

Military Technical College
Kobry Elkobbah,
Cairo, Egypt



8th International Conference
on Aerospace Sciences &
Aviation Technology

PROPOSED TUNING TIPS FOR FUZZY CONTROLLERS USING OFFLINE AND ONLINE EXPERIMENTAL AIDS

S.A. Hammad*, A. Hassan*, A.A.Wahdan* and M.A.Sheirah*

Abstract

This paper presents a new real time fuzzy control toolbox. It's associated with online and offline monitoring tools. This is to illustrate the details of the fuzzification, rule base evaluation, and defuzzification processes.

A new systematic tuning approach is proposed. The new toolbox supports both PD and PI Fuzzy Control. Real time experimental results of different processes show the power of the toolbox and the proposed fuzzy control tuning method. Real time Robustness experiment shows the need of adaptation if the process change from slow process to a faster one.

Key Words

Fuzzy, FLC (Fuzzy Logic Controllers).

1. Introduction

Since Lotfi A. Zadeh initiated Fuzzy Logic in 1965, many researchers proved that fuzzy logic is a profitable tool for controlling complex industrial processes. Constructing FLC (Fuzzy Logic Controllers) had received a lot of interest. Papers are made to evaluate its performance [1], [2]. Many researchers tried to get an analytical model for fuzzy controllers and to prove that its behavior looks like a nonlinear PD controller, or a nonlinear PI controller as discussed in [3]. Nowadays, fuzzy has become a keyword for marketing. So we must differentiate when we should use fuzzy logic and when we don't need it. Fuzzy proved itself in nonlinear processes and multi-model processes. Is fuzzy adequate for linear processes? Is it a robust controller against process variation? We try to answer these questions by building a real time fuzzy logic PD-like or PI-like controller with an online and offline monitoring features.

*Ain Shams University, Faculty Of Engineering, Computer and Systems Department

This paper is organized as follows, the next section discusses the parameters of the fuzzy controllers. It's followed by a discussion of P-like, PD-like and PI-like fuzzy logic controllers. Section 3 demonstrates the developed package showing the power of the fuzzy logic controller and its monitoring features and provides a systematic approach to tune such processes with this controller. Section 4 presents the experimental results of the package with different processes. Section 5 justifies the robustness of fuzzy controller.

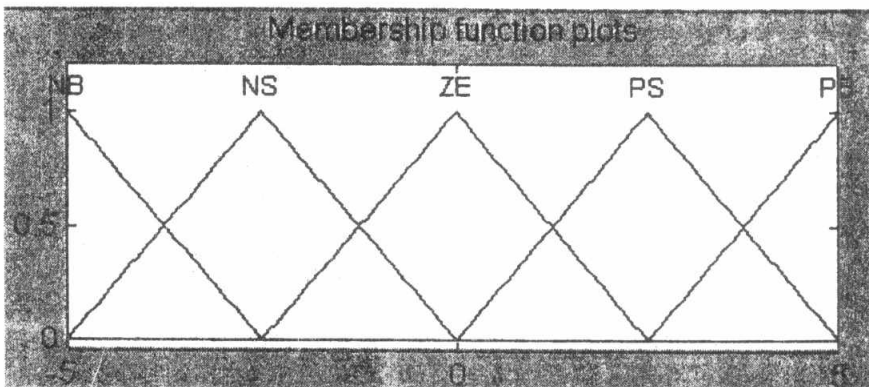
2. Fuzzy Controller Parameters and Conventional Controllers

2.1. Fuzzy Parameter Design

In the following we illustrate the fuzzy inference system parameters.

Membership functions

Five triangular membership functions are used over the universe of discourse in all inputs and outputs of the fuzzy inference system. The overlapping is 0.5 in all membership functions.



Rule base

For the PI and PD controllers, 25 rules are used. They are described as in the following table

e	NB	NS	ZE	PS	PB
e	NB	NS	ZE	PS	PB
B	NB	NB	NB	NS	ZE
S	NB	NB	NS	ZE	PS
E	NB	NS	ZE	PS	PB
S	NS	ZE	PS	PB	PB
B	ZE	PS	PB	PB	PB

As discussed in Mamdani controller, "min" is considered as the "And Method", "max" as the "Or Method" in the new toolbox.

2.2. Types Of Fuzzy Controller

a. P-like Fuzzy Logic Controller

The simplest fuzzy controller presented so far has the following block diagram

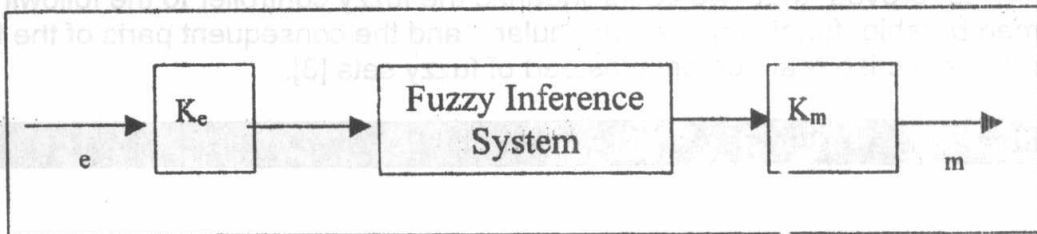


Fig. 1

Where e is the error signal and m is the control. The fuzzy inference system contains the fuzzification, rule evaluation, and defuzzification stages. The scaling factor K_e and K_m are the tuning parameters. K_m usually set to 1 and K_e is used to vary the universe of discourse of the error.

