FUZZY CONTROLLERS FOR SINGLE POINT CONTROLLER-1 (SPC-1) SYSTEMS

Azzam A Marouf
Al-Khawirizmi College of Engeneering
Information Engineering Department University of Baghdad, Iraq.

Omar W. Abdul-Wahab
College of engeneering
Computer Engineering Department University of Baghdad, Iraq.

Luma A. Kamel
College of engeneering
Computer Engineering Department University of Baghdad, Iraq.

ABSTRACT
Advances in computer technology have introduced computers everywhere. One of the fields that the computers have entered is the field of process control and data acquisition systems. On the other hand, fuzzy control is emerging as an alternative to conventional control to control different systems. This paper is concerned with applying fuzzy control to a locally designed and manufactured process controller. This controller is designated by single point controller-1 (SPC-1). It is basically a flexible and general-purpose, stand-alone, single-point controller.
The CPU section of the SPC-1 is the AT89C51 general purpose microcontroller. The fuzzy control algorithms were implemented as programs to be executed by this microcontroller. These programs were written in C language and were translated to machine language by Keil8051 C compiler μ Vision V5.1.

الخلاصة

هذا البحث يهتم بتطبيق السيطرة الضبابية على مسيطر عملية تم تصميمه وتصنيعه محليا . انه في الاســـاس مسيطر ذو نقطة واحدة ، مرن ومتعدد الاغراض .

وحدة المعالجة المركزية لهذا المسيطر هي AT89C51 . خوارزميات السيطرة الضبابية قد طبقت على شكل برامج تم تنفيذها من قبل هذا المسيطر . كتبت هذه البرامج بلغة C وتمت ترجمتها الى لغة الماكنة بواسطة المترجم .KEIL8051 C Compiler u Vision V5.1.

KEY WORDS
Fuzzy control, single point controller.

INTRODUCTION
The simplest and the most usual way to implement a fuzzy controller is to realize it as a computer program on a general purpose processor. However, a large number of fuzzy control applications require a real-time operation to interface high-speed external devices. For example, automobile speed control, electric motor control and robot control are characterized by severe speed constraints. Software implementation of fuzzy logic on general purpose processors cannot be considered as a suitable design solution for this type of application. In such cases, design specifications can be matched by specialized fuzzy processors.

The requirements to the hardware implementation are:

- high-speed performance;

- low complexity;
- high flexibility.

Low complexity means that algorithms for fuzzy processing, fuzzification and defuzzification have to be very simple and demand as small an amount of memory as possible for their realization. Flexibility means the ability of the hardware to be used successfully in different applications and configurations (Reznik 1997).

SINGLE POINT CONTROLLER

The single point controller is basically a flexible and general-purpose, stand-alone controller. The hardware organization of the SPC-1 mainly consists of the following sections: Microcomputer section, memory section, communication section, data acquisition section and front panel section. The schematic diagram of these sections is shown in **Fig (1).** This microprocessor-based process controller was designed and implemented using general-purpose microcontroller Atmel 89C51 (Amera 2000).

The implemented software incorporates control and management tasks, communication with a central master computer, and finally application library modules to allow easy installation of a complete application program module.

