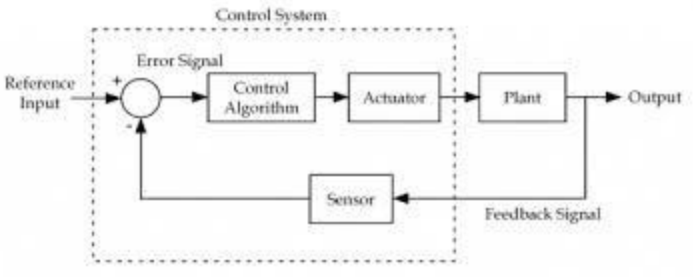
PID AND FUZZY CONTROLLER DESIGN

(c180069) SHRUTI SHARMA (BITS Pilani)

1. Introduction

From flights to toys, when we talk about an action or any response by a system, we want it to be in a given range and in accordance with our need, say if we are dealing with a flight control system, we want the plane to follow the instructed path irrespective of the disturbances caused by winds or clouds. This device which is used to maintain the desired behaviour is called controller and the part of the system is controlled is called plant. The output, of the controller, is referred to as a control variable, which is then supplied to an actuator, which changes the physical properties of the system in accordance with the controller output. In some cases, such as conventional traffic lights, we just need an input signal to output the control signal for each of the three colours at regular intervals such control systems are called open-loop control systems. However, think about complex systems, where there is a higher uncertainty in response, due to the presence of disturbances. In such cases we need a device which can measure the effect of controller output on the system and according to the amount of deviation, a controller can change the output to attain the predefined response. This type of control algorithm is referred to as a closed-loop feedback control. A simple feedback control system comprises of - a controller, actuator and sensor. Here, I am compiling my set of experiments done over this summers in Control Systems Laboratory, at Hiroshima University, which includes design and implementation of PID and a PID like fuzzy controller on actual systems.



2. PID controller design methods

2.1 What is PID control?

PID controller refers to a combination of proportional, integral and derivative control actions.

It consists of an SV – a set value which defines the set point for the output and provides a measure of the deviation from the reference value. The actual controller output is compared with the set point and this value or error is used to modify the output. In the case of a PID controller, we have three control actions-

Proportional control – Here controller output is changed proportionally to the error;

Integral control – Here the change in output is proportional to the integration of error, it is analogous to taking in account all the previous deviations of the control system for deciding the next output.

Derivative control – In this case, we adjust the control output according to the derivative of the error, or the change of error which helps to predict the next error value and influences the controller output accordingly.

PI-controller

When we have two actions, in our controller, one integral and other proportional, then the configuration is referred to as PI – controller design.

PID – controller

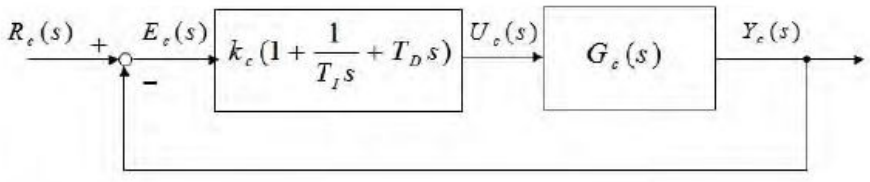
When a controller has all of proportional, integral and derivative actions, it is supposed to behave like a PID controller. The control law for a PID control in continuous form could be written as –

$$u_c(t) = k_c\{e_c(t) + \frac{1}{T_I}\int_0^t e_c(\tau)d\tau + T_D\frac{de_c(t)}{dt}\}.$$

where e_c indicates the control error as follows.

$$e_c(t) = r_c(t) - y_c(t).$$

Moreover, k_c is the proportional gain, T_I is the reset time, and T_D is the derivative ,time. The block diagram of the PID controller is shown as follows.



Block diagram of the PID control system.

