, y)$
Algebraic product $x \cdot y = xy$ (1)
Union $x \vee y = \max(x, y)$
Algebraic sum. $x \dotplus y = x + y - x\,y$ (2)

Assuming x and y are fuzzy subsets in the universe of discourse (U), the equations for norms and co-norms can be written in equation 1 and 2, respectively [16].

2.2. Fuzzy Logic Controller

Figure 1 shows the block diagram of the fuzzy control system. The inputs to the controller are error- e (t) and the change of error-ce (t). Based on these two values, fuzzification is performed by applying them to the membership functions. Using the obtained values, inference is carried out using all the membership functions and rules defined for the system. Finally, defuzzification is performed using the combination, maximum, and centroid values of the output membership functions. The resulting output value is then applied to the input of the system.

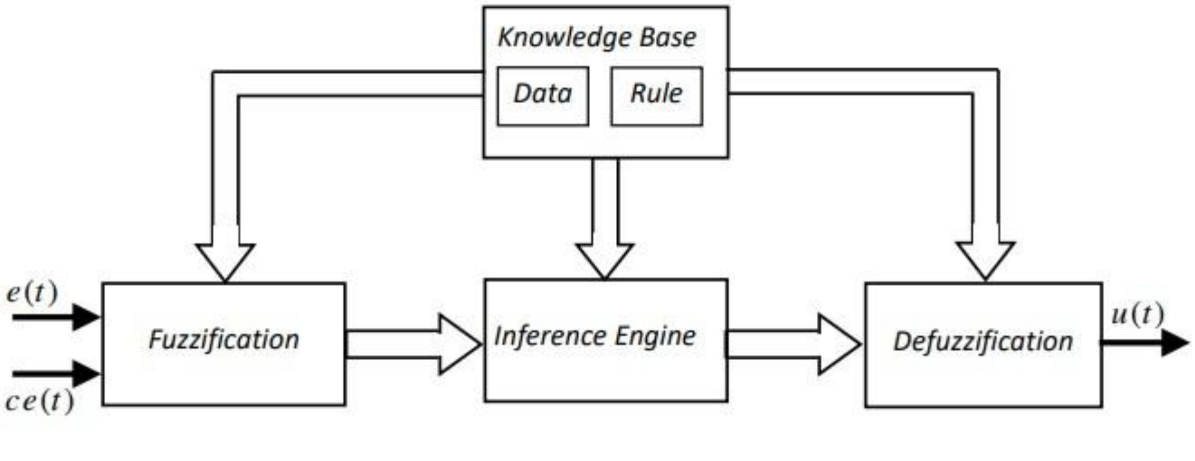


Figure 1. Block diagram of fuzzy logic controller (FLC)

In equation 3, a rule set containing the relationship R_1-R_i for *i* rules is shown. *x* and *y* are system inputs, and *z* is the FLC system output. A_1-A_i and B_1-B_i represent the linguistic labels of the membership functions for the inputs *x and y*, respectively, while C_1-C_i represent the linguistic labels of the membership functions for the output *z*.

$$\begin{array}{c} R_1\text{: if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z{=}C_1, \\ R_2\text{: if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z{=}C_2, \\ \ldots.\quad \ldots.\quad \ldots.\quad \ldots.\quad \ldots. \\ \ldots.\quad \ldots.\quad \ldots.\quad \ldots.\quad \ldots. \\ R_n\text{: if } x \text{ is } A_n \text{ and } y \text{ is } B_n \text{ then } z{=}C_n, \end{array} \tag{3}$$

For inference operations, each rule in equation 3 can be written in the form of a relation expression as in equation 4a and 4b shows the equality written according to the weights of the sets in equation 4a [17,18].

$$R_i \triangleq (A_i \ and\ B_i) \rightarrow C_i \tag{4a}$$

$$\mu_{R_i} \triangleq (\mu_{A_i}(u) and\ \mu_{B_i}(v)) \rightarrow \mu_{C_i}(w) \tag{4b}$$

In equation 5, when the A' and B' values are applied to the *x* and *y* rule inputs in equation 3, the output expression is given. In this expression, it is stated that the composition operation with the $A'$and B' values and the union of all the rules is equal to the composition operation separately with each rule. Here, “o” represents the composition, and ∪ denotes the union (co-norm) operation.

$$\begin{gathered} C_i' = (A', B')\text{o}\bigcup_{i=1}^{n} R_i = \bigcup_{i=1}^{n}(A', B')\text{o}(A_i \ and\ B_i) \rightarrow C_i \\ C_i' = (A', B')\text{o}(A_i \ and\ B_i \rightarrow C_i) \\ C' = \bigcup_{i=1}^{n} C_i' \end{gathered} \tag{5}$$

In equation 6, A' and B' give the C_i' output expression when applied to the inputs of the i-th rule.

When the composition operation is applied to the known inputs $A^{'}$ and $B^{'}$ for the i-th rule, this expression shows that it is equal to the minimum of the composition of the i-th rule, which includes the inputs and outputs written separately for $A^{'}$ and $B^{'}$ within their universe of discourse.

$$\begin{gathered} (A', B')\text{o}(A_i \ and\ B_i \rightarrow C_i) = [A'(A_i \rightarrow C_i)] \cap [B'\text{o}(B_i \rightarrow C_i)] \\ \text{if } \mu_{A_i} \times \mu_{B_i} = \mu_{A_i} \wedge \mu_{B_i} \end{gathered} \tag{6}$$

In equation 7, when u_0 and v_0 is applied to the x and y inputs in equation 2, respectively, it represents the weights for the i. Rule

$$A' \triangleq \mu A_i(u)|_{u=u_0}, B' \triangleq \mu B_i(v)|_{v=v_0} \tag{7}$$

In equation 6, when the expressions A' and B' are replaced with $\mu_{A_i}(u_0)$ and $\mu_{B_i}(v_0)$ respectively, equation 8 is obtained.