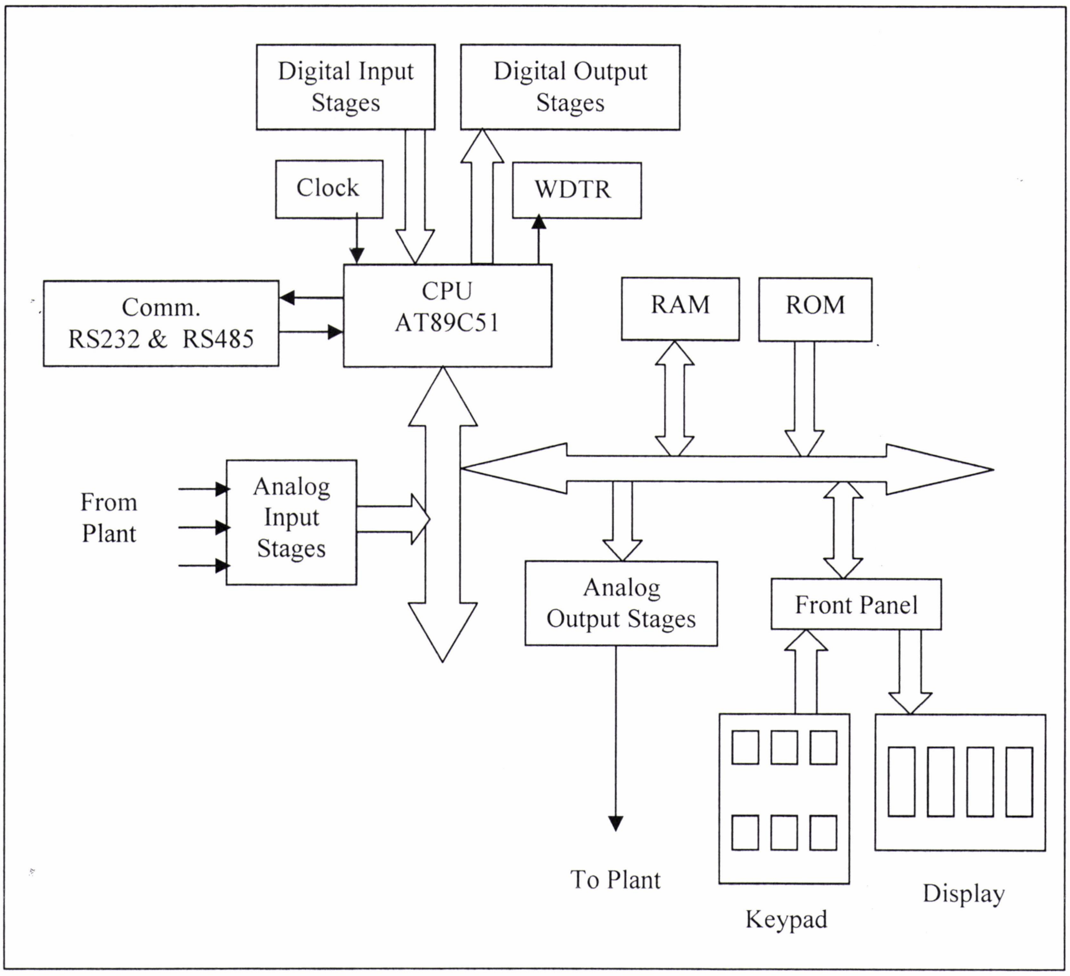

Fig. (1) Schematic diagram of the main sections of the SPC-1.

IMPLEMENTATION OF FUZZY CONTROLLERS

The CPU section of the SPC-1 is the AT89C51 general purpose microcontroller. Therefore, the fuzzy logic controllers were implemented as software programs to be executed by this microcontroller. The 8051 microcontroller family is supported in only five languages-assembly, PL/M, C, Forth, and BASIC. In this work C language was used since C is much easier to write than assembly language, because the development software manages the details (Thomas 1998). These programs were translated to machine language by Keil8051 C compiler μ Vision V5.1.

Fuzzification Strategy

Singleton fuzzification strategy was used since, without a presence of noise, we are absolutely certain that the inputs take on their measured values (and no other value), and since it provides certain savings in the computations needed to implement a fuzzy system (relative to, for example, "Gaussian fuzzification," which would involve forming bell-shaped membership functions about input points, or triangular fuzzification, which would use triangles (Passino 1998).

Data Base

Since our compiler has no floating-point library, the range of any membership function was transformed from the interval [0,1] to the set {0, 1, 2, ...,409}. The selection of the set {0, 1, 2, ..., 409} is the most suitable selection, since the resolution of our ADC is 12-bit, and the range of the input voltages to it is (0-10)Volts, hence by dividing the largest 12-bit binary number $(1111\ \ 1111\ \ 1111)_2 = (\mathrm{FFF})_{16}$ by the maximum voltage $(10)_{10} = (\mathrm{A})_{16}$ we obtain:

$$\frac{(\mathrm{FFF})_H}{(\mathrm{A})_{16}} = (199)_{16} = (409)_{10}.$$

Therefore, the horizontal axes (e-axis, Δe-axis, and u-axis representing the error, change of error and control signal respectively)had to be scaled by the factor $(409)_{10}$, and it is more suitable from the computational point of view to scale the vertical axes (the membership function axes) by the same factor.

For each input of the fuzzy controller five triangular membership functions were used on the universe of discourse for e and Δe. The output membership functions are singletons centered at the appropriate positions, see **Fig. (2).** Different scaling factors were used in the experimental results to show their effect on the closed loop response.

To minimize the time required to compute the control variable, a minimization technique was used that make use of the fact that no more than four rules would be "ON" at one time. The other rules would be "OFF" since the certainty of their antecedents would equal to zero for a particular input values. **Fig. (3)** shows the flow chart of the minimization technique used. This flow chart contains the following variables:

- $f1$ and $f2$: Two flags used to mark the current index of the input membership function.
- i: The index of the current input membership function.
- $mf1[i]$ and $mf2[i]$: The values of the input membership functions of index i.
- $istar$ and $jstar$: The indices of the first nonzero value of $mf1$ and $mf2$, respectively.