The ratio of final output Y to that of initial reference R is called a transfer function,

$$W(s) = \frac{Y(s)}{R(s)}W(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$$

Where G = plant's mathematical model

C = control law for the controller

Here the denominator of W is calledthe characteristic equation and defines the stability of the system. In case of discrete time-domain, we can the control law could be written as –

$$u(k) = k_c\{e(k) + \frac{T_s}{T_I}\sum_{i=0}^{k} e(i) + T_D\frac{\Delta e(k)}{T_s}\}, \qquad \Delta u(k) = k_c\Delta e(k) + \frac{k_c \cdot T_s}{T_I}e(k) + \frac{k_c \cdot T_D}{T_s}\Delta^2 e(k)$$

where Ts = sampling time

2.2 PID gains tuning

2.2.1 PI parameter tuning for first-order systems based on a pole assignment control

Let us assume a simple first-order system, with the time constant – T and proportional gain K

$$G(s) = \frac{K}{1+Ts},$$

PID control law could be given as –

$$C(s) = \left(Kp_s + Kd_s s + \frac{Ki_s}{s}\right)E(s);$$

Where Kp_s = proportional gain Kd_s = derivative gain, Ki_s = integral gain

$$T(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$$

The denominator of transfer function = characteristic equation,

$$= KKi_s + (1 + KKp_s)s + (KKp_s + T)s^2);$$

Here, the roots of this equation are called poles and via modifying the roots of this equation we can tune the given system. The ideal transfer function for a first order system =

$$\frac{1}{1+sigma*.5s}$$

Comparing the denominator of the ideal transfer function and that of our characteristic equation and assuming $Kd_s = 0,$ (for computational simplicity), we get

$$Ki_s = \frac{4T}{Ksigma\hat{}2}$$

$$Kp_s = \frac{4T - sigma}{Ksigma}$$

2.2.2 PID parameter tuning for second order systems based on pole assignmentcontrol

Let us assume a second order system as –

$$G(s) = \frac{K}{(1+Ts)(1+Ls)}$$

where L and T are the two-time constants and K is the proportional gainand control law is given as -

$$C(s) = \left(Kp_s + Kd_s s + \frac{Ki_s}{s}\right) E(s);$$

$$T(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$$

The ideal transfer function for a second order system is given as -

$$\frac{1}{(1+sigma\hat{}3)}$$

Comparing the denominator of the ideal transfer function with a characteristic equation we get

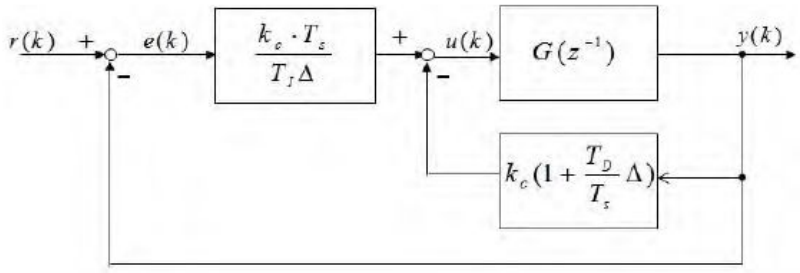
$$Kp_s = \frac{27LT - sigma\hat{}2}{Ksigma\hat{}2}$$

$$Ki_s = \frac{27LT}{sigma\hat{}3K}$$

$$Kd_s = \frac{9LT - sigma(L+T)}{Ksigma}$$

2.2.3 I-PD control law

Changing the structure of the controller and in place of adding the responses from derivative and controller allows us to decrease the overshoot, however, it may lead to a little increase in rising time.



Block diagram of I-PD control system.

I-PD control law could be written as –

$$\Delta u(k) = \frac{k_c \cdot T_s}{T_I} e(k) - k_c \Delta y(k) - \frac{k_c \cdot T_D}{T_s} \Delta^2 y(k)$$

Where delta^2 = change of error

2.3 Simulation

2.3.1 System description

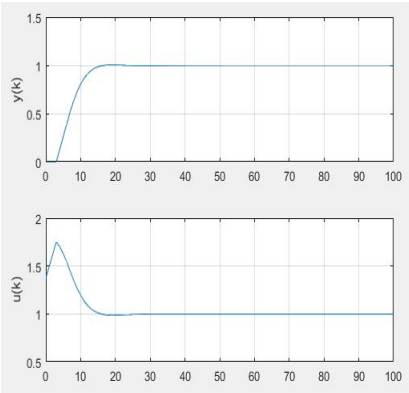
Here we use a first-order system given by with K = 1, T = 10, and then manually tune the controller to get a stable result.Next, we have a second-order system with time constants T = 10, L = 3 and same proportional gain K = 1, which can be thought as a first-order system with dead time L = 3 and which is approximated as a second order system using Pade's approximation.

