World Academy of Science, Engineering and Technology
International Journal of Environmental and Ecological Engineering
Vol:11, No:2, 2017

Modelling for Temperature Non-Isothermal Continuous Stirred Tank Reactor Using Fuzzy Logic

Nasser Mohamed Ramli, Mohamad Syafiq Mohamad

Open Science Index, Environmental and Ecological Engineering Vol:11, No:2, 2017 publications.waset.org/10006421.pdf

***Abstract*—**Many types of controllers were applied on the continuous stirred tank reactor (CSTR) unit to control the temperature. In this research paper, Proportional-Integral-Derivative (PID) controller are compared with Fuzzy Logic controller for temperature control of CSTR. The control system for temperature non-isothermal of a CSTR will produce a stable response curve to its set point temperature. A mathematical model of a CSTR using the most general operating condition was developed through a set of differential equations into S-function using MATLAB. The reactor model and S-function are developed using m.file. After developing the S-function of CSTR model, User-Defined functions are used to link to SIMULINK file. Results that are obtained from simulation and temperature control were better when using Fuzzy logic control compared to PID control.

***Keywords*—**CSTR, temperature, PID, fuzzy logic.

I. INTRODUCTION

IN the industry of chemical processes, a reactor is the main basis of equipment in which the raw materials undergo a chemical reaction and change to form desired products [1]. The whole success of the industrial operation was due to the design and operation of chemical reactors. Reactors can be classified into various forms depending on the environment of the process, raw materials, and the products [2]. The understanding of non-steady behaviour of process equipment is necessary for the design and operation of automatic control systems. One main type of the process reactor is the continuous stirred tank reactor (CSTR). The CSTR is one example of many reactor designs that are used in chemical engineering and it is to be said as a common ideal reactor type [3].

Fuzzy logic is a type of approximate reasoning that uses the multi-valued logic. Fuzzy logic is commonly applied in machinery control. Theoretically, fuzzy logic uses an approach to computing based on partiality rather than the typical "true or false" Boolean logic which is the basis of the modern computer [4]. The best thing about fuzzy controlled systems model is that it does not involve any specific model for application of system [5]. The successful measurement of the fuzzy logic was based on approximate reasoning instead of modelling assumption which remarks the sturdiness of this method in real live application.

Fuzzy logic does not require mathematical modelling, so this makes it more flexible while facing complex non-linear problem [6].

II. CONTINUOUS STIRRED TANK REACTOR

CSTR basically includes a jacket or coil which is used to sustain and maintain the temperature of the reaction. A coolant stream is required to pass through the jacket coil to remove the excessive heat when an exothermic reaction takes place. On the contrary, if endothermic reaction arises in the system, a heating flow will be passed through the jacket. An isothermal reactor is a reactor which is operated at a constant temperature. It is required to develop the energy balance for the CSTR because the temperature tends to deviate with respect to time.

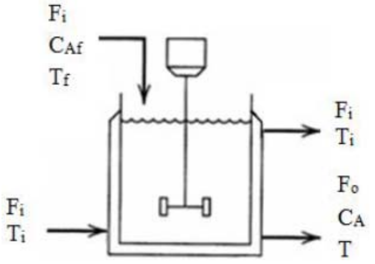


Fig. 1 Schematic representation of CSTR

We treat jacket temperature (Tj) as an input, while concentration (Ca) and temperature (T) are taken as our output. The mathematical model equations are obtained by the components of mass balance and energy balance principle in the reactor. Below, we can see that (1) is the temperature model of the CSTR:

$$\frac{dT}{dt} = \frac{F}{V}\left(T_f - T\right) - \frac{\Delta H}{\rho C_p} k_o \exp\left[\frac{E_a}{R(T+460)}\right] C_a - \left(\frac{UA}{\rho C_p V}\right)(T - T_j) \quad (1)$$