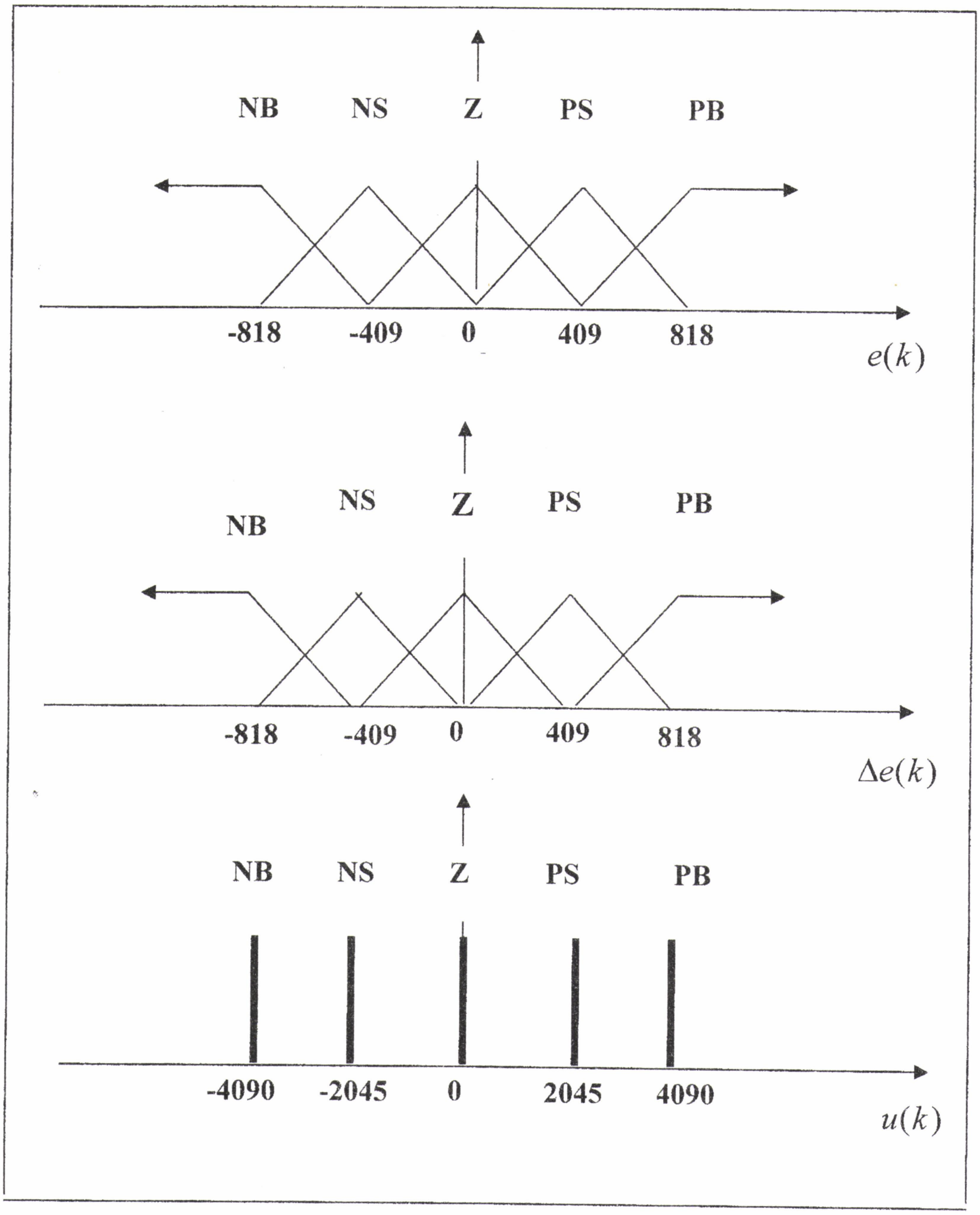

FIG (2) The input and output membership functions for the implemented fuzzy controller.