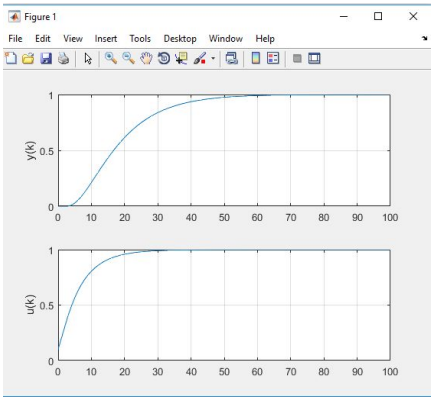
$$G(s) = \frac{K}{(1+Ts)(1+Ls)}$$

Where K =1, T = 10, L = 3.

2.3.2 Simulation results

After changing the values of sigma manually, we get stable results as shown -

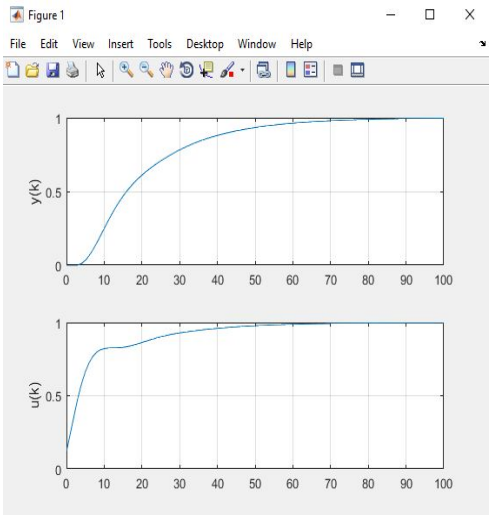


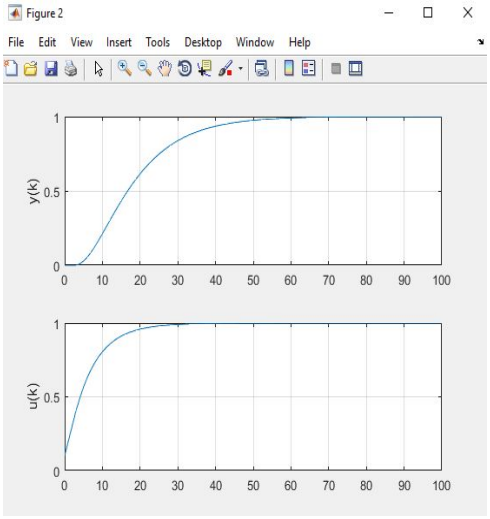


Case 1(sigma = 17.8)

Case 2(sigma = 20)

In case of I-PD configuration, the results as shown –





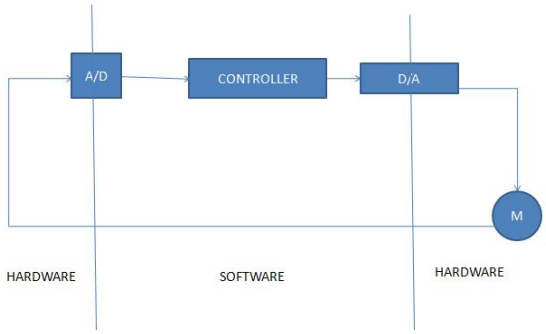
In case 1(sigma = 17.5)

In case 2 (sigma = 20)

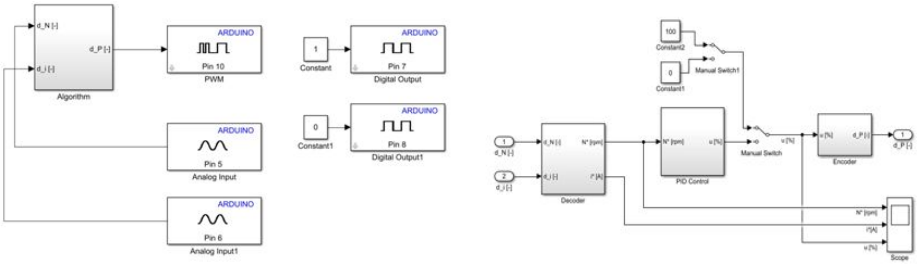
2.4Experimental results

2.4.1 **System description**

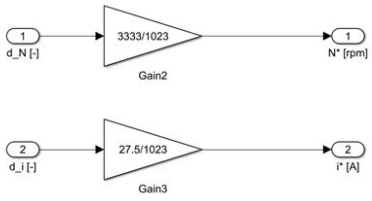
Here we have a DC motor control system, whose block diagram could be represented as-