The rule base is usually in the form

If e is e_i , then m is m_i (e_i is a label of error and m_i is a label of control)

Usually e_i and m_i are linguistic values such PB (Positive Big), PS (Positive Small), etc. So two rules of the rule base may be

If e is PB then m is PB
 If e is ZE then m is ZE

This means that control value is in proportional relation to the error. This is analogous to the p controller which have the mathematical form $m = K_p * e$. This is the reason that this fuzzy controller is considered a p-like fuzzy controller in spite of its inherent nonlinearity due to the defuzzification process.

b. PD -Like Fuzzy Logic Controller

The idea of the PD – like fuzzy controller is the addition of the change in error (de) as an input to the fuzzy controller. Then the control is in direct nonlinear proportional relation with the error and the change in error. The dominance of error or change in error is set by the scaling factors K_e and K_{de} .

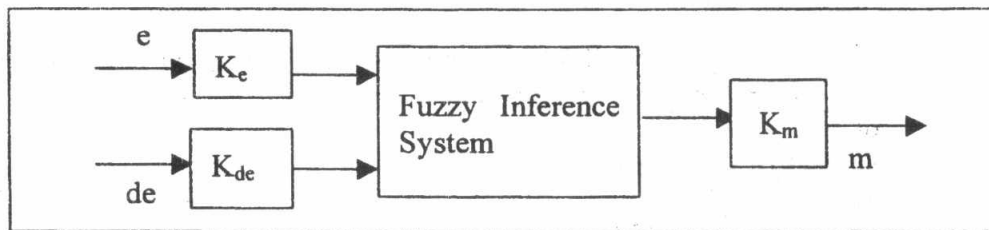


Fig. 2

It was proved that we could linearize the fuzzy controller to the following form if the membership functions are triangular, and the consequent parts of the fuzzy control rules are crisp real numbers instead of fuzzy sets [3].

$$m = A + B e + C de \tag{1}$$

The conventional PD controller has the mathematical form

$$m = 100/p (e + D * de/dt) \tag{2}$$

The symmetry in the equations gives us the chance to evaluate the performance of this controller to be fast controller that doesn't guarantee zero steady state error. It increases the damping of the system thus reduces the overshoot and settling time of the system response.

c. PI -Like Fuzzy Logic Controller

The idea of the PI – like fuzzy controller is the addition of an integrator after the PD controller. Then the change in control is in direct nonlinear proportional relation with the error and the change in error. The dominance of error or change in error is set by the scaling factors K_e and K_{de} .

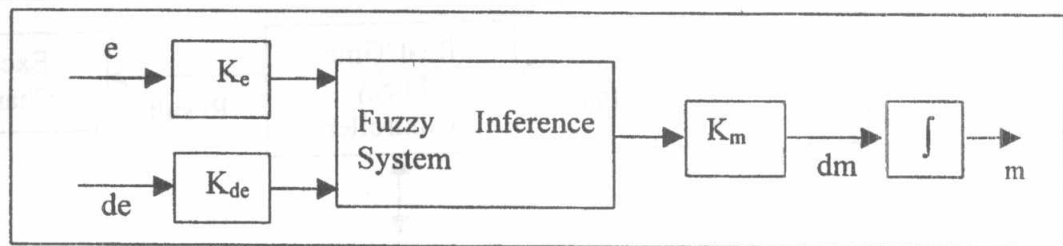


Fig.3

As discussed in b, we could linearize this fuzzy controller to the following form under the same conditions.

$$m = At + B \int e(t) dt + C e \quad (3)$$

The conventional PI controller has the mathematical form

$$m = 100/p (e + 1/I \int e(t) dt) \quad (4)$$

The symmetry in equations 3,4 gives us the chance to evaluate the performance of this controller to be slow controller that guarantees zero steady state error.