A. CSTR Modelling Using S-Function in MATLAB

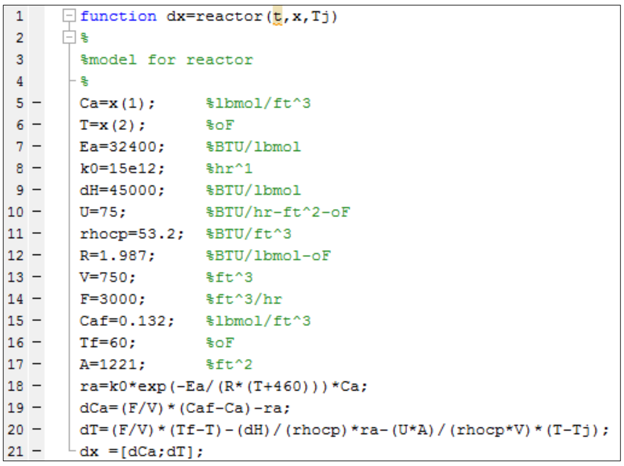
The process can be modelled by writing an m.file in MATLAB solvers, for example the ode45. Fig. 2 shows the coding which will be named as reactor.m. We express temperature jacket (Tj) as the manipulating parameter.

Then, an m.file S-function was written and was saved as m.file. It contains the rules in which Simulink can access information from MATLAB. In this case, we show the S-function file as in Fig. 3, and the file is saved as reactor_sfcn.m. This file is also saved as m-file.

Next is to use Simulink to add the S-Function block into the model browser in order to turn these coding into a block function. In the Simulink library browser, there is a group labelled as User-Defined Functions. Drag-drop the S-Function block and then double-click on the S-function block and fill in

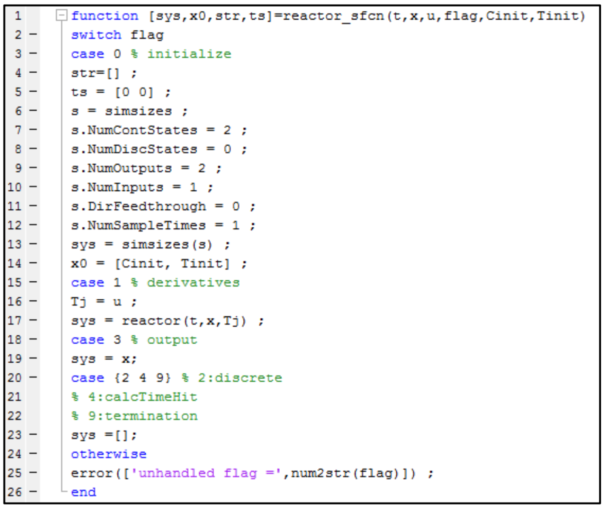
Elham Saraee is with the Boston University, United States (e-mail: esaraee@bu.edu).

the parameters. Change the S-function name to reactor_sfc and insert the parameters of the block.



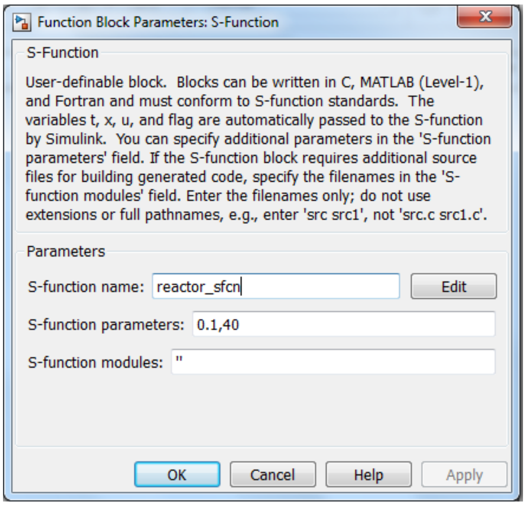
```
function dx=reactor(t,x,Tj)
%
%model for reactor
%
Ca=x(1);    %lbmol/ft^3
T=x(2);     %oF
Ea=32400;   %BTU/lbmol
k0=15e12;   %hr^1
dH=45000;   %BTU/lbmol
U=75;       %BTU/hr-ft^2-oF
rhocp=53.2; %BTU/ft^3
R=1.987;    %BTU/lbmol-oF
V=750;      %ft^3
F=3000;     %ft^3/hr
Caf=0.132;  %lbmol/ft^3
Tf=60;      %oF
A=1221;     %ft^2
ra=k0*exp(-Ea/(R*(T+460)))*Ca;
dCa=(F/V)*(Caf-Ca)-ra;
dT=(F/V)*(Tf-T)-(dH)/(rhocp)*ra-(U*A)/(rhocp*V)*(T-Tj);
dx =[dCa;dT];
```