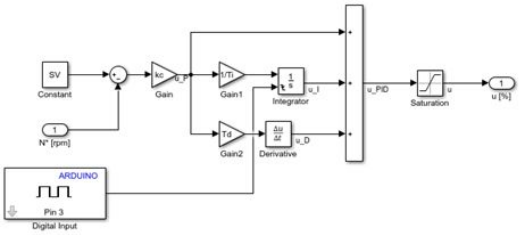
where M represents a motor; here via SIMULINK an experiment to identify the mathematical model of the given system is performed. The SIMULINK part of the system could be described as –



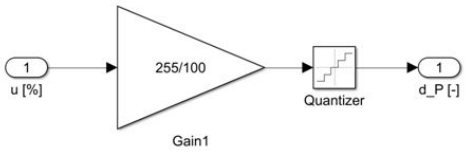
Algorithm



Encoder



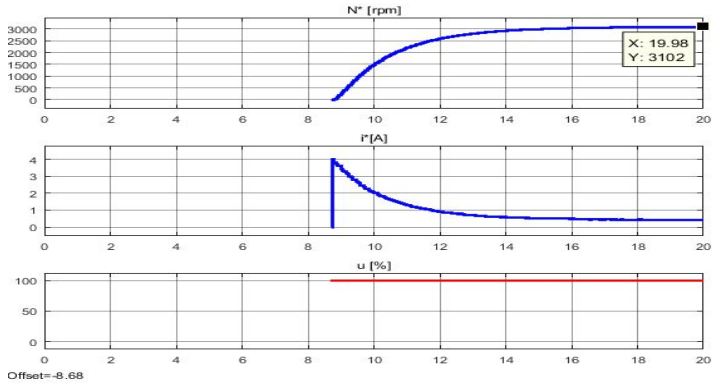
PID block



Decoder

2.4.2 Experiment description-

To identify the properties of the plant, or here DC motor system, we use the manual switch to study with the step response of the system, we conduct the experiment for five different End times; which are - 20s, 30s, 40s, 50s, 100s.



And now assuming the system to be a first-order system,

$$G(s) = \frac{K}{1+Ts}$$

Clearly, at s tending to zero or at G(s) approaching the steady state at time equal to infinity, we get G(s) = K, or here from above step response, K = y/r, at steady state ; and T= time when y = 0.632 K; L = time difference between the growth of input and output.

2.4.3Experimental results

Measuring these values for different time sets and averaging out we get –

L = 0;
T = 1.85;
K = 31.192;

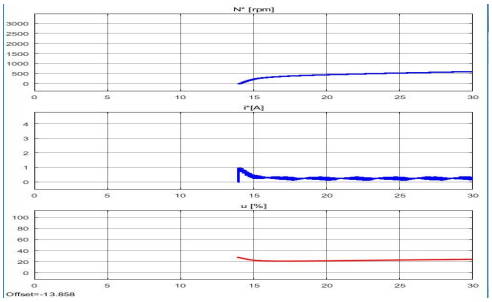
$$G(s) = \frac{31.192}{1+1.85s}$$

Using the pole placement method for a first order system, and assuming $Kd_s = 0,$ and

$$Ki_s = \frac{4T}{Ksigma\hat{}2}$$

$$Kp_s = \frac{4T - sigma}{Ksigma}$$

And changing the value of sigma to get a stable result we have, following results,



With PID parameters to be Ti = 1.82, Td = 0, with proportional gain = Kc = 0.0273

3. Fuzzy controller design methods

3.1 What is fuzzy control?

Fuzzy control involves controlling the output of a plant, via a set of linguistic rules, which are closer to that of the actual human working. The structure of a general fuzzy controller could be defined as –

Crisp input → fuzzification → Fuzzy processing → Defuzzification → crisp output

Fuzzification – refers to the process of conversion of a crisp set of values into a fuzzy set, via assignment of a degree to each of the inputs, this degree defines the extent to which an element belongs to that fuzzy set. This degree is defined as a membership degree. For example, let a crisp set be {0}, and corresponding fuzzy set will be = all the numbers which are close to zero each one with a degree proportional to its closeness to zero. Say {1/0.5 , 0/1}, here in a/b format, a denotes the element and the b denotes the membership degree of that element. So, say 0 belongs wholly to the given fuzzy set while 1 belongs to the given fuzzy set to a degree of 0.5.