3. Developed Package and Systematic Tuning Approach

3.1. Developed Package

The following figure is a block diagram of the developed package

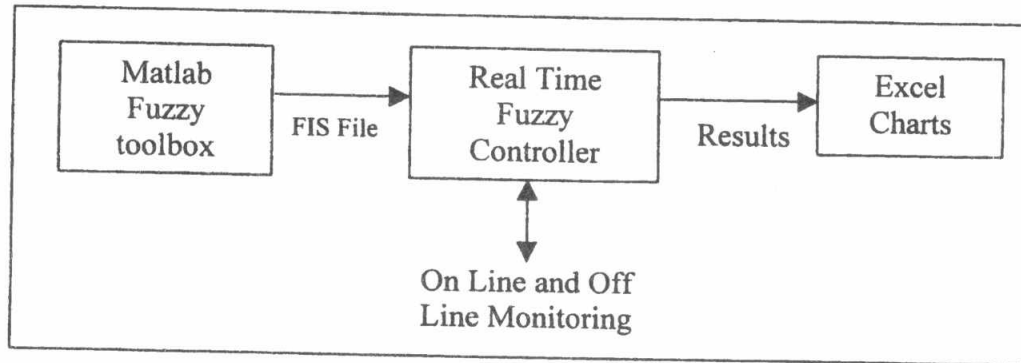


Fig. 4

we use the **Matlab fuzzy toolbox** to define

- a. The membership functions of the inputs and outputs of the real time fuzzy logic controller (error and derivative of error as input and change in control or control as output).
- b. The rule base of the real time fuzzy logic controller.
- c. The fuzzification and defuzzification method.

This data is saved as FIS (fuzzy inference system) file and send to the real time fuzzy logic controller being presented.

The **real time fuzzy logic controller** includes the logic of the fuzzy inference system. The operation sequence will be as follows:

- a. Take the process controlled variable via ADA card.
- b. Calculate error, derivative of error.
- c. Infer the fuzzy inference system with its inputs and calculate its output (Control or change in control).
- d. Send the value of the control via ADA card to the process.

The **On Line Monitoring Feature** includes drawing the process output, the error and the change in error signals. It includes also the values of these signals at each sample. The main feature in the online monitoring is that it displays the fuzzy values of fuzzy inference system input and outputs, and the rules actually fired at each sample.

The **Off Line Monitoring Feature** includes a cursor to show the signals stated above for any optional sample. The addition here is the possibility to display visually the interpretation of the fuzzification and defuzzification processes.

The following figure shows the off line monitoring feature.