Fig. 2 CSTR Model saved as reactor.m



```
function [sys,x0,str,ts]=reactor_sfcn(t,x,u,flag,Cinit,Tinit)
switch flag
case 0 % initialize
str=[] ;
ts = [0 0] ;
s = simsizes ;
s.NumContStates = 2 ;
s.NumDiscStates = 0 ;
s.NumOutputs = 2 ;
s.NumInputs = 1 ;
s.DirFeedthrough = 0 ;
s.NumSampleTimes = 1 ;
sys = simsizes(s) ;
x0 = [Cinit, Tinit] ;
case 1 % derivatives
Tj = u ;
sys = reactor(t,x,Tj) ;
case 3 % output
sys = x;
case {2 4 9} % 2:discrete
% 4:calcTimeHit
% 9:termination
sys =[];
otherwise
error(['unhandled flag =',num2str(flag)]) ;
end
```

Fig. 3 S-function file saved as reactor_sfcn.m



User-definable block. Blocks can be written in C, MATLAB (Level-1), and Fortran and must conform to S-function standards. The variables t, x, u, and flag are automatically passed to the S-function by Simulink. You can specify additional parameters in the 'S-function parameters' field. If the S-function block requires additional source files for building generated code, specify the filenames in the 'S-function modules' field. Enter the filenames only; do not use extensions or full pathnames, e.g., enter 'src src1', not 'src.c src1.c'.

S-function name: reactor_sfcn

S-function parameters: 0.1,40

S-function modules: ''

Fig. 4 S-Function block parameter editor

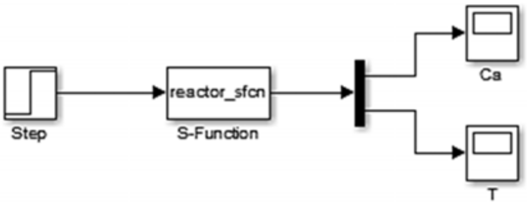


Fig. 5 Reactor_sfcn added with other Simulink blocks

B. PID Controller Design

An example of the simplest controller is the PID type which is commonly used in industries because of structure easiness in design and lower cost. PID means Proportional-Integral-Derivative, referring to the three terms operating on the error signal to produce a controlled signal. PID controller cannot yield an efficient control performance if control object is nonlinear. PID controller is a linear type and is the most-used feedback controller. PID controller constantly calculates error value as the difference between a measured process variable and a desired setpoint. The controller tries to reduce the error over time by alteration of the control variable.

$$u(t) = K_p e(t) + K_I \int_0^t e(t)dt + K_d \frac{de}{dt} \tag{2}$$

Kp, Ki, and Kd resemble the coefficient for the proportional, integral, and derivative terms. In the model, P denotes for the present values of the error. While I stands for the past values of the error, if the output is not satisfactory to reduce the size of the error, the control variable will accumulate over time, causing the controller to smear a stronger action and D stands for possible future values of the error, according its present rate of change [9].

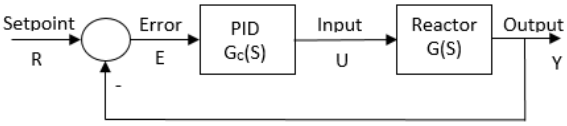


Fig. 6 Block diagram of PID controller with G(S) as reactor_sfcn

Using the trial and error method, the proportional action is the key control, while the integral and derivative actions improve it. The controller gain (Kc) is tuned with the integral and derivative actions detained at a minimum, until a desired output is reached [10]. The following tuning rule is present [11].

TABLE I
TRIAL AND ERROR METHOD TUNING RULE

Gains	Temperature process
K_p	2-10
K_i	2-10
K_d	0-5

During the simulation, the initial values of Kc, Ki, and Kd were introduced at 8, 3, and 1 was used as the initial basis of the trial and error in the PID controller. The values changed after the PID is tuned, and a controlled response was obtained in the simulation.