Fuzzy processing – It consists of a set of If-Then rules, which are linguistically defined to identify the output when a set of fuzzified crisp values fire a set of rules.

Defuzzification – is the same process of conversion of the fuzzy outputs into a crisp output.

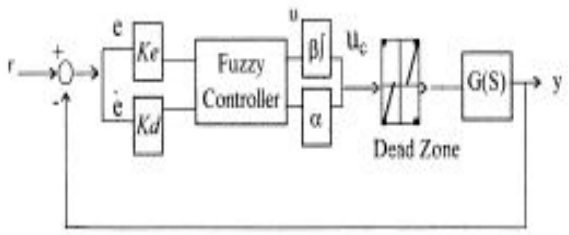
3.2PID like fuzzy controller design

For design SIMULINK and fuzzy logic app of MATLAB is used.

3.3PD like Fuzzy controller

Here we two crisp inputs to the fuzzy inference system, one is error and other is the change of error which resembles a PD-controller

3.3 PID like fuzzy controller



The output of the PID type fuzzy controller is

$$\begin{aligned} u_c &= \alpha u + \beta \int u\,dt \\ &= \alpha(A + PK_e e + DK_d \dot{e}) \\ &\quad + \beta \int (A + PK_e e + DK_d \dot{e})\,dt \\ &= \alpha A + \beta A t + (\alpha K_e P + \beta K_d D)e \\ &\quad + \beta K_e P \int e\,dt + \alpha K_d D\dot{e}. \end{aligned}$$

Proportional: $\alpha K_e P + \beta k_d D$
integral: $\beta k_e P$
derivative: $\alpha k_d D$

3.4 Adaptive PID like fuzzy controller

Here in this case, at each time instant Kd and beta are constantly modified as -

$$k_d = k_{ds}/\delta_k, \quad \beta = \delta_k * \beta_s,$$

where k_{ds} and β_s are the initial value of k_d and β, respectively. δ_k is the absolute peak value at the peak time t_k $(k = 1, 2, 3, \ldots)$.

3.5 **Simulation**

3.5.1 System description

Here we apply the PD type controller on the simple first order system with T=K=1, with a set of linguistic variables for the two input error and change of error and output u as –

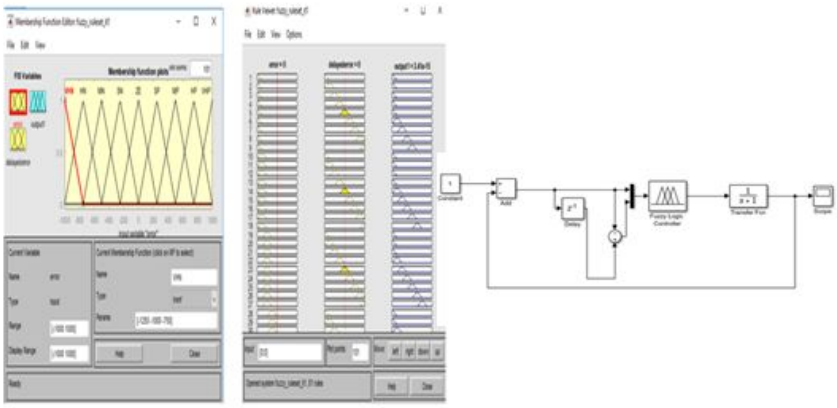
= {VHN, HN, MN, SN, ZE, SP, MP, HP, VHP}

= {Very high Negative, High Negative, Medium Negative, Small Negative, Zero, Small Positive, Medium Positive, High Positive, Very High Positive}, respectively.