$$\mu_{C_i'} = \min\left[\mu_{A_i}(u_0)\text{o}\left(\mu_{A_i}(u) \rightarrow \mu_{C_i}(w)\right)\right], [(\mu_{B_i}(v_0)\text{o}\left(\mu_{B_i}(v) \rightarrow \mu_{C_i}(w)\right)] \tag{8}$$

The fuzzification process is shown in equation 9.

$$\begin{gathered}\mu_{A_i}(u_0) = \mu_{A_i}(u_0)\text{o}\mu_{A_i}(u) \\ \mu_{B_i}(v_0) = \mu_{B_i}(v_0)\text{o}\mu_{B_i}(v)\end{gathered} \tag{9}$$

If equation 9 is substituted into equation 8, equation 10 is obtained.

$$\mu_{C_i'} = \min[\mu_{A_i}(u_0) \rightarrow \mu_{C_i}(w)], [(\mu_{B_i}(v_0) \rightarrow \mu_{C_i}(w)] \tag{10}$$

$$C' = \cup_{i=1}^{n} C_i' \tag{11}$$

Therefore, the defuzzification output can be written as in equation 12.

$$\mu_{C'} = \cup_{i=1}^{n}\{\min\left[\mu_{A_i}(u_0) \rightarrow \mu_{C_i}(w)\right], [\left(\mu_{B_i}(v_0) \rightarrow \mu_{C_i}(w)\right]\} \tag{12}$$

2.3. The center of Area Method (COA)

There are various methods available to obtain a singelton value from the defuzzification result. The COA method is commonly used to find the singelton value. Expression of COA method is given in equation 12.

$$z_0 = \frac{\sum_{j=1}^{n} \mu_z(w_j)\cdot w_j}{\sum_{j=1}^{n} \mu_z(w_j)} \tag{13}$$

Here, $\mu_z(w_j)$ represents the maximum value of the membership functions determined by the inference result, and w_j represents the center of the corresponding membership function in the universe of discourse.

2.4. Fuzzy core unit of the microcontroller

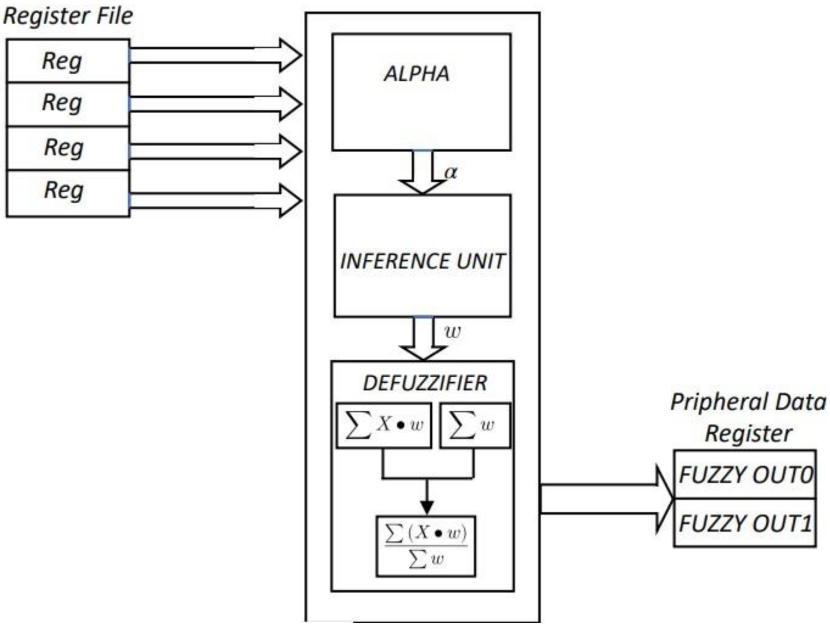


Figure 2. Fuzzy core block diagram

In figure 2, the block diagram of the fuzzy logic controller available in the ST52 microcontroller hardware is shown [15].

This unit performs the operations in figure 1 and equations 3-13 quickly and easily using the fuzzy commands available in the processor. Reg0, Reg1, Reg2, Reg3 ve FUZZY OUT0, FUZZY OUT1 respectively represent the rule inputs and the rule outputs with a maximum of two values. The Alpha Calculator corresponds to the fuzzification block in figure 1. In the defuzzification process, it uses the COA method.

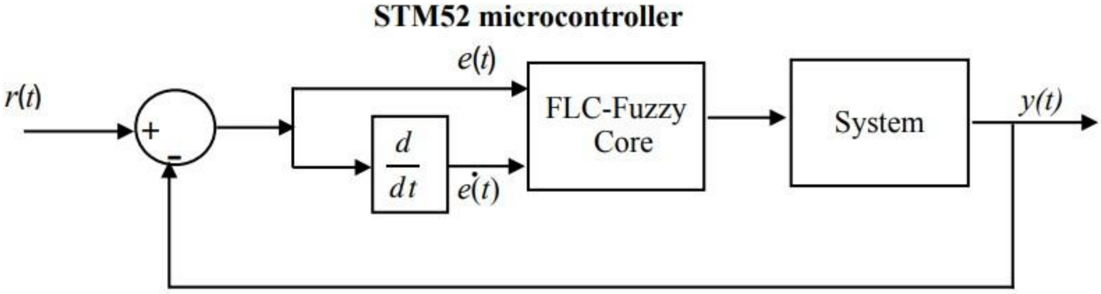


Figure 3. Block diagram of control systems

In figure 3, the block diagram of a feedback control system containing a Fuzzy Logic Controller (FLC) is presented. The operations within the FLC block are performed in the microcontroller's fuzzy core unit. In the block diagram, there are reference input- $r(t)$ and output- $y(t)$. The output determines the required positional value. In other words, the output follows the input through the feedback control system. In this article, $r(t)$ represents the angular position of the potentiometer at the input, denoted by $\theta(t)$, and the output response $y(t)$ is determined by the controller based on the input position, or in other words, the potentiometer's angle.