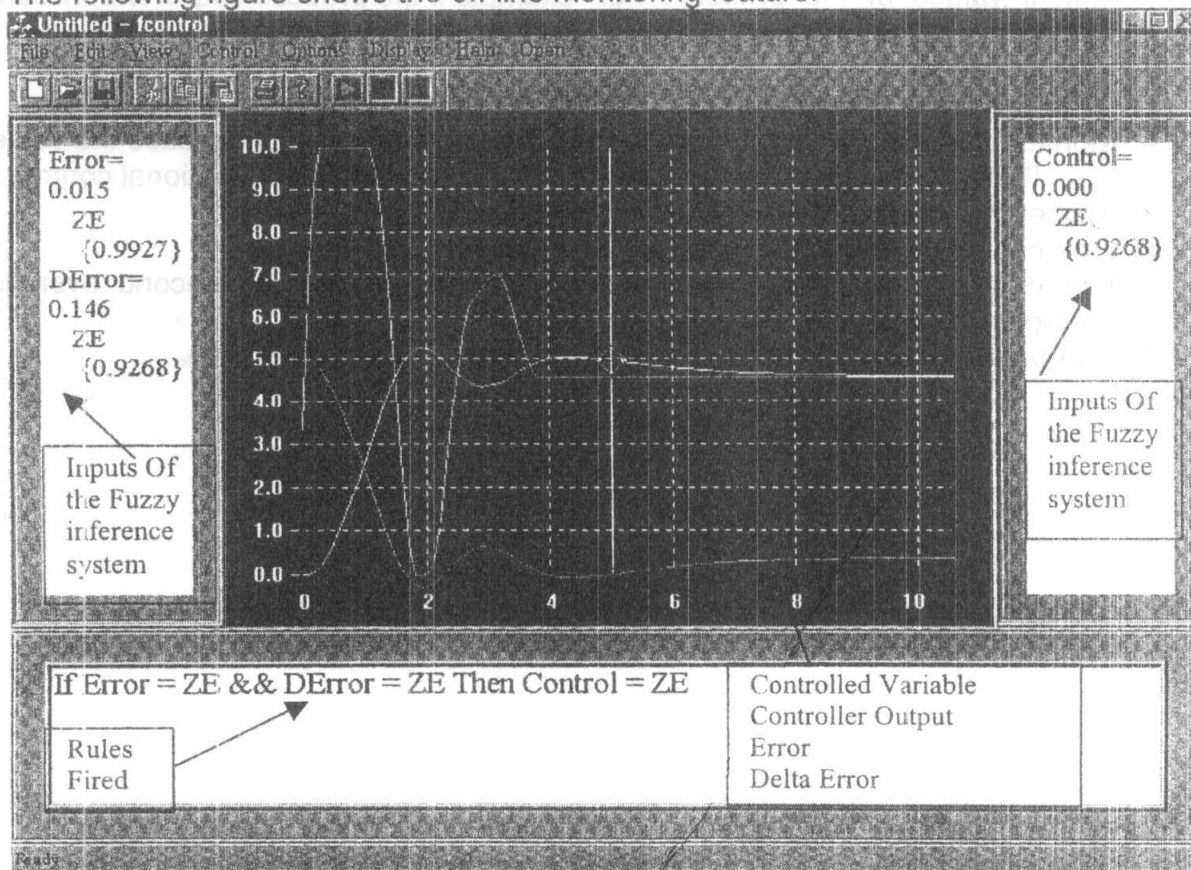


Fig 5

The power of the monitoring techniques is to associate the response of the closed loop to the fuzzy controller parameters. With these aids one can troubleshoot any unwanted performance, knowing which rules are fired at this time, what are the values of the scaling factors, what are the values of the fuzzification and defuzzification processes. One can detect the sort of error, correct it and try the closed loop again to justify the changes in performance due to the changes in controller parameters.

3.2. Systematic tuning approach

After many tuning experiments we state a reasonable approach as follows

1. Put the universe of discourse of error, derivative of error and control as their actual values of the process. For example if the process output is between 0 v and 10 v, set the universe of discourse of error to be -10 to 10 v
2. Set all scaling factors to be 1
3. If the system is unstable in the closed loop decrease the value of K_m .
4. Monitor the response in closed loop, if the response is slow increase the value of K_m . It has the effect of proportional control constant in conventional control. The higher the value of K_m the faster response you get.
5. One can adjust also the overshoot with varying K_m .
6. If the system still have its first overshoot lower than the second overshoot, decrease the value of K_{de} .
7. To obtain zero steady state error, change the values of K_e and K_{de} .

4. Experimental Results

4.1 Variation in K_m , K_e , K_{de} Scaling Factors

Process Description: We use a process containing a dead time of 1 sec and 2 lags each with a 1 second time constant.

Controller Description: We use a PI-Like fuzzy controller with the membership functions and rule base discussed in section 2

4.1.1. Effect Of K_{de} Variation

The variation in K_e yields to a significant variation of steady state error and type of oscillation. Decreasing K_e yields to more oscillatory system. This is shown in graph 1.

4.1.2. Effect Of K_m Variation

The experimental results shows that increase in K_m increases the rise time of the closed loop response. If K_m increases more, we reach instability. It affects also the overshoot of the closed loop response, if K_m increases the overshoot increases. It has an effect on the steady state error of the system. This is shown in graph 2.

4.1.3. Effect Of K_e Variation

The variation in K_e yields a significant variation of steady state error and type of oscillation. Decreasing K_e yields to more oscillatory system. This is shown in graph 3.

4.2. Best Tuning for different processes

4.2.1. Best Tuning for 2 lags and dead time process

The tuning parameter was $K_m = 0.6$, $K_e = 1$, $K_{de} = .95$ to get a good response with 20% overshoot, zero steady state error and 4 seconds rise time. This is shown in graph 4.

4.2.2. Best Tuning for a 2 lags process

We find that a good performance arises if the tuning was $K_m = 3$, $K_e = .68$, $K_{de} = 0.45$. The process is 2 lags process. This is shown in graph 5.

4.3. Robustness Test

A given process is tuned to get a good performance, then a change in the process is done to estimate the robustness of the controller.

4.3.1. Variation from a fast process to a slower one

The fast process was two lags process with each lag equals one second, the controller is tuned for this process. Then we get the performance of the loop if a slower process (two lags and dead time process) replace the old process. The new loop performance is poor and near instability. This means that a simple fuzzy logic controller is not robust against variation of process. This is shown in graph 6.