The rules table for the processing of the rules could be defined as –

Δe / e	VHN	HN	MN	SN	ZE	SP	MP	HP	VHP
VHN	VHN	VHN	VHN	VHN	VHN	HN	MN	SN	ZE
HN	VHN	VHN	VHN	VHN	HN	MN	SN	ZE	SP
MN	VHN	VHN	VHN	HN	MN	SN	ZE	SP	MP
SN	VHN	VHN	HN	MN	SN	ZE	SP	MP	HP
ZE	VHN	HN	MN	SN	ZE	SP	MP	HP	VHP
SP	HN	MN	SN	ZE	SP	MP	HP	VHP	VHP
MP	MN	SN	ZE	SP	MP	HP	VHP	VHP	VHP
HP	SN	ZE	SP	MP	HP	VHP	VHP	VHP	VHP
VHP	ZE	SP	MP	HP	VHP	VHP	VHP	VHP	VHP

The SIMULINK model and the design of the fuzzy controller in Fuzzy Logic Toolbox for the system is given as –



Membership functions – equally spaced over a given interval

Rule set

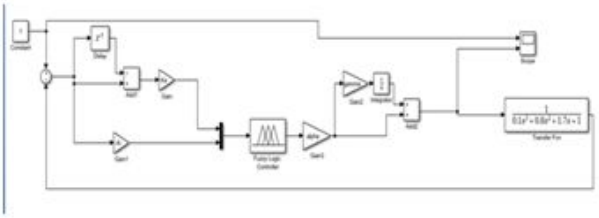
The second system is defined as a second-order system with a dead time of 0.2 seconds. Approximating this system using Pade's approximation we have –

$$G(s) = \frac{1}{0.1s^3+0.8s^2+1.7s+1}$$

With input variables having a linguistic set of variables as - { NL, NM, NS, ZR, PS, PM, PL }, == { Negative Large, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, Positive Large}

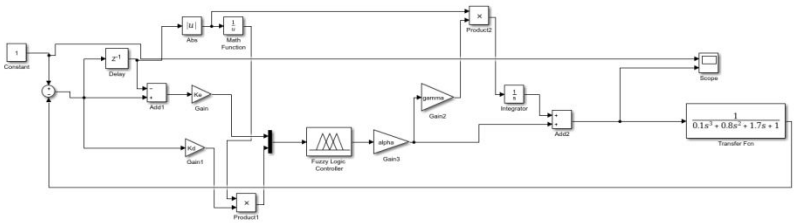
And the rule set and the SIMULINK model of the system is given as

E/E	NL	NM	NS	ZR	PS	PM	PL
PL	ZR	ps	pm	pl	pl	pl	pl
PM	ns	ZR	ps	pm	pl	pl	pl
PS	nm	ns	ZR	ps	pm	pl	pl
ZR	nl	nm	ns	ZR	ps	pm	pl
NS	nl	nl	nm	ns	ZR	ps	pm
NM	nl	nl	nl	nm	ns	ZR	ps
NL	nl	nl	nl	nl	nm	ns	ZR

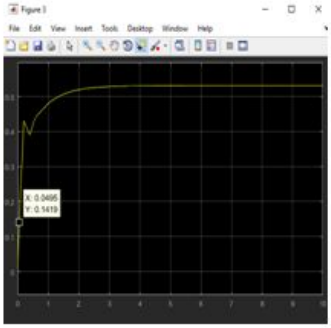


WhereKe = 1, Kd = 0.25, apha = 0.2; beta = 1, $gamma = \frac{beta}{alpha}$.

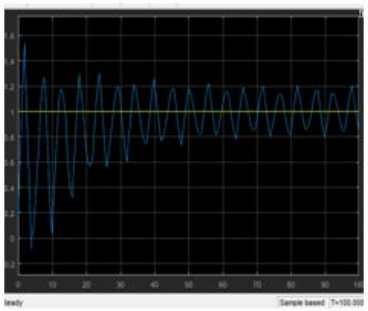
In case of parameter adaption method, SIMULINK model is given as –