2.5. Position control using servo controller

The configuration of the servo controller used in the experimental setup is given in figure 4. In this figure V is the supply voltage, θ_1 is the angle of reference position which is set at the input and θ_2 is the angle of actual position at the output. Potentiometers P_1 and P_2 are transducers that convert the level of rotational movement into a change of voltage. u_1 and u_2 in this figure represent reference voltage corresponding to θ_1 and output voltage corresponding to θ_2, respectively,

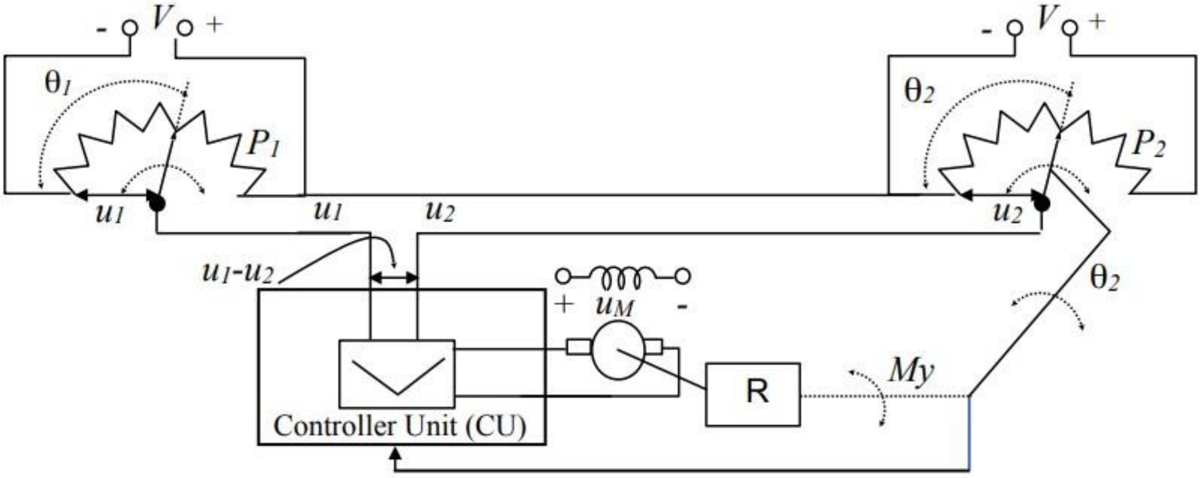


Figure 4. Schematic diagram of servo controller

and are measured in order to find out the desired position and the actual positions and to evaluate the error and the change in error. Position control with fuzzy and neuro-fuzzy controller Fuzzy logic is similar to the humankind thinking method, using the fuzzy reasoning and logic to think and get control signal [16-18]. Fuzzy logic control can play an important role because knowledge based design rules can easily be implemented in systems with unknown structure. Therefore, much research has been done on application of fuzzy logic to the control systems [1, 3, 8, 9, 19]. Although fuzzy logic can encode expert knowledge directly using rules with linguistic labels, it usually takes a lot of time design and tune the membership function that quantitatively represent these linguistic labels, and applications of pure fuzzy control systems are restricted mainly to those fields where expert knowledge is available and the number of input variables is small. Neural network learning techniques can automate this process and substantially reduce development time and cost while improving performance [20]. Neural network are also used to pre-process data and to extract fuzzy control rules from numerical data automatically, as well as tune membership functions of fuzzy systems [21-26]. Block diagram for the fuzzy logic processing automatic control system with ST52 microcontroller that can be also used for neuro-fuzzy control is shown in figure 5. Potentiometers are used at the input and output as transducers for position angle-voltage transformation as in the servo controller. The error e and the change in error ce are the state variables for the control system, where e is the difference between input and output position and ce is the difference between the error in subsequent time steps. The state variables are processed in the fuzzy logic unit and a digital output is produced depending on the sign and the level of error. Then, the motor direction is specified and the motor is driven in either forward or in the reverse direction regarding the sign of error. Analog voltage produced by D/A converter is applied to a driver (amplifier) which converts and amplifies the produced analog signal into an appropriate voltage level to drive the motor. A speed reductor (block R) is located between the motor and output potentiometer to be controlled.