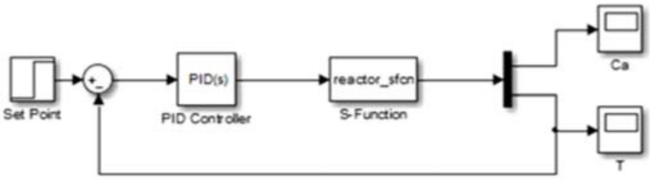


Fig. 7 System reactor_sfcn with Tuned PID feedback control in Simulink

III. FUZZY LOGIC CONTROLLER DESIGN

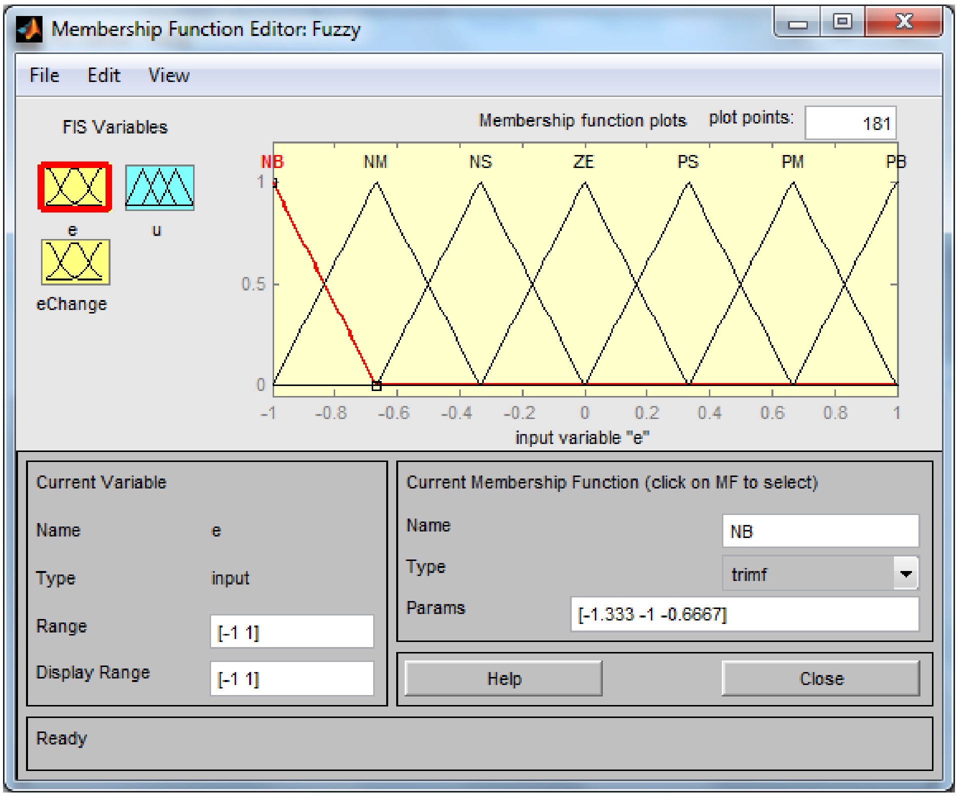
The first step to design the fuzzy logic control system is to decide on the rules that are going to be applied into the membership function in the fuzzy logic system. In this case, there are two inputs; control error is labelled as e, while the change in the control error is labelled as eChange. The output is the control action which is defined as u [7].

In Fuzzy Logic Controller for a non-linear CSTR plant, the set of rules is described in Fig. 8. These rules were added to the membership function in the FIS editor as a set of rules for the FLC. The word definition for the membership variables is outlined in Fig. 8 [8]. The interface of the fuzzy membership function editor is shown in Figs. 8 and 9, while Fig. 10 shows the Simulink model of the CSTR using FLC in the simulation. The simulation was run, and the data and reading were obtained respectively via the scope block which shows the temperature response of the system.

Controller Output u(t)		Error [e(t)]						
		NB	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
Error Change [Δe(t)]	**NB**	NB	NB	NB	NB	NM	NS	ZE
	NM	NB	NM	NM	NM	NS	ZE	PS
	NS	NB	NM	NS	NS	ZE	PS	PM
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NM	NS	ZE	PS	PS	PM	PB
	PM	NS	ZE	PS	PM	PM	PM	PB
	PB	ZE	PS	PM	PB	PB	PB	PB

NB	Negative Big	**PS**	Positive Small
NM	Negative Medium	**PM**	Positive Medium
NS	Negative Small	**PB**	Positive Big
ZE	Zero		

Fig. 8 Interval Rules for Fuzzy Logic Controller and variables used in the membership function



plot points: 181

Name e

Type input

Range [-1 1]

Display Range [-1 1]