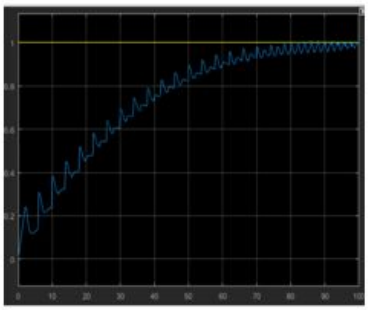
3.5.2 Results



PD –controller



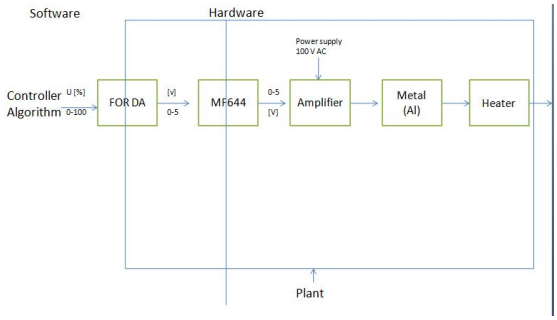
Simple PID-controller



PID-controller with adaptive parameters

3.6 Experimental results

3.6.1 System description- Heater control system –



Identifying the system parameters for the heater control system we get

K = 5, T = 41 and L = 4;

$$G(s) = \frac{5}{1+41s} e^{-4Ls}$$

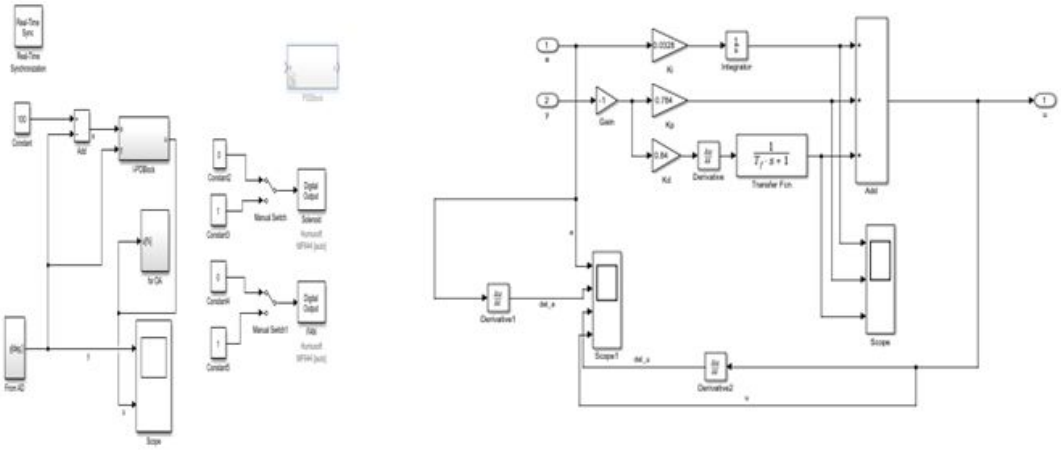
And tuning the system using pole placement method for a second order system we get,

$$Ki_S = 0.032,$$

$$Kp_s = 0.784,$$

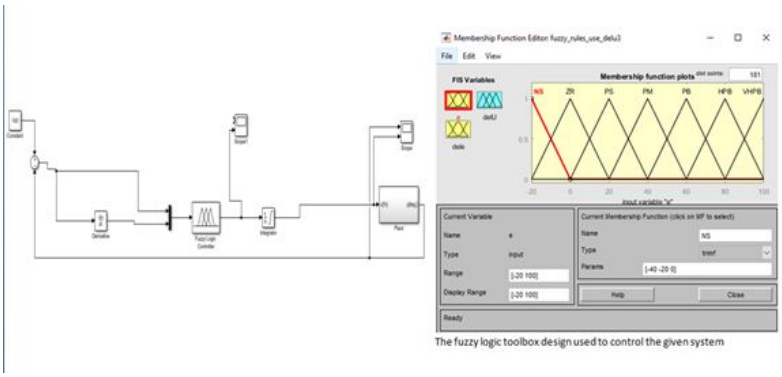
$$Kd_s = 0.84$$

Later, to reduce the overshoot I-PD configuration is chosen for better results. The SIMULINK model for I-PD configuration is given as-



I-PD block with a first-order filter

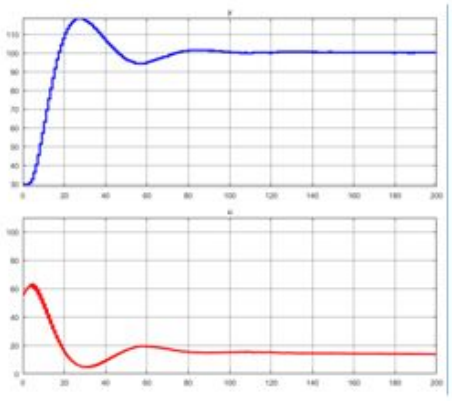
Finally, a simple fuzzy controller is designed using the simulation result of I-PD configuration. Its SIMULINK modeland a fuzzy controller is given as-