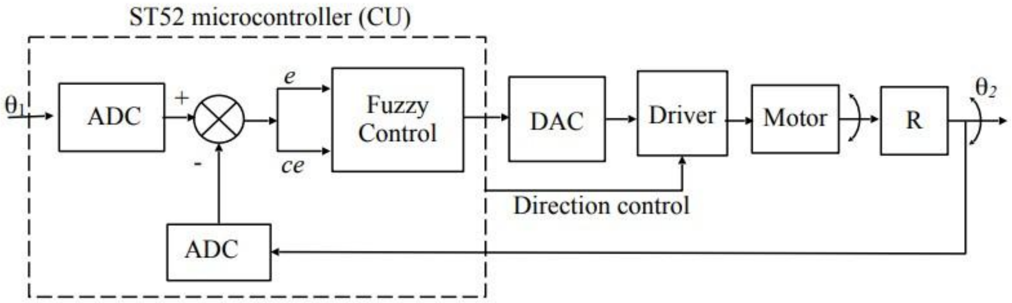


Figure 5. Block diagram of the fuzzy controller

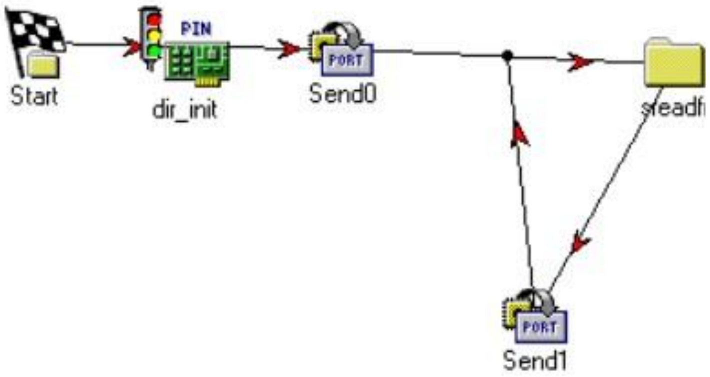
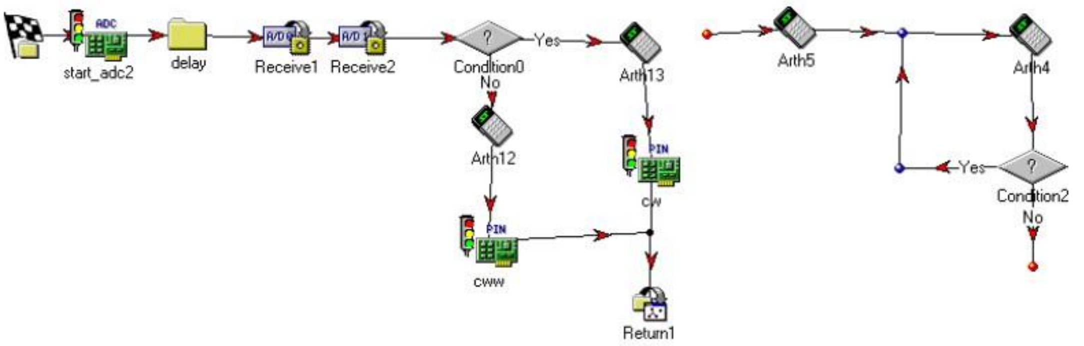


Figure 6. Structure of main program

Structure of control program for servo controller Structure of the control program for the servo controller is shown in figure 6. Listing of the main program and folders for the subprograms are given in Appendix I. As it can be seen from the figure, the control program is very simple. For the control process, the voltages on the potentiometers at input and output are measured and the measured data for u_1 and u_2 is applied to A/D converter to establish the error and the direction of the motor. The data received from the A/D converter is processed in the folder sreadfr to specify the direction and the level of motor drive. Structure of this folder is shown in figure 7a. A time delay is required for conversion in ADC, which is done in the folder delay shown in figure 7b. The calculated error is sent to D/A converter and an analog signal is generated. The generated output is applied to the motor through a driver. Depending on the sign of the error ($e = u_1 - u_2$), the motor is driven either in forward or in the reverse direction. If the actual position is equal to the desired location, the error and hence the voltage applied to the motor would be zero and the motor would stop. These operations repeat in a closed loop continuously.



a) b)

Figure 7. Structure of folders; a) sreadfr, b) delay

Structure of the control program for fuzzy and neuro-fuzzy controller Control algorithm for the fuzzy controller is based on the fuzzy set theory [16-18]. Structure of the control program developed for the fuzzy controller is shown in figure 8 and the program listings for fuzzy logic are given in Appendix-II. The program starts by initialization of the control system by loading the input of D/A converter by the block send0, which feeds the motor with a controlled voltage produced by the microcontroller. The control process then continues by analog to digital conversion of the input and feedback output signals by A/D converter in ST52. Using this data, the error e is determined as a difference between the input and output voltages to specify the motor desired drive level. At the same time, the error change ce is determined to specify the motor direction. The data of the desired position loaded by the A/D converter and the data of the actual position are stored for initialization. Folder adcread is developed to perform this task. Fuzzystudio3 implementation of this folder is shown in figure 9a. Folder delay shown in figure 9b is used to provide a time delay required for conversion in ADC. It should be noted that there is a difficulty in accumulation of the real variables; because real variables e and ce can vary within the range -2.5 V and 2.5 V but the whole range of input variables need to be coded within the range [0, 255]. For this reason -2.5 V is coded as binary -255 and 2.5 V is coded as binary 255 in the program. For this purpose an adaptation procedure that transfers -255 to 0, 0 to 127 and 255 to 255 is used, which is expressed by the following equations:

$$e_n = 127 + \frac{e}{2} \tag{14}$$

$$ce_n = 127 + \frac{ce}{2} \qquad (15)$$

Another difficulty in the implementation is the microcontroller have no command to divide by two, which is required to determine en and cen in equations (14) and (15). This problem overcomes by the logical shift right operation. Folder divby2a shown in figure 9c is built to divide the input received from A/D converter using the shift right operation, and then 127 is added to the generated result. Next, input data is transferred into the fuzzy block after the adaptation process described above. Then, error ei and cei=ei - ei-1 in the step i is calculated in binary form. These operations are carried out in a closed loop at every cycle of operation by the folder divby2b. Structure of this folder is the same as that of divby2a. This folder also receives input and output data at every cycle by means of routine freadfr. Structure of this folder is illustrated in figure 9d. Folder adcread1 is used to read digital output at ADC at every cycle of operation. Structure of this folder is shown in figure 9e. To set the desired position, direction of the motor is determined first. If the reference input is greater than the output, the motor is driven in the forward direction by an amount proportional to the error. The motor is driven in the reverse direction and on the contrary. Produced output as a result of this operation is sent to D/A converter by block send1and the generated analog output at this unit is fed to the amplifier. Whenever the desired position is located at the output, the error e becomes zero and the motor stops. The control mechanism repeats in a closed loop.