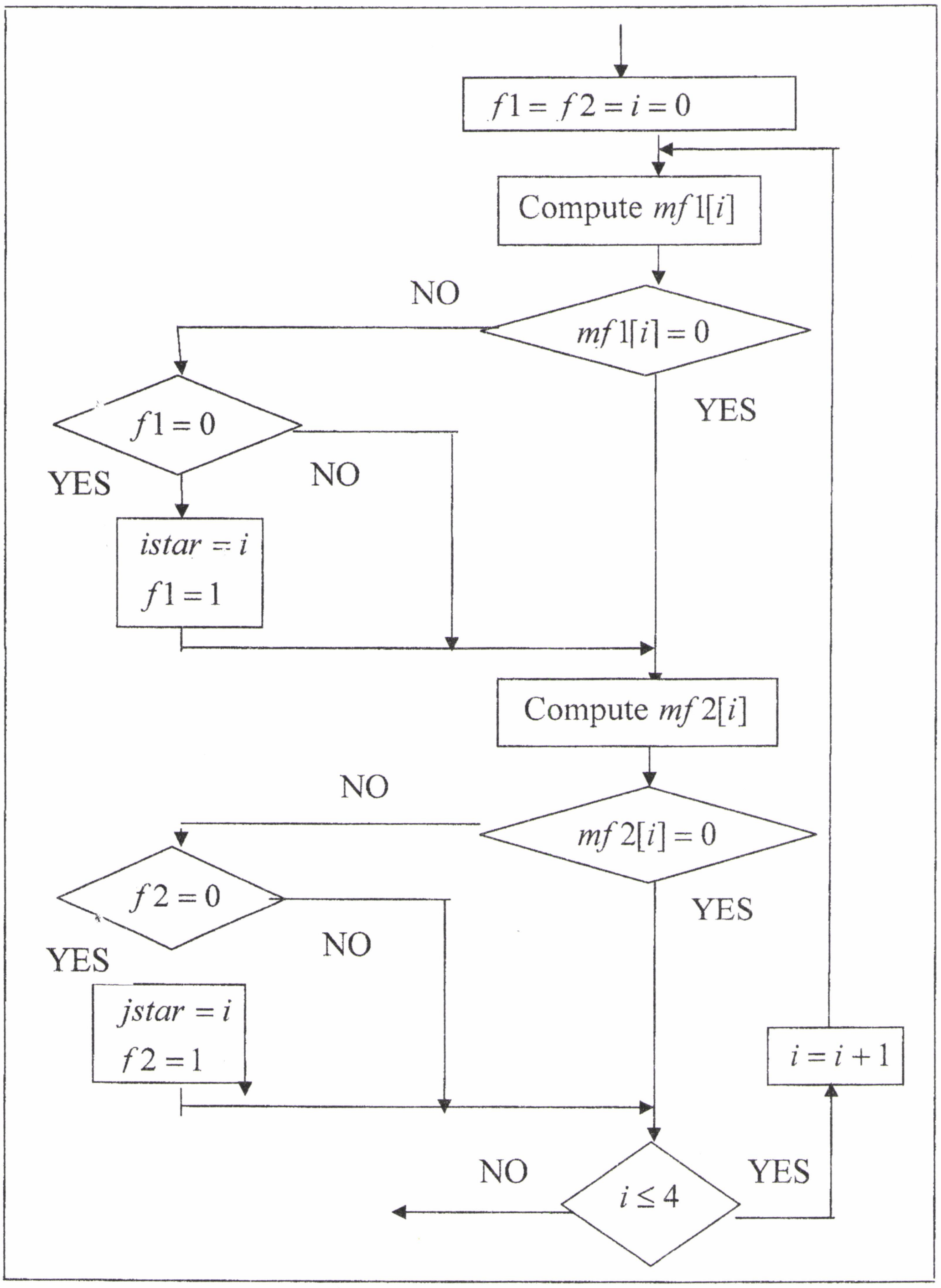

FIG (3) The minimization technique used to reduce the computation time of the fuzzy controller output.

Inference Engine

The minimum operator was used to represent the meaning of the antecedent part of the rule, since it reserves the range of the membership functions as $\{0,1, \ldots, 408, 409\}$, while if the product operator was use instead, the range of the certainty of the antecedent part would be $\{0,1,\ldots,167281\}$. The inference step used here is "implied fuzzy sets", since using the overall implied fuzzy set in defuzzification is often undesirable for two reasons:

1- The overall implied fuzzy set is itself difficult to compute in general, and

2- The defuzzification techniques based on an inference mechanism that provides the overall implied fuzzy set are also difficult to compute. It is for this reason that most existing fuzzy controllers use defuzzification techniques based on the implied fuzzy sets, such as center-average or center of gravity (COG) (Passino 1998).

Defuzzification Strategy

The Center-average defuzzification technique was used. The reason for this choice is that since each membership function for the output fuzzy sets has a maximum value of 1, then for many representation techniques of the meaning of the rule we have

$$\sup\{\mu_{\text{implied fuzzy set } i}\} = \mu_{\text{antecedent } i}$$

This implies that the shape of the membership functions for the output fuzzy sets does not matter; hence, we used singletons centered at the appropriate positions. On the other hand, the COG defuzzification technique contains the computation of the area under the membership function of each implied fuzzy set, which results in large numbers that will exceed the range that is allowed for the variables of the fuzzy controller algorithm, while in the center-average defuzzification technique no number will exceed the range that is allowed by those variables.