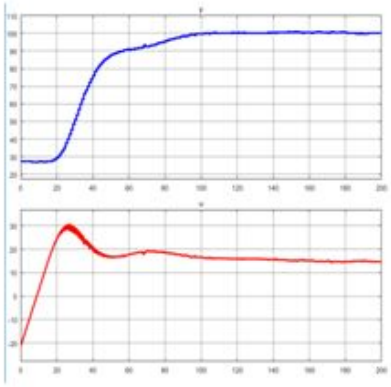
The fuzzy logic toolbox design used to control the given system

3.6.2Results

The final simulation results for the above-tuned systems are as –

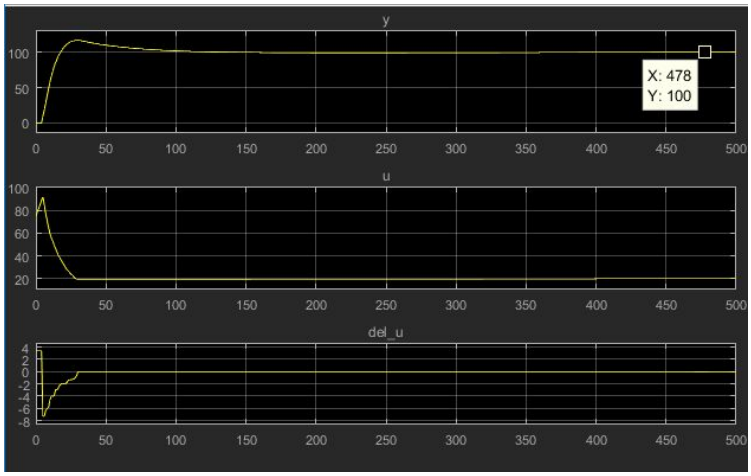


PID-controller output

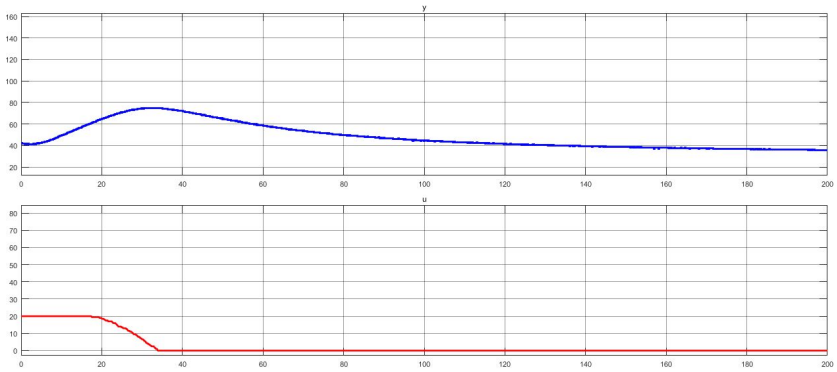


I-PD controller output

Simulation result of a fuzzy controller is given as-



However, while implementing the same controller on the actual heater control system, unsteady results with high steady-state error appear.



This was mainly due to the lack of proper inference mechanism and choice of proper membership function and lack of a proper algorithm for deriving and tuning of fuzzy inference mechanism and fuzzy rules.

4. Conclusion

I worked on the identification and design of a first-order and a second-order system, and then on designing of the PI and PID controller for the same systems. In case of a second-order system, which was a heater control system, I-PD configuration was more suited. For the same heater control system, I tried implementing the fuzzy controller, to further improve the rise time and overshoot. While in simulation, settling time decreased and overshoot was reduced, but I was not able to implement the results on actual experimental setup.

Future works -

Tuning and derivation of fuzzy inference mechanism for the fuzzy controller is an interesting field to work upon. Derivation of fuzzy rules and membership functions directly from the experimental data, obtained via an expert's knowledge or from an existing controller operating in similar conditions or via various experimentation setups could be an interesting field to work on. The tuning of the membership functions and rules can be done via training on the basis of obtained data via various learning techniques such as supervised or reinforcement learning.