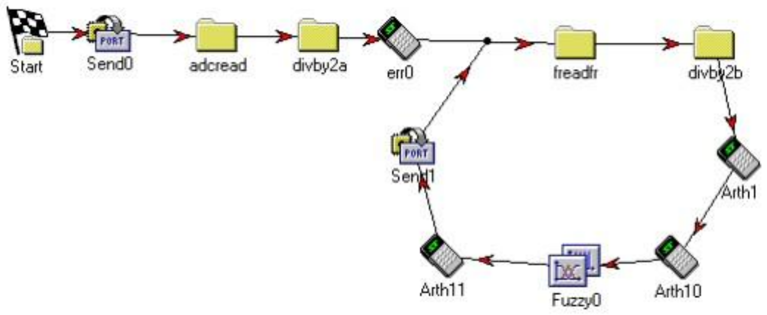


Figure 8. Structure of main program prepared for fuzzy and neuro-fuzzy controls

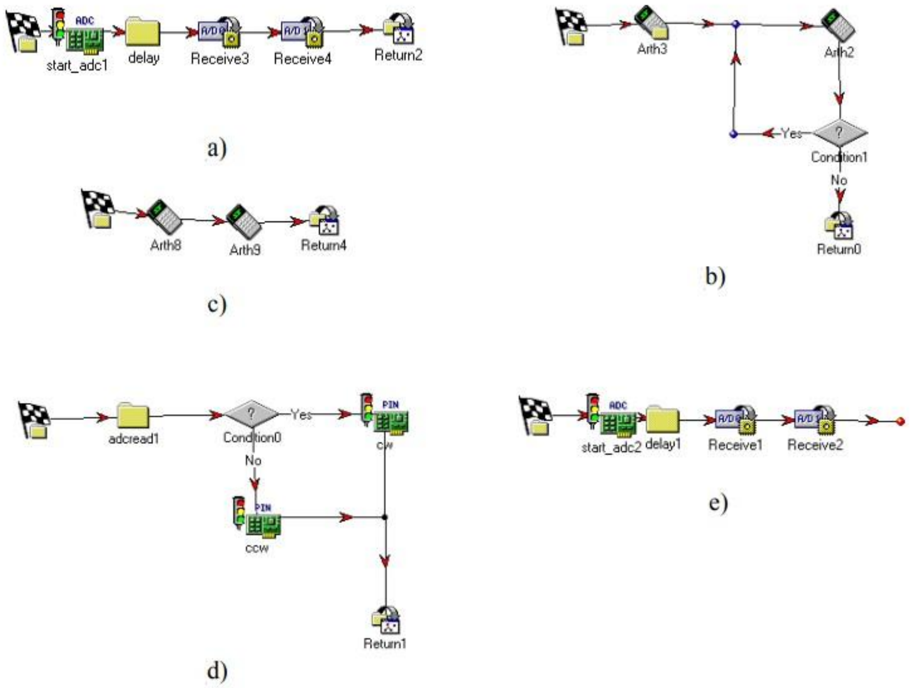


Figure 9. a) Structure of folders; a) adcread, b) delay, c) divby2a, d) freadf, e) adcread1

2.6. Fuzzy logic processing

The fuzzy block shown in figure 10 is developed for fuzzy logic processing, which involves input and output membership functions shown in figures 11a and 11b and the fuzzy rules given are in table 1. The program algorithm characterized by this block is converted into ST52 fuzzy commands after compilation in Fuzzystudio3.0. Following digitizing this data by the program involved in ST52 microcontroller, a crisp output is obtained. This output is then applied to D/A converter to produce the voltage to be applied to motor. In fuzzy logic, 7 linguistic labels; negative big (NB), negative mid (NM), negative low (NL), zero (ZR), positive low (PL), positive mid (PM), positive big (PB) are defined for the error and the change of error, and 4 linguistic labels (ZR, PL, PM, PB) are defined for the output u. Thus 49 rules given in table 1 are defined and stored using Fuzzystudio3.0 software.

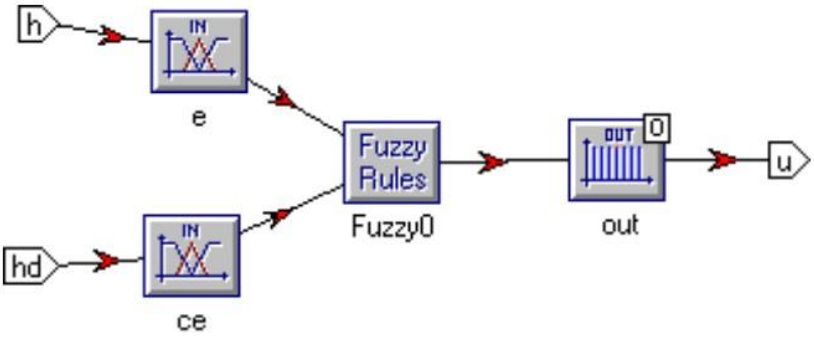
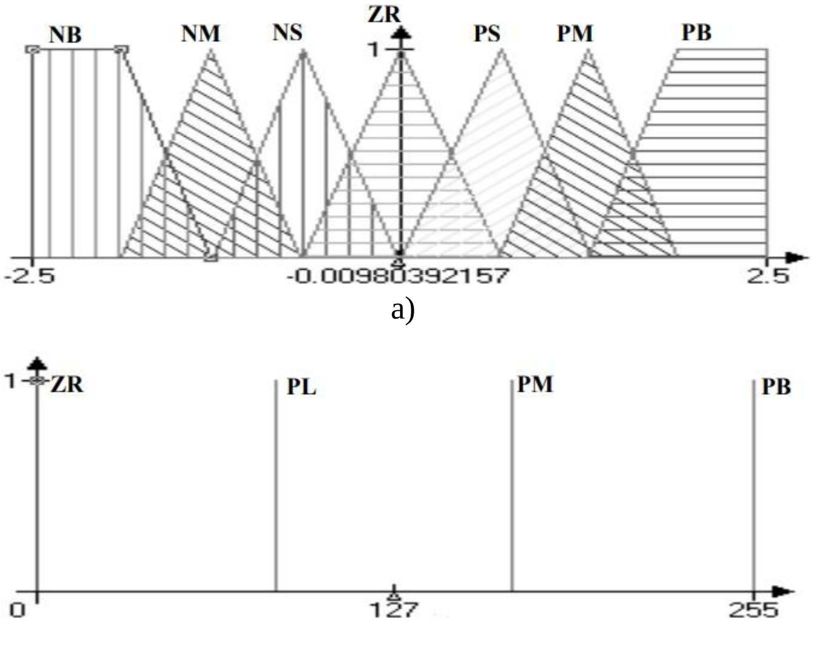