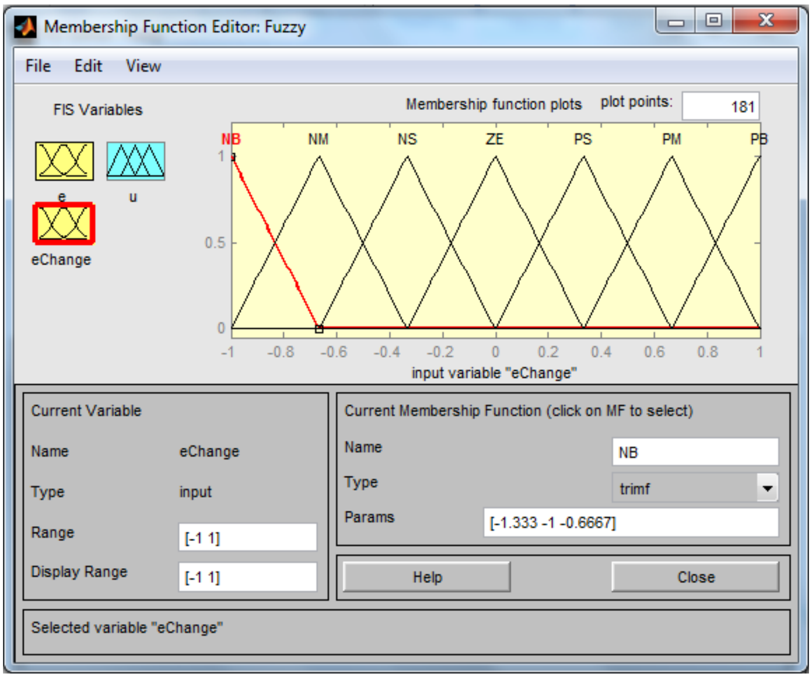
Name NB

Type trimf

Params [-1.333 -1 -0.6667]

Ready

Fig. 9 Membership function for error (input 1)



plot points: 181

Current Variable

Name eChange

Type input

Range [-1 1]

Display Range [-1 1]

Current Membership Function (click on MF to select)

Name NB

Type trimf

Params [-1.333 -1 -0.6667]

Selected variable "eChange"

Fig. 10 Membership function for error change (input 2)

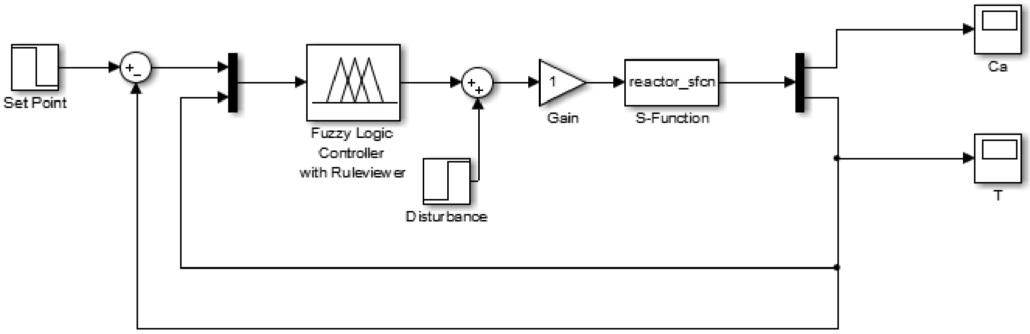


Fig. 11 System reactor_sfcn with FLC in Simulink

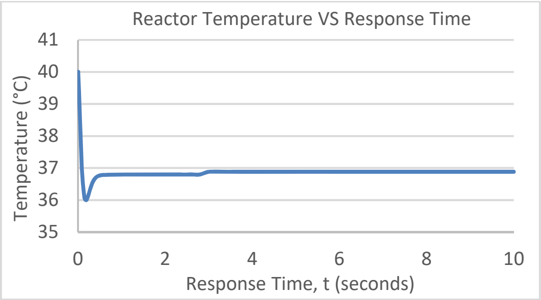


Fig. 12 Temperature vs response time uncontrolled system

IV. RESULTS & DISCUSSION

The temperature response that we obtained before applying any control system design can be seen in Fig. 12 below which it is taken from the scope of the block. In Fig. 12, we can see that there is one peak that overshoots the valued set point. Initially, the temperature was at 40 °C and was set to reach a temperature of 36.7 °C. The final output temperature shown was 37 °C which varies from the set point value that needed to be attained.

