DEVELOPMENT AND IMPLEMENTATION OF ADAPTIVE FUZZY PID CONTROLLER (AFPIDC) FOR FLOW CONTROL APPLICATION

By

ZULFADHLI BIN MAZLAN
11028

Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

SEPTEMBER 2011

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DEVELOPMENT AND IMPLEMENTATION OF ADAPTIVE FUZZY PID CONTROLLER (AFPIDC) FOR FLOW CONTROL APPLICATION

by

ZULFADHLI BIN MAZLAN

FINAL DISSERTATION

Submitted to the

ELECTRICAL & ELECTRONICS ENGINEERING PROGRAMME

Universiti Teknologi PETRONAS

in partial fulfillment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,

____

Dr. Rosdiazli Ibrahim

Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

SEPTEMBER 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the reference and acknowledgements and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ZULFADHLI BIN MAZLAN

ABSTRACT

In general, this project aims to enhance the capability of conventional PID controller by designing and implementing the Adaptive Fuzzy Logic PID Controller (AFPIDC) and compare its performance with the conventional Fuzzy Logic Controller (FLC) for Flow Control Application in the process plant. The implementation has been done onto the *PcA SimExpert Mobile Pilot Plant SE231B-21 Flow Control and Calibration Process Unit.* This mobile plant consists of several flow measurements which are Orifice, Coriolis and Vortex Flow Meter. Currently, controlling and tuning is done via *KONICS* PID controller that is mounted on the local control panel. However, current PID controller does not provide faster response and need to be manually tuned. Thus, the AFPIDC will be developed and implemented to compare with the existing PID controller and to design a DCS-HMI interface using MATLAB/Simulink for this pilot plant. The required hardware tools for this project will be USB-1208 FS Personal Measurement Device and MATLAB product family. The Fuzzy Inference System (FIS) are developed using Mamdani Approach. This involves designing and tuning of the membership functions, input/output rules and the de-fuzzification technique. Fuzzy Logic reasoning is used to produce adaptive PID gain that will be added up with the initial PID gain. Overall, the control performances of each controller (PID, FLC and AFPIDC) will be compared and analyzed for flow control application.

ACKNOWLEDGEMENTS

First and foremost, my deepest gratitude goes to Dr. Rosdiazli Ibrahim, Final Year Project Supervisor and a mentor. He has provided continuous guidance and support throughout the progress of this project over the past year. He has always give new ideas and set a high standard for this project in order to produce one of the quality projects.

Secondly, special thanks to Mr Azhar Zainal Abidin, Intstrument Laboratory Technician, for his assistance and understanding of the operation for the pilot plant. His support in giving the flexibility time and opportunity to use the pilot plant in any time of the day has helped to complete this project. His effort in troubleshooting problems related to the pilot plant is also highly appreciated and acknowledged.

Last but not least, special thanks also to the friends, family and others who were directly and indirectly involved in making this final year projects a success.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

PID	Proportional Integral Derivative
PI	Proportional + Integral
PD	Proportional + Derivative
FLC	Fuzzy Logic Controller
ANN	Artificial Neural Network
AFPIDC	Adaptive Fuzzy PID Controller
FIS	Fuzzy Inference System
DAQ	Data Acquisition
ANFIS	Adaptive Neural Fuzzy Inference System
FODT	First Order with Dead Time
MF	Membership Function
CoA	Center of Gravity
DCS	Distribution Control System
HMI	Human Machine Interface
PRC	Process Reaction Curve

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Control refers to those things that maintain a desired system output by altering the flow of energy from the source to the output device. So, control will give influences to the final outcome of a process or operation. The fundamental of any control system is the ability to measure the output of the system and take any necessary action for the correction if the measured values deviate with the desired values. Automation or feedback control's history can be traced back to 1769 where James Watt designed a flyball or centrifugal speed governor to control the speed rotation of the steam engine without human intervention [1].

Generally, a feedback control system will have several important elements such as controller, final element (control valve), plant or process and sensors [2]. Each of these elements will determine the performance and response of the process control. Good controller will examine the measured values received from the sensor and compare it with the set desired values from the user. Then, it will determine the necessary amount of action needed to be taken in order to reduce the error of the control system into acceptable range. So, for stability and fast response of feedback control system, controller plays an important role in the feedback control system.

Flow control is an important application in certain industrial application with the aims to:

- Regulate amount of feed or recycle product into reactor tank.
- Regulate the amount of heating fluid medium into the shell and tube type heat exchanger in order to control the temperature of the product flow through heat exchanger
- Regulate level of a tank by controlling the flow in and flow out of the product.