Figure 10. Structure of procedure Fuzzy0



a)

b)

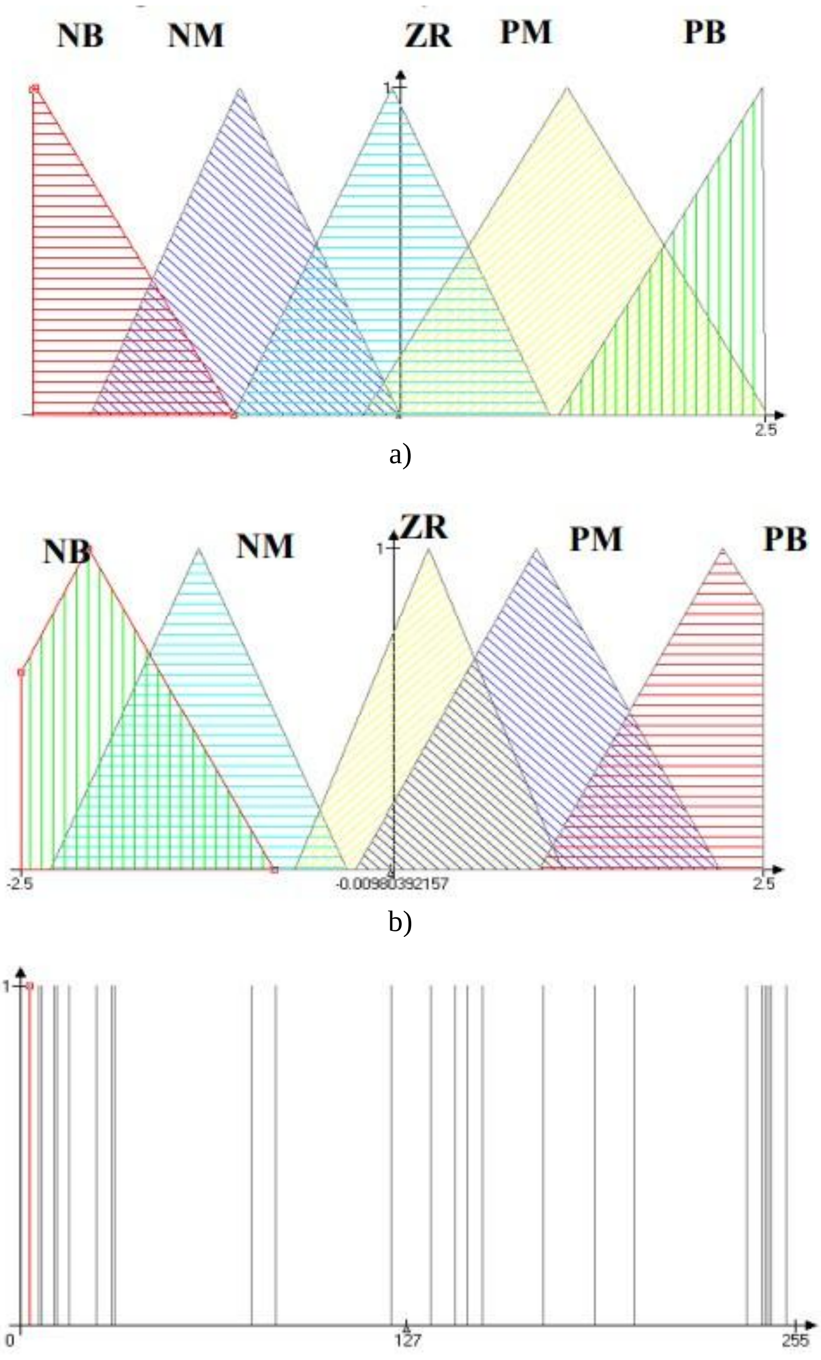
Figure 11. Membership functions for; a) input variables e and ce, and b) output u

e/ce	**NB**	**NM**	**NS**	**ZR**	**PS**	**PM**	**PB**
NB	PB	PB	PB	PB	PB	PB	PB
NM	PB	PM	PM	PM	PM	PM	PB
NS	PB	PM	PS	PS	PS	PM	PB
ZR	PB	PM	PS	ZR	PS	PM	PB
PS	PB	PM	PS	PS	PS	PM	PB
PM	PB	PM	PM	PM	PM	PM	PB
PB	PB	PB	PB	PB	PB	PB	PB

Table 1. Look up table for if-then rules

2.7.Neuro-fuzzy logic processing

To obtain proper rules and membership functions for neuro-fuzzy control, a triangular wave that provides maximum position angle at both directions is applied to the input of the system, i.e. to the motor to be controlled. The position angle is interpreted by means of potentiometers mounted to the motor shaft. A data file is created in the PC for input and output by using a DAQ board. The input-output data pairs in this file are processed in the computer program Adaptive Fuzzy Modeller software and proper rules and membership functions are obtained. The rules, membership functions and outputs obtained using this software package are shown in figure 12 and table 2. These rules and membership functions are used in Fuzzystudio3 program in Fuzzy0 folder for determination of e, ce and membership functions. The program is then loaded in microcontroller to measure the performance of the system.