In the PID controlled CSTR system, the PID tuning coefficient is automatically generated using the tuner in the Function Block Parameters: PID Controller menu. The original coefficient that was input into the PID controller was Kc = 8, Ki = 3, and Kd = 1. After tuning, the coefficient values that were achieved were changed to Kc = 4.11 Ki = 50.38 and Kd = -0.12. In Fig. 13, we can see the temperature response of the CSTR system when applied PID tuned controller.

Initially, the temperature falls quickly and settles in a split second to an intermediate temperature value which was at T=37.5 °C. When the response time reaches 1 second, it drops further to the set point value which is at T=36.7 °C. Comparing the PID controlled system with the uncontrolled CSTR system, we see that, in a PID controlled environment, there is no peak overshoot in the temperature response. In Fig.14, we can see the PID step behaviour of the original

CSTR response and the tuned response, while Fig. 14 shows the bode response both for block and tuned.

In the FLC design of the CSTR system, we can see from Fig. 16 that the output temperature response of the reactor function is much more stable compared to the PID controlled system. We can observe that there is no peak overshooting, while the setpoint temperature T=36.8 ºC value was reached in a smooth and steady manner. The time taken for the initial temperature to reach the setpoint value is approximately below 1 seconds, while the temperature does not rest at any other temperature value.