4.3.2. Variation from a slow process to a faster one

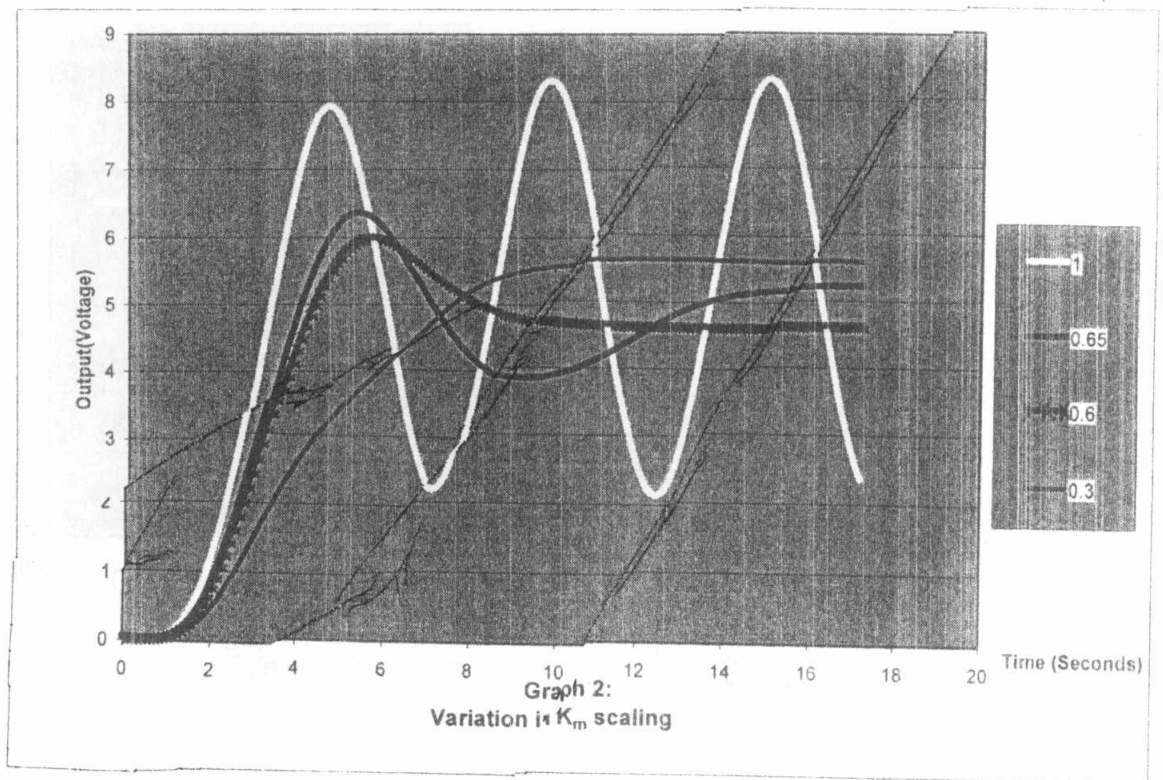
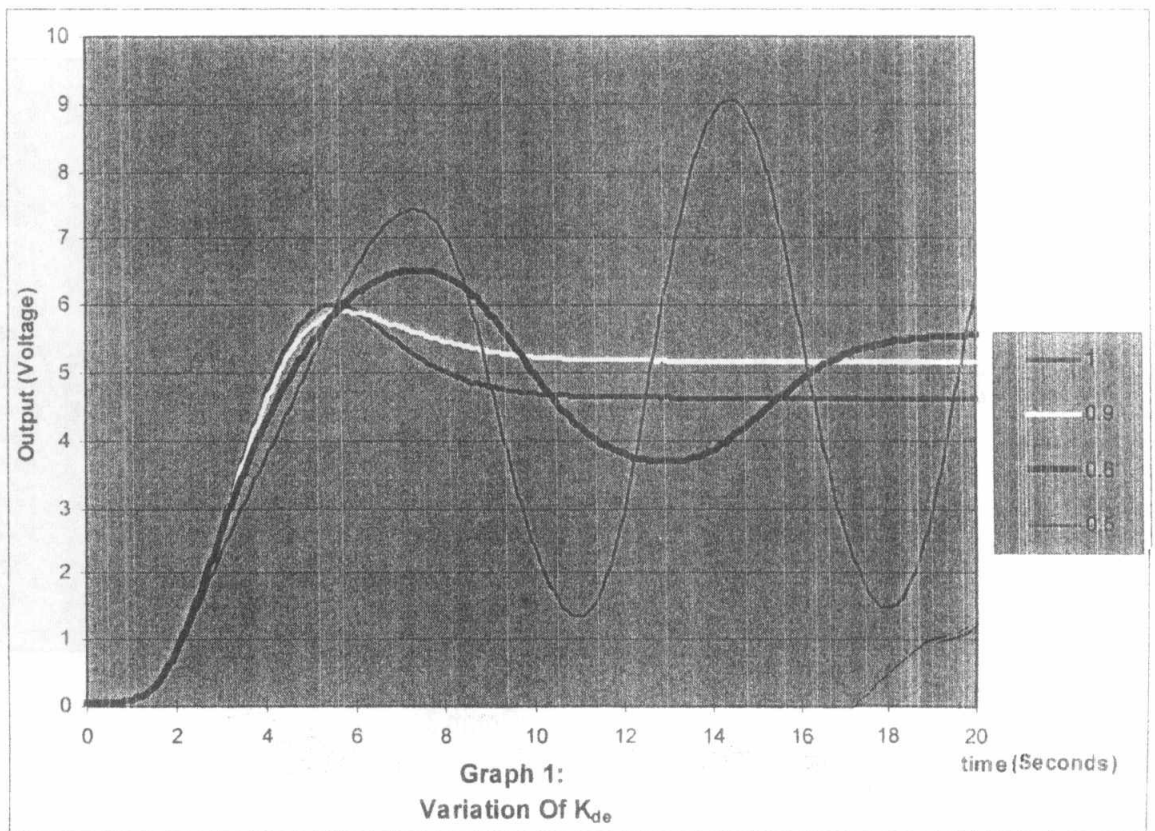
The slow process was two lags and dead time process with each lag equals one second, the controller is tuned for this process. Then we get the performance of the loop if a faster process (two lags process) replace the old process. The new loop performance is stable but settles little bit slowly. It means that fuzzy logic controller is not robust enough in this situation also. This is shown in graph 7.