a)

b)

c)

Figure 12. Membership functions for; a) input variable e, b) input variables ce, and c) for output u

e/ce	NB	NM	ZR	PM	PB
NB	Mbf0=244	Mbf5=202	Mbf10=252	Mbf15=143	Mbf20=25
NM	Mbf1=244	Mbf6=152	Mbf11=30	Mbf16=135	Mbf21=10
ZR	Mbf2=246	Mbf7=122	Mbf12=147	Mbf17=172	Mbf22=3
PM	Mbf3=247	Mbf8=84	Mbf13=189	Mbf18=30	Mbf2=6
PB	Mbf4=239	Mbf9=76	Mbf14=15	Mbf19=12	Mbf24=7

Table 2. Look up table for if-then rules

3.Laboratory test results

In the experimental setup, a 13.2 V, 2400 rev/min DC servo motor which is driven by an amplifier with 1/10 speed reductor is used. In the applications, voltage on the potentiometer is measured to specify the position angle. Change of the voltage is linearly proportional to the position angle, and -2.5 V and 2.5 V corresponds zero and 2700, respectively.

The test results of the servo controller recorded by a Tektronix TDS220 digital oscilloscope are shown in figure 13. To test the system, the input is changed suddenly to new positions and the response of the controller is observed. It can be seen from this figure that the targeted position is recovered satisfactorily, but the output cannot maintain at this position and some deviations around the set point in the form of oscillations are observed. It can also be seen that the set time reduces as the percent change in the output position reduces, which is because of the set time depend on the level of change in the position.

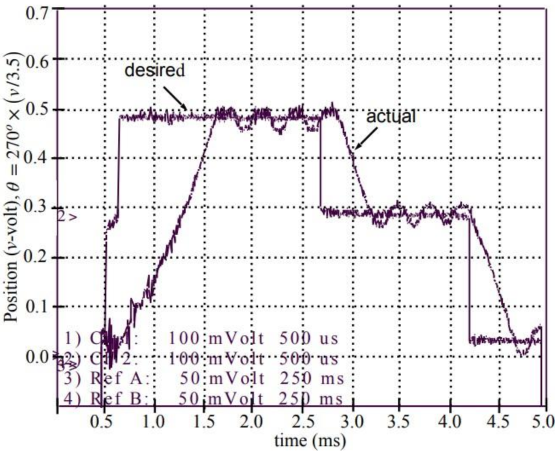


Figure 13. Servo controller position response

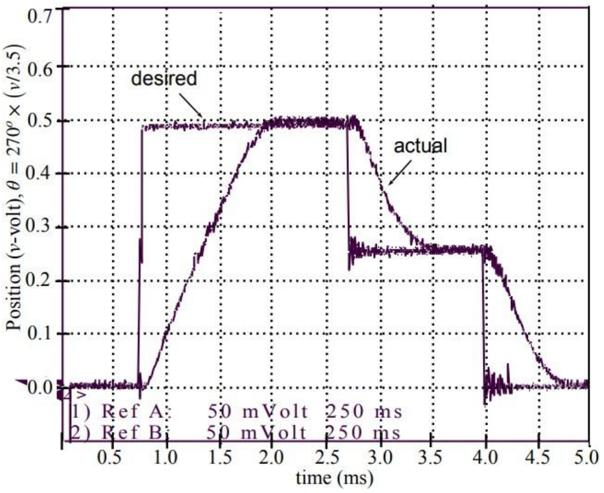


Figure 14. Fuzzy controller position response

The test results obtained using the fuzzy logic controller recorded by the digital oscilloscope are shown in figure 14. It can be observed from this figure that the output maintains at the required position satisfactorily without any deviation and the targeted position is recovered within less then 0.5 s for the first position change. It can be seen also the set time reduces as the percent change in the output position reduces, which is because of the set time depend on the level of change in the position.

The experimental results for the neuro-fuzzy controller are shown in figure 15. As it can be seen from the figure, the controller can satisfactorily set the position in a faster time than servo and fuzzy controllers. However small oscillations around the desired position are observed, which is expected to be due to electrical and mechanical noise. When simulations are carried out in MATLAB, these oscillations are not observed.

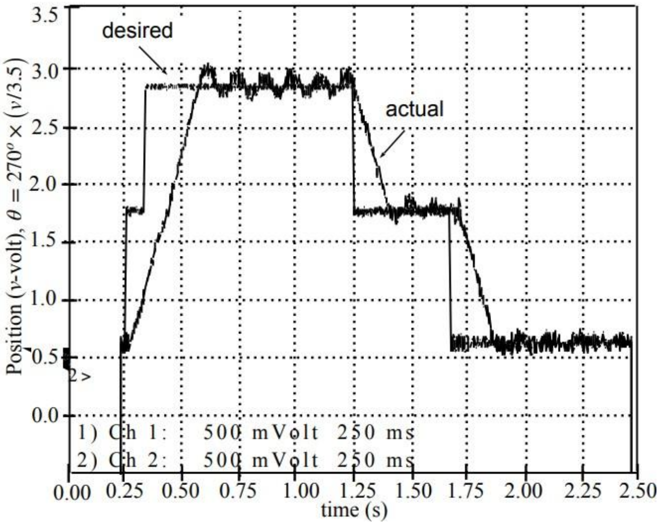


Figure 15. Neuro-fuzzy controller position response

4.Conclusion

The usage of ST52 microcontroller for position control is introduced for industrial applications. Servo control, fuzzy and neuro-fuzzy methods has been implemented. It has been shown that position control with ST52 microcontroller is easy and suitable for experimental studies on FLC. Test results show that servo controller can set the desired position, but angular deviations in the form of oscillations around the set point are observed. Nevertheless, these oscillations are not effective as much as to cause any instability. Fuzzy controller with the implemented program array is satisfactory in locating the position of the potentiometer at the output to the desired location and no deviations are observed. Neuro-fuzzy is also satisfactory in setting a position and this controller is much faster than the servo and fuzzy controllers. However some oscillations are observed.