1.2 Problem Statement

Flow measurement in the process control is crucial for any industrial process operation. Normally, flow rate measurement in the industrial are different for liquid and gas but both of them can be measured in volumetric (liters per second) or mass flow rates (kilograms per second). For liquid flow measurement, the volume flow rates can be converted easily to mass flow rates if given the fluid density. The liquid density is almost the same for all liquid conditions. However, for gas flow measurement, the pressure, temperature and gas composition has to be taken into account because it gives different gas density. So, with many variables that can affect the flow rate (disturbances), it is crucial to have a robust flow controller to optimize the performance of the process control. So, in this project, the Adaptive Fuzzy Logic PID Controller (AFPIDC) is proposed to be implemented for the flow control in the pilot plant to replace conventional PID Controller due to its superior applicability and robustness [3]. Besides that, the current PID controller is mounted on the Local Control Panel and there is no Human Machine Interface (HMI) on the local panel where the data trend can be viewed. So, with the AFPIDC, it can be accessed from a remote PC through MATLAB/Simulink for easy monitoring, control and tuning.

1.3 Objectives and Scope of study

The objectives of this project are:

- To investigate, design and develop Adaptive Fuzzy Logic PID Controller (AFPIDC) for flow control of the pilot plant.
- To implement AFPIDC and obtain data trend by using DAQ card, remote PC and MATLAB/Simulink.
- To analyze and compare the controller between the conventional PID controller, FLC and AFPIDC.

The scope of study for the new controller is to be used on *PcA SimExpert Mobile Pilot Plant SE231B-21 Flow Control and Calibration Process Unit.* The controller will be implemented on the pilot plant via USB-1208 FS Personal Measurement Device for DAQ with the remote PC. Besides that, the controller design and data trend will be done and shown in MATLAB/Simulink on the remote PC.

Figure 1 : PcA SimExpert Mobile Pilot Plant SE231B-21 Flow Control & Calibration Process Unit

1.4 Relevancy, Feasibility and Significance of the project

By using MATLAB/Simulink software, the design and development of AFPIDC can be done and act as the alternative HMI for the control system of the pilot plant. Besides that, the remote PC and USB-1208 FS Personal Measurement Device will be connected to pilot plant for implementation of the designed controller. The significance of this project is the pilot plant has been fabricated according to real world industrial standards and conditions. So, the input- output and uncertainties (disturbances) is well captured for the flow control of the pilot plant. Hence, the implemented controller can be used and applied for the industrial process control. Last but not least, this project has the feasibility to be completed within the time frame and scope of work.

CHAPTER 2
LITERATURE REVIEW

2.1 Conventional Controller

For the conventional feedback controllers, there are basically two types of automatic controller for every process control system which is discontinuous and continuous control. Discontinuous automatic controller involved an on-off switch operation (relay). However, the problems with on-off process control are wear in the controls, sudden changes causing damage to plant and low accuracy [4].

For the continuous automatic controller, the most common controller would be the Proportional (P), Integral (I) and Derivative (D) controller. The method for determining the appropriate settings for the PID controller was initially introduced by the J.G Ziegler and N.B Nichols [5]. The PID controller algorithm involves three separate constant parameters i.e.K_p for proportional (P), K_i for integral (I) and K_d for derivative (D) which also sometimes called three term control.