EXPERIMENTAL RESULTS

The implemented fuzzy controllers were used to control a simulated second order system and a real thermal system. The experimental results were as follows:

Simulated Second Order System

This system is an electrical circuit built with two operational amplifiers and the associated resistors and capacitors to simulate a second order differential equation. The circuit diagram of this system is shown in **Fig. (4).** The transfer function of this system is:

$$\frac{Y(s)}{U(s)} = \frac{1}{(s+1)^2}$$

The objective is to make the output voltage of this system equal to the reference input voltage, which was taken to be two volts.

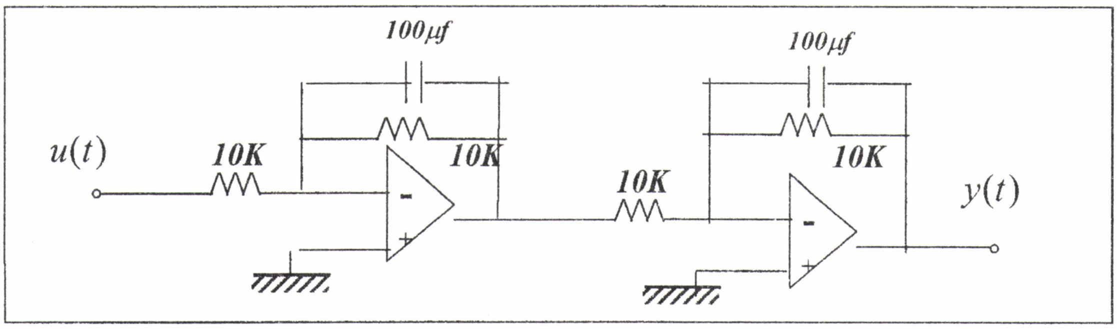

Fig. (4): Circuit diagram of the simulated second order system

Figs. (5), (6), (7), (8) and (9) show the step response of the simulated second order system when a PI-like fuzzy controller was used to control this system, for different scaling factors. From these responses we see that a PI-like fuzzy controller always gives zero steady state error, but the response may have an overshoot for some values of the scaling factors. **Fig. (5)** clearly indicates the best response. g1, g2 and h represent the scaling factors for e, Δe, and u respectively.

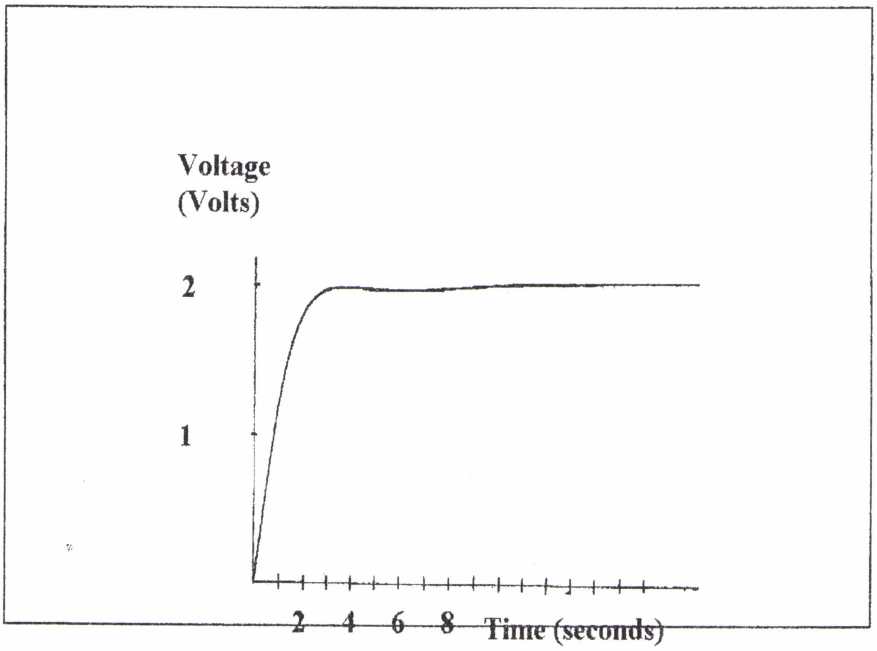

Fig. (5) Step response of the second order system using PI-like fuzzy controller with g1=1, g2=1 and h=1.