Authors' Declaration

The authors declare no conflict of interests regarding the publication of this article.

Appendix I

Program listings for servo control

Listing of *Arth12*
en=vo-vi;
Listing of *Arth13*
en=vi-vo;

Listing of *Arth4*
count-=1;
Listingt of Arth5
count=40;

Appendix II

Program listings for servo control

Listing of *Arth3*

```
count=40;
```

Listing of *Arth2*

```
count-=1;
```

Listing of *Condition1*

```
count>0
```

Listing of *Arth8*

```
duty=vi;
//
m=0;
if duty>=128 then
duty-=128;
m+=64;
endif;
if duty>=64 then
duty-=64;
m+=32;
endif;
if duty>=32 then
duty-=32;
m+=16;
endif;
if duty>=16 then
duty-=16;
m+=8;
endif;
if duty>=8 then
duty-=8;
m+=4;
endif;
if duty>=4 then
duty-=4;
m+=2;
endif;
if duty>=2 then
m+=1;
endif;
//
vi=m;
```

Listing of *Arth9*

```
duty=vo;
//
m=0;
if duty>=128 then
duty-=128;
m+=64;
endif;
if duty>=64 then
duty-=64;
m+=32;
endif;
if duty>=32 then
duty-=32;
m+=16;
endif;
if duty>=16 then
duty-=16;
m+=8;
endif;
if duty>=8 then
duty-=8;
m+=4;
endif;
if duty>=4 then
duty-=4;
m+=2;
endif;
if duty>=2 then
m+=1;
endif;
//
vo=m;
```

Listing of *err0*

```
en1=vo-vi;
```

Listing of *Condition0*

```
vo>vi
```

Listing of *Arth1*

```
en=vo-vi;
ce=en-en1;
en1=en;
```

Listing of *Arth10*

```
h=127;
h+=en;
hd=127;
hd+=ce;
```

Listing of *Arth11*

```
u=u&248;
```

References

1. Y. Ye, Ch.-B. Yin, Y. Gong, J. Zhou, Mechanical Systems and Signal Processing **83** (2017) 241.
2. H. Navabi, S. Sadeghnejad, S. Ramezani, J. Baltes, Advances in Fuzzy Systems 2017 (2017) 1.

3. H. – C. Lu, R. Guang-Zheng, and D. Teng-Fei, Mechatronics and Automation Engineering: Proceedings of the International Conference on Mechatronics and Automation Engineering (ICMAE2016) 2017 3.
4. T. Bucella, Microchip Technology Inc. (1997) 4.
5. G. Muscato, Microprocessors and Microsystems **23**(6) (1999) 385.
6. H.C. Lu, 1997 IEEE International Conference on Intelligent Processing Systems (Cat. No.97TH8335), Beijing, China **1** (1997) 242.
7. E.S. Mohammed, S. Nazlibilek, F.S. Benkoura, International Journal of Modern Engineering Research **8**(12) (2018) 33.
8. A.G. Sreenatha, M. Pradhan, Acta Astronautica **50**(11) (2002) 665.
9. N. Khongkoom, A. Kanchanathep, S. Nopnakeepong, S. Tanuthong, S. Tunyasrirut and R. Kagawa, 2000 TENCON Proceedings, Kuala Lumpur, Malaysia **3** (2000) 354.
10. B. Sapiński, Transactions of the VŠB–Technical University of Ostrava, Mechanical Series **2**(LIV) (2008) 127.
11. H. Ammar, W. Salim, Internatonal Journal Of Integrated Engineering **12**(3) (2020) 235.
12. H. Dirman, M.N. Ribuan, W.H. Wan Abas, Hidayat, E. Johana, H.B. Wahid, R. Ghazali, H. Abdul Rahman, MATEC Web of Conference **248** (2018) 02005.
13. R. Patjoshi, K. Mohapatra, International Journal of Advanced Technology and Engineering Exploration **8**(5) (2010) 88.
14. H. Franc, R. Šafarič, 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010), Budapest, Hungary (2010) 417.
15. ST 52X301, Preliminary Data Sheet, ST-Microelectronics (1998).
16. L.A. Zadeh, Information and Control **8** (1965) 338.
17. C.C. Lee, IEEE Transactions SMC **20**(2) (1990) 404.
18. C.C. Lee, IEEE Transactions SMC **20**(2) (1990) 419.
19. M.S. Ahmed, U.L. Bhatti, F.M. Al-Sunni and M. El-Shafei**,** Fuzzy Sets and System **124**(2) (1 DEC 2001) 231.
20. C.F. Wong, J. Shippen and B. Jones**,** International Journal of Machine Tools and Manufacture **38**(9) (1998) 1109.
21. H.T. Nguyen, N.R. Prasad, C.L. Walker, E.A. Walker, A first Course in Fuzzy and Neural Control, CRC press LCC Florida (2003) 312p.
22. R.J. Wai, H.H. Lin, F.J. Lin, Neurocomputing **35** (2000) 91.
23. S.M. Kim and W.Y. Han**,** Control Engineering Practice **14**(5) (2006) 481.
24. J.H. Tarn, F.Y. Hsu, IEEE International Conference on Systems, Man and Cybernetics, USA **2** (1992) 1289.
25. Z. Kovacic, M. Balenovic, S. Bogdan, IEEE Control Systems Magazine **18**(3) (1998) 41.
26. W. Panhai, X. Dianguo, S. Jingzhuo, Industrial Electronics Society, IEEE 2002 28th Annual Conference of the Industrial Electronics Society, Spain **3** (2002) 2137.