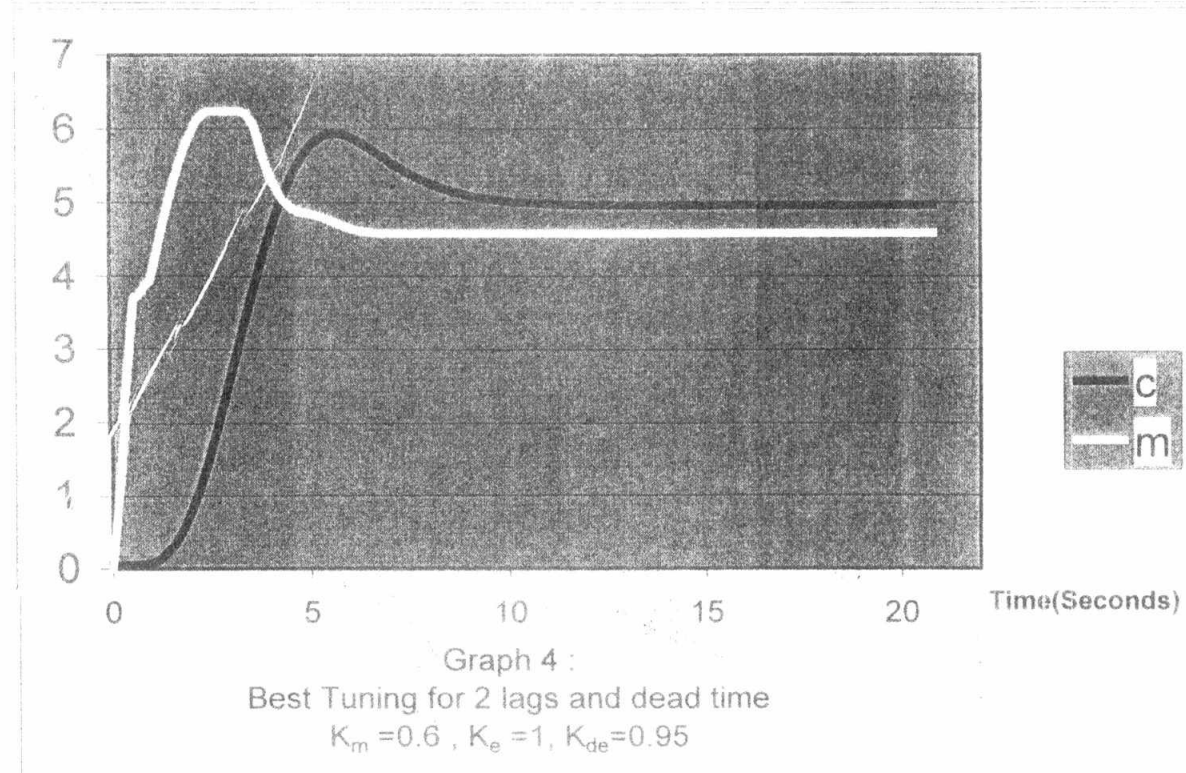
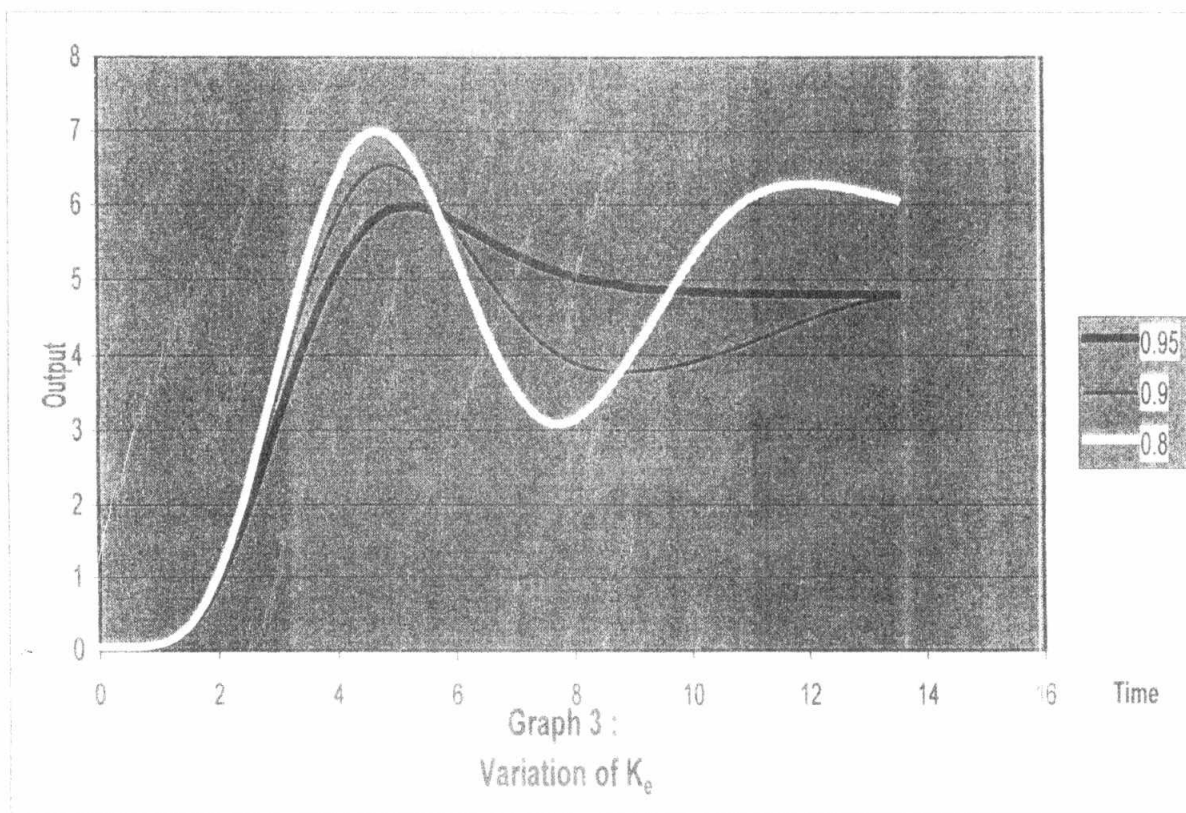
5. Conclusion