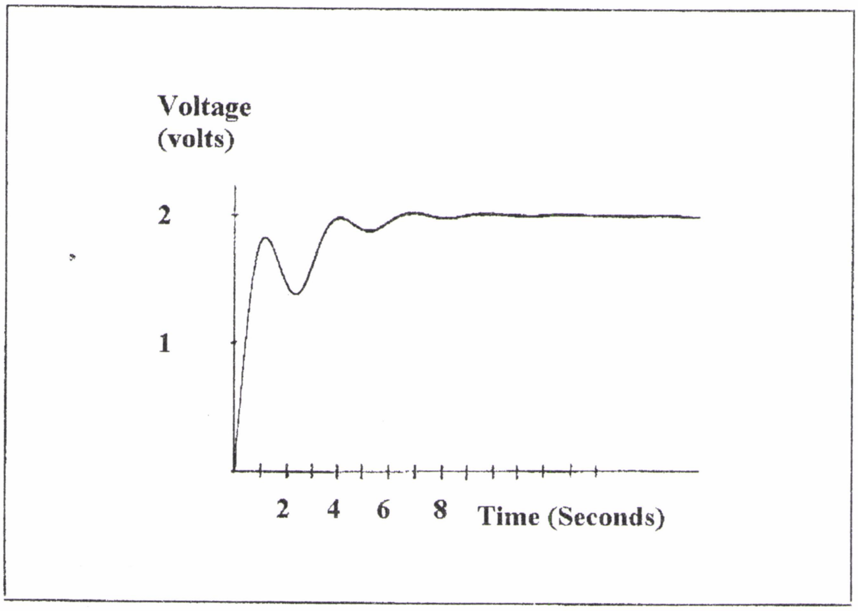

Fig. (6) Step response of the second order system using PI-like fuzzy controller with g1=1, g2=1 and h=5.

Fig. (10) and (11) show the step response of the simulated second order system when a PD-like fuzzy controller was used to control this system, for different scaling factors. From these responses we see that a PD-like fuzzy controller gives good transient response, but it always has a steady state error because of the absence of the integration action in this type of controller.

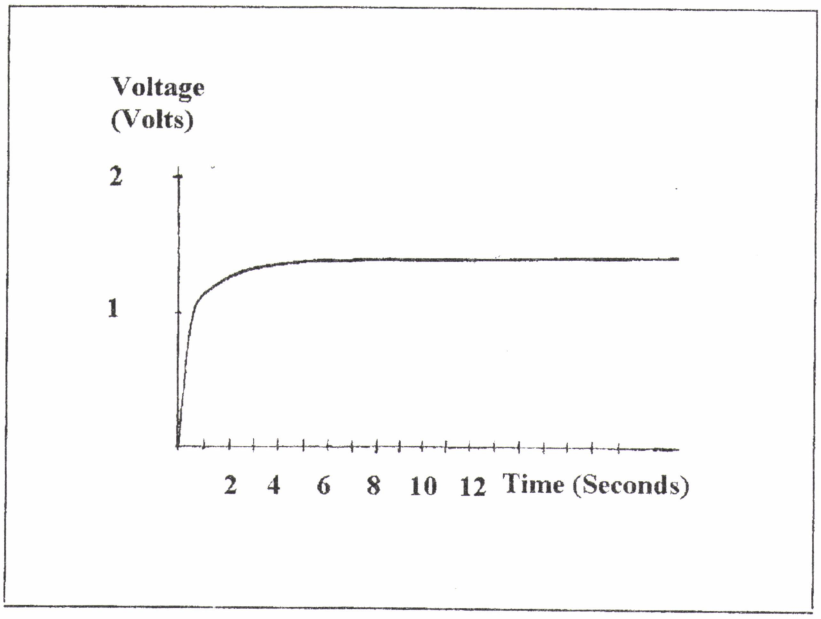

Fig. (10) Step response of the second order system using PD-like fuzzy controller with g1=2, g2=1 and h=2.

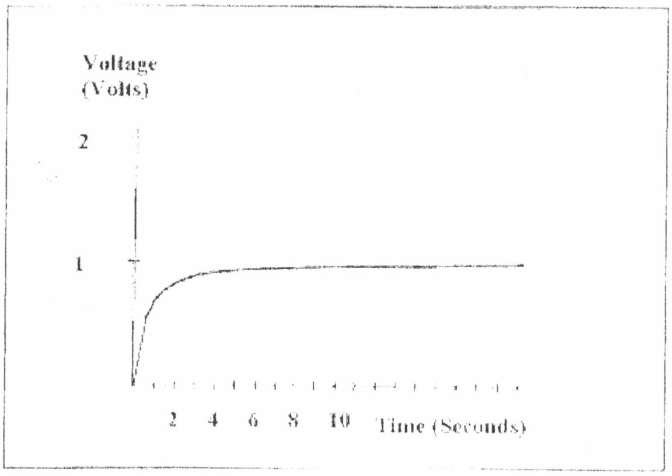

Fig. (11) Step response of the second order system using PD-like fuzzy controller with g1=2, g2=1 and h=1.

Fig. (12) shows the step response of the simulated second order system when a P-like fuzzy controller was used to control this system. From this response we see that a P-like fuzzy controller is the worst controller for this system because of the absence of the integration and differentiation actions.