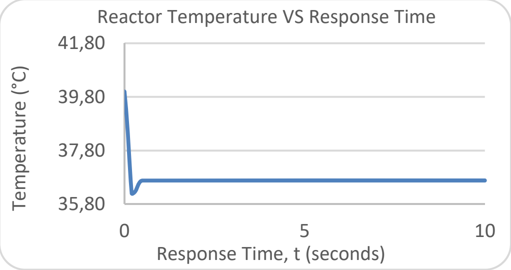


Fig. 13 Temperature vs response time for PID controlled system

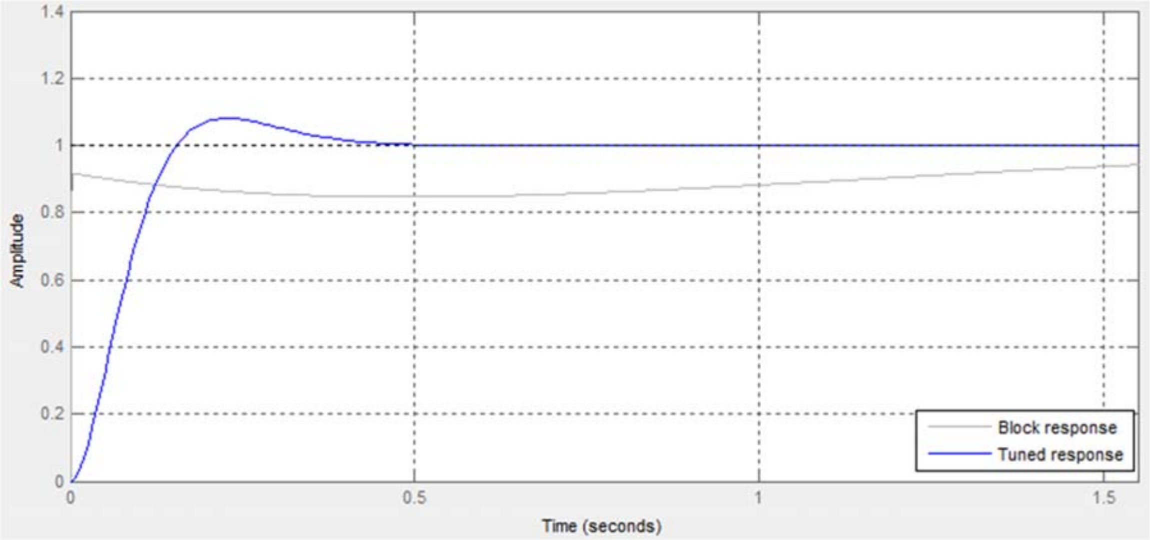


Fig. 14 PID step amplitude against response time behaviour for block and tuned

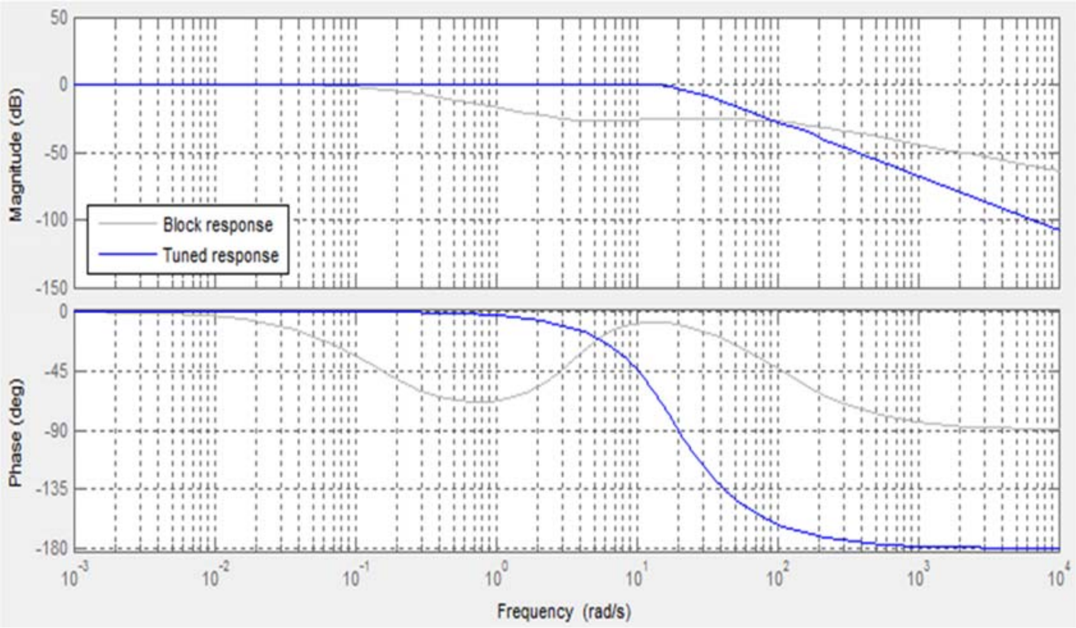


Fig. 15 Bode Diagram for the PID control phase and magnitude against frequency

Comparing the PID and the FLC methods of controlling the non-isothermal CSTR model, we can observe that the FLC is better at controlling a non-isothermal system compared to the PID control system although that the tuning method of PID is very simple compared to the 49 rules relating membership functions of the Mamdani FLC design which finally gives the desired and quickest response of a controlling parameters.

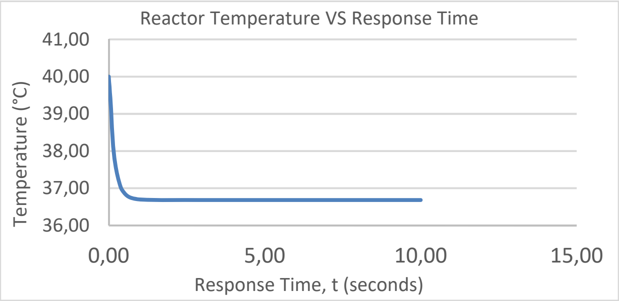


Fig. 16 Temperature vs response time for FLC CSTR system

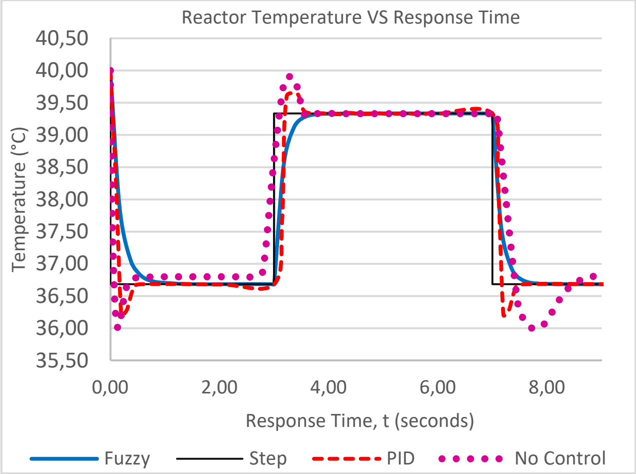


Fig. 17 Temperature vs response time for FLC, Step and PID CSTR system pulse generator