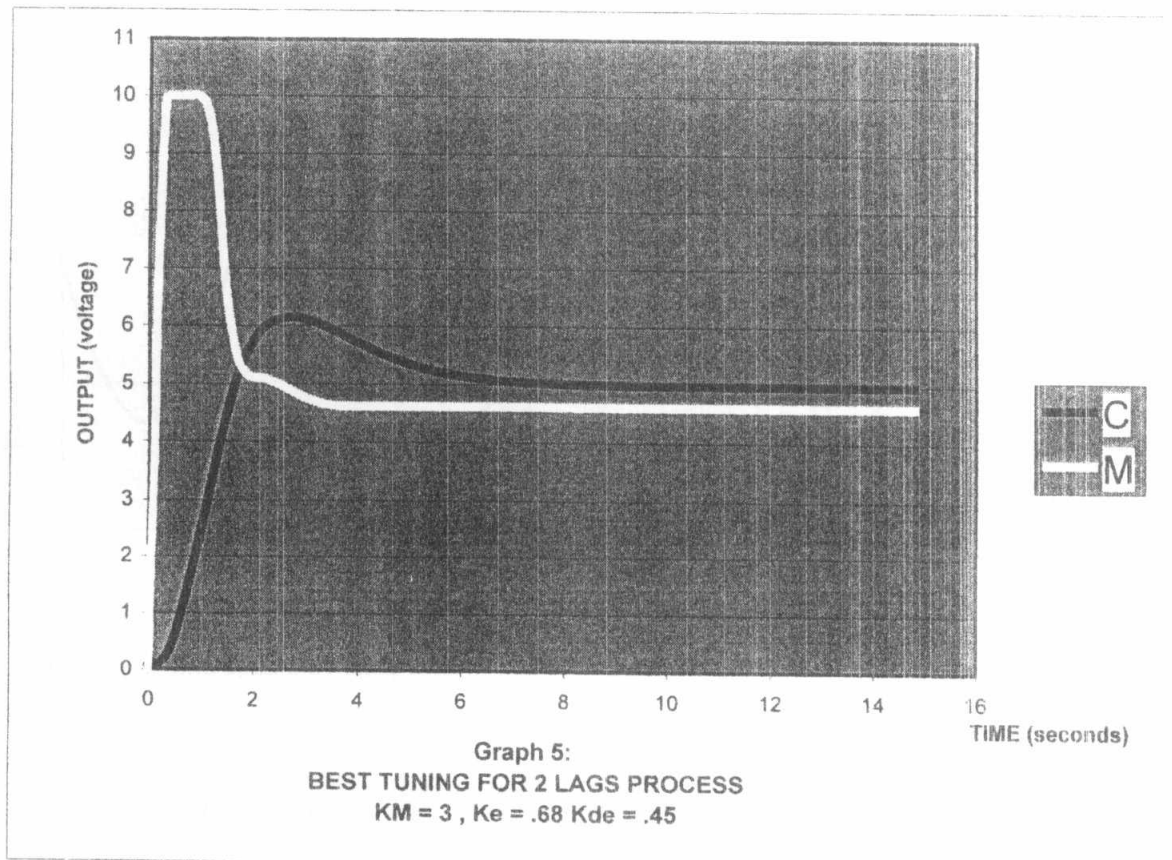
In this work, the PI and PD fuzzy controllers were made, and a tuning algorithm is proposed to facilitate the control engineer's tuning problem. A test is made to evaluate the fuzzy controller robustness against process variation. The experimental results show the power of the fuzzy toolbox and proposed algorithm. It shows also that fuzzy is not robust if the process variation is from a slow process to a faster one. Adaptation will be needed to overcome this disadvantage.