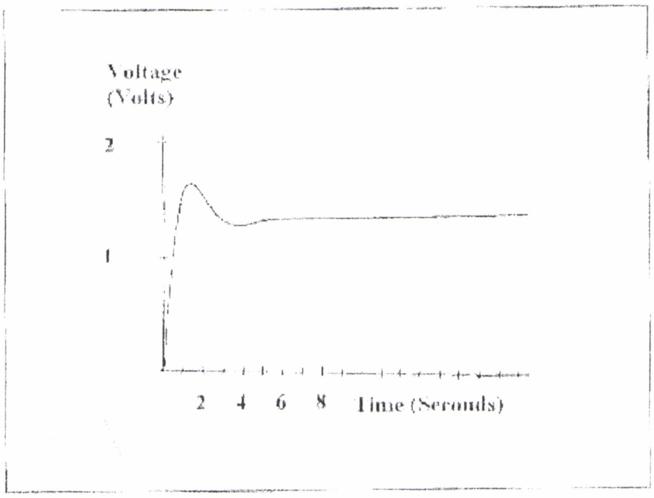

Fig. (12) Step response of the second order system using P-like fuzzy controller with g1=2 and h=2.

Real Thermal System

This system consists of a heater, which is the controlled element, a sensor to measure the temperature of the heater and an actuator that provides the necessary control signal to drive the heater to adjust its temperature. The block diagram of this system with a controller is shown in **Fig. (13).** The objective is to make the temperature of the heater reaches the desired temperature level specified by the operator. In our experiments the desired temperature was $40^{\circ}C$.

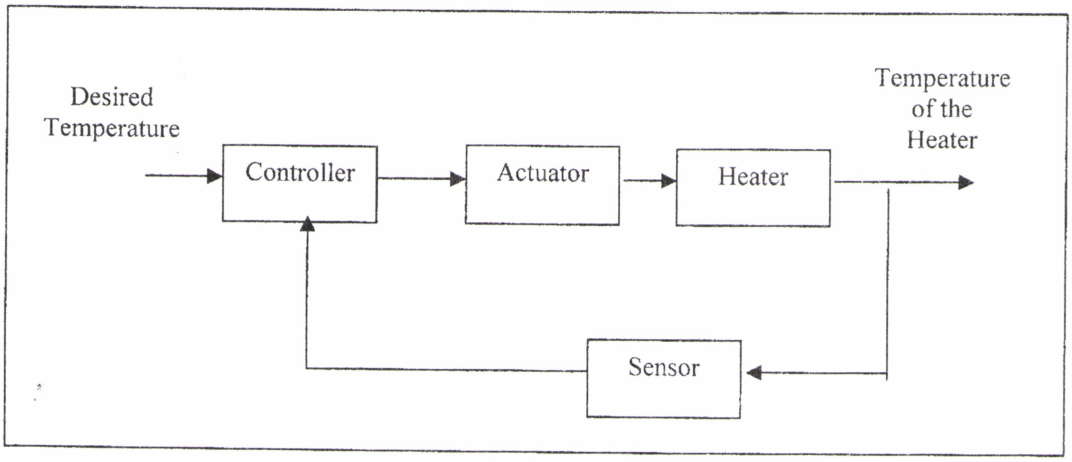

Fig. (13) Block diagram of the thermal system with a controller.

Applying conventional PID controller to this system gives the response shown in **Fig. (14).** This response is better than the previous one since the differentiation action of the conventional PID controller reduces the overshoot and eliminates the oscillation around the set point.

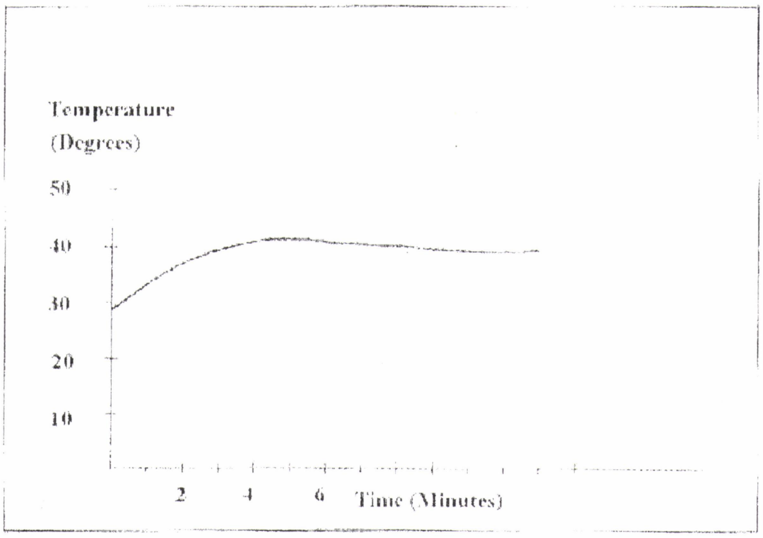

Fig. (14) Step response of the thermal system using conventional PID controller with Kp=4, Ti=300,Td=0.5.