TABLE II
COMPARISON BETWEEN ALL TYPES OF CONTROLLERS

Controller Types	Plot Behavior		
	Maximum Peak Temperature, °C	Settling Time (seconds)	Steady State Value
Uncontrolled System	36	0.78	36.79
PID Controlled System	36.21	0.5	36.68
Fuzzy Controlled System	36.68	1.15	36.68

Similarly, the FLC gives the best performance in terms of steady state deviation and peak overshoots, but when comparing the settling time, the FLC did not do well comparing to the PID System

V. CONCLUSION

This paper presents the method to develop a control system for a non-isothermal process in a CSTR which is theoretically hard to be controlled by the conventional PID controllers. The outcome of this research was to design a block diagram control system of a non-isothermal temperature of a CSTR in order to produce a stable response curve by comparing it with PID controller and FLC.

This CSTR design was not inserted into transfer function. Instead, the application of S-Function block makes the modelling of a non-isothermal CSTR easier and has lesser time consumption compared to transfer function modelling which involves many mathematical equations and can be difficult to achieve an accurate model. The S-Function model was then later included in the Simulink block editor to be included with other block functions.

In the PID controller design, it was initially specified that the tuning parameter was going to be achieved through trial and error method with a specific range of values to be applied and finally comes up with an initial guess of the parameter which are Kc = 8 Ki = 3 and Kd = 1. After tuning, the coefficient values that were achieved were changed to Kc = 4.11 Ki = 50.38 and Kd = -0.12. The results show that PID controller gives 0% peak overshoots but gives a response delay which is approximately equal to 1 second until it reaches the desired temperature.

For the FLC design, the defined type of FLC that was used in this project was the Mamdani type FLC. We used one output variable and two input variables where each of these variables was given seven membership function labelled as in Table II. The results of the FLC CSTR simulation are that there were also 0% peak overshoots, and also there is no time delay for the temperature system. For CSTR system, the most required criterion is that the system has no overshoot and zero steady-state error. Between these controllers, a comparison has been done to see which controller can meet the criterion. From the result and discussion section, the two controllers successfully designed were compared. Based on the results, we conclude that the FLC shows the best performance because it has zero steady-state error and takes the shortest time response.

REFERENCES

[1] M. Araki, "PID Control," Encyclopedia of Life Support Systems, vol. II, pp. 1-7.

[2] S. Boobalan, K. Prabhu and V. M. Bhaskaran, "Fuzzy Based Temperature Controller For Continuous Stirred Tank Reactor," International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, vol. II, no. 12, 2013.

[3] S. Deepa, N. Anipriya and R. Subbulakshmy, "Design of Controllers for Continuous Stirred Tank Reactor," International Journal of Power Electronics and Drive System, vol. V, no. 4, pp. 576-582, 2014.

[4] A. Farhad and K. Gagandeep, "Comparative Analysis of Conventional, P, PI, PID and Fuzzy Logic Controllers for the Efficient Control of Concentration in CSTR," International Journal of Computer Applications, pp. 12-16, 2011.

[5] B. Maurya and S. Bajpai, "Fuzzy Logic Based Temperature Control of Continuous Stirred Tank Reactor," International Journal of Engineering Trends and Technology.

[6] P. Poongodi and R. Madhu Sudhanan, "Simulation of temperature control methodologies for chemical reactor," Journal of Chemical and Pharmaceutical Research, vol. 7, no. 9, pp. 682-689, 2015.

[7] S. Shahin and M. Shahrokhi, "Adaptive fuzzy backstepping approach for temperature control of continuous stirred tank reactors," Fuzzy Sets and Systems, vol. 160, no. 12, pp. 1804-1818, 2009.

[8] N. Singh and S. Kumar, "Comparative Analysis Of PID, Cascade and Fuzzy Logic Control For the Efficient Temperature control in CSTR," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, pp. 11-16, 2012.

[9] S. Vaneshani and H. Jazayeri-Rad, "Optimized Fuzzy Control by Particle Swarm Optimization Technique for Control of CSTR," World Academy of Science, Engineering and Technology, vol. 5, pp. 405-410, 2011.

[10] D. F. Ahed and M. N. Esmaeel, "Fuzzy logic Control of Continuous Stirred Tank Reactor," Tikrit Journal of Engineering S ciences, vol. 20, no. 2, pp. 70-80, 2013.
[11] P. Woolf "Chemical Process Dynamic and Controls," open online textbook University of Michigan Chemical Engineering.