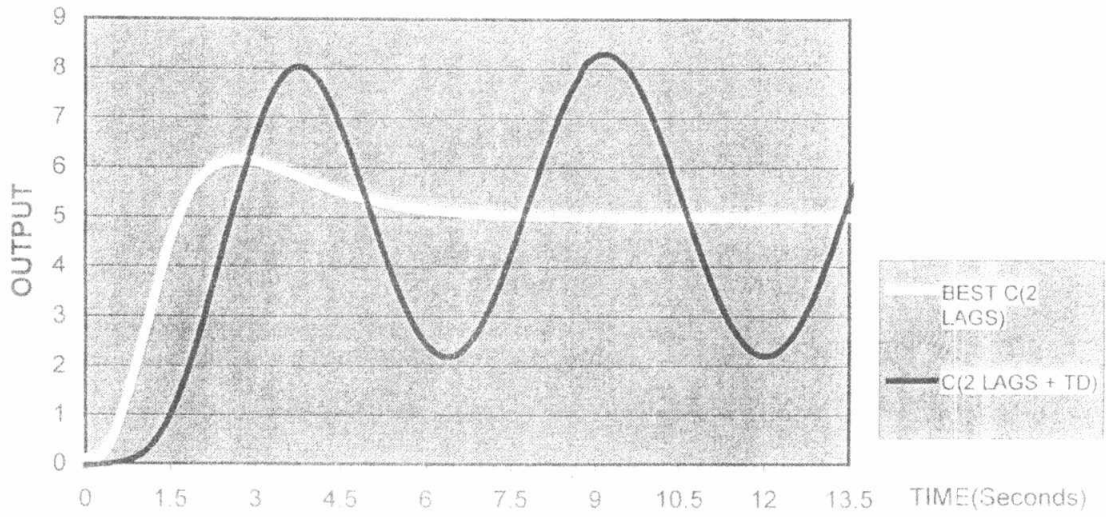
References

- [1] C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part 1", IEEE: TRANSACTIONS ON SYSTEMS vol. 20 pages 404-418, 1990
- [2] C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part 2", IEEE: TRANSACTIONS ON SYSTEMS vol. 20 pages 419-435, 1990
- [3] Wu Zhi Qiao, "PID type fuzzy controller and parameters adaptive method", Elsevier Science, 1996
- [4] "Matlab Fuzzy Toolbox Documentation", Mathworks Company, 1998.

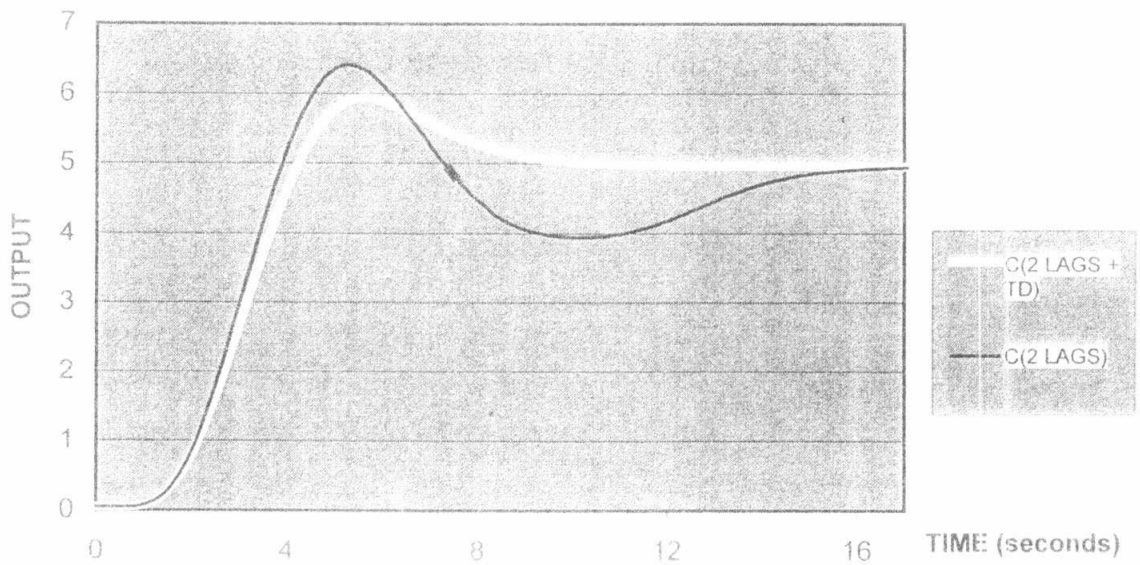








Graph 6:
Process Variation From 2 Lags To 2 Lags and Dead Time



Graph 7
Variation Of Process From 2 Lags and Dead Time To 2 Lags Only