Applying PI-like fuzzy controller to this system gives the response shown in **Fig. (15).** This response has a large overshoot and a continuous oscillation around the set point due to the absence of differentiation action in this type of fuzzy controllers. The response can be improved using a PID-like fuzzy controller but unfortunately such a controller couldn't be implemented the earlier mentioned compiler. The problem with such compiler was its limited translation capabilities which was restricted to programs having 2Kbyte binary code.

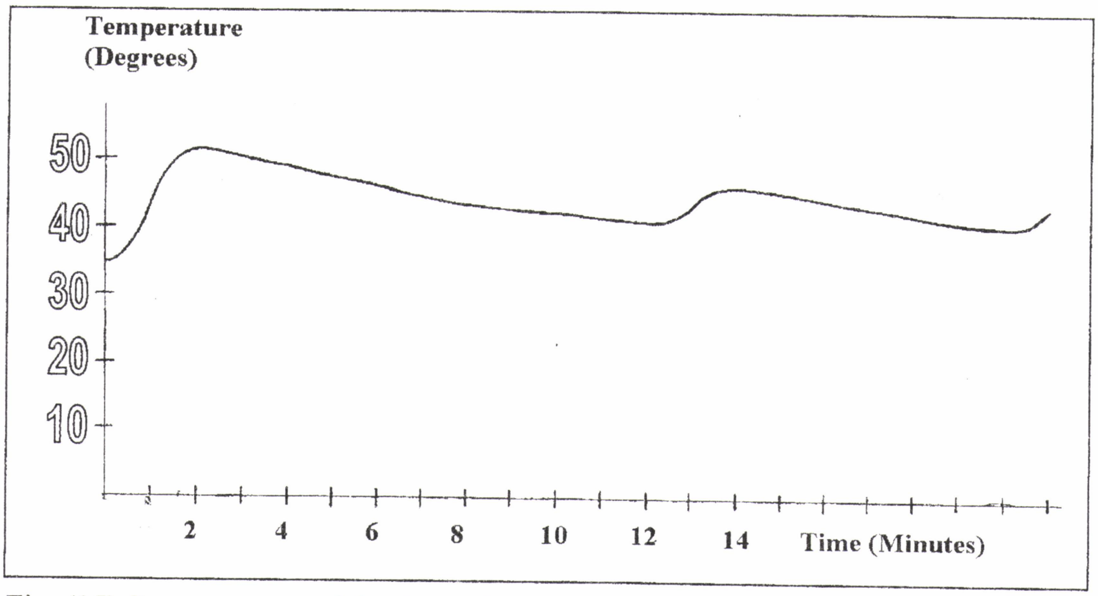

Fig. (15) Step response of the thermal system using PI-like fuzzy controller with g1=1, g2=1, h=1.

CONCLUTIONS

The CPU section of the SPC-1 is the AT89C51 general purpose microcontroller. Therefore, the fuzzy logic controllers were implemented as software programs to be executed by this microcontroller. These programs were written in C language and were translated to machine language by Keil8051 C compiler μ Vision V5.1. In general, if there is a need for higher operation speed, a specialized fuzzy processor can be used. In our work, the speed of the CPU is 12MHz, which is sufficient to execute the fuzzy logic control algorithm in a sampling period of 0.2 second.
The execution time of the fuzzy logic control algorithms was minimized by using an optimization technique that made use of the fact that not all of the rules in the rule base are "ON" for certain input variables. Therefore, with this optimization technique, the fuzzy control algorithm does not waste time with computations that concern "OFF" rules.
P, PI, and PD-like fuzzy controllers were implemented successfully to control simple simulated and real systems. However, to implement PID-like fuzzy controller a compiler with better specifications must be used.

REFERENCES

Amera A. (2000), Single Point Controller, M.Sc. Thesis Submitted to University of Technology.

Dimiter D. (1996), An Introduction To Fuzzy Control, Springer-Verlag Berlin Heidelberg.

Gaith A. (1997), Programmable Multi-Channel VME-Based Process Controller, M.Sc. Thesis Submitted to University of Technology.

Mohammed M. (1993), Rule-Based Computer of Industrial Process Using Fuzzy Logic, M.Sc. Thesis Submitted to University of Technology.

Passino, Kevin M.: (1998), Fuzzy Control, Addison-Wesley Longman, Inc.

Reznik, Leond: (1997), Fuzzy Controllers, Newnes.

Schultz, Thomas W.: (1998), C and the 8051, Prentice Hall PTR.