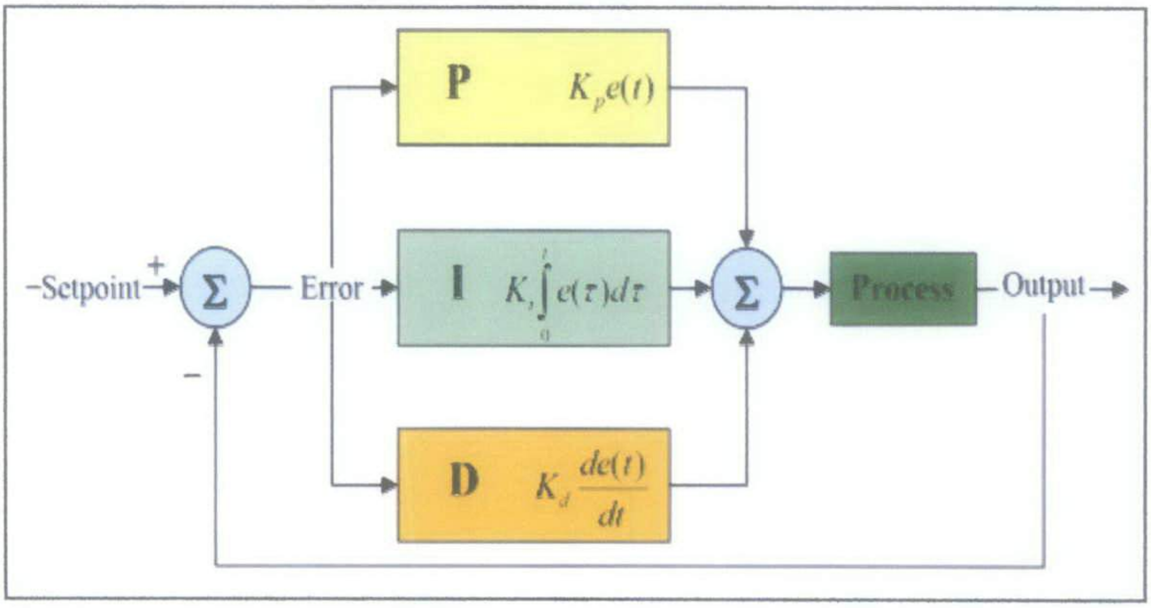


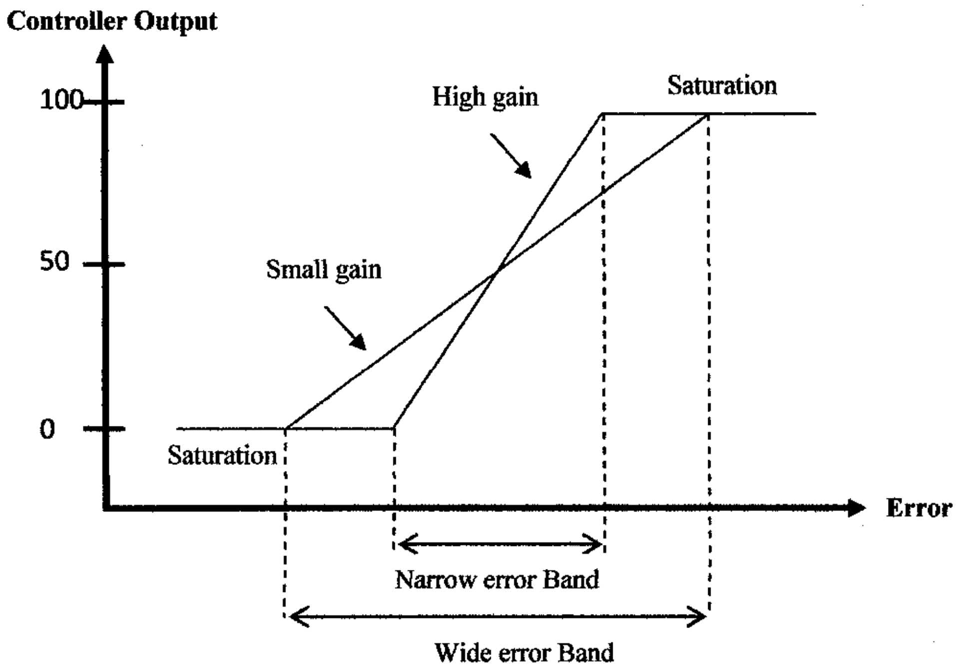
Figure 2: Block diagram for PID controller

Proportional (P) control mode simply is where the output the controller is simple proportional to the error e(t). The relation between the error, e(t) and the controller output, p is determined by a constant called proportional gain constant denoted as K_p. In P- mode, the natural extension of it would create a smooth and linear relationship between the controller output [p(t)] and the error [e(t)]. However when there is a zero error in the control system, the controller output should not be zero, otherwise would caused the process operation to halt. So, there will be some controller output (p_0) when there is zero error. The controller output equation would be:

$$p(t) = K_p . e(t) + p_o$$

where K_p= proportional gain constant
p_o= controller output with zero error

Proportional mode controller can be used whether for direct or reverse action where in the direct action; if the input to the controller increases it will cause the controller output to increase [5].



Graph 1 shows relationship between controller output as error increases and differences between higher and smaller gain where higher gain would create a wide error band.

Integral (I) control mode would persistently eliminate error in the control system. This would not be achieved by the proportional control mode as it will always have an offset error. Integral action is provided by summing the error over time, multiplying that sum by a gain and adding the result to the present controller output. For integral control mode, it is usually denoted as integral constant, K_i. The integral action will start to accumulate and make changes to controller output when the error becomes positive or negative for an extended period of time.

The integral control mode mathematical equation is:

$$p(t) = K_i \int_0^t e(t)dt + p(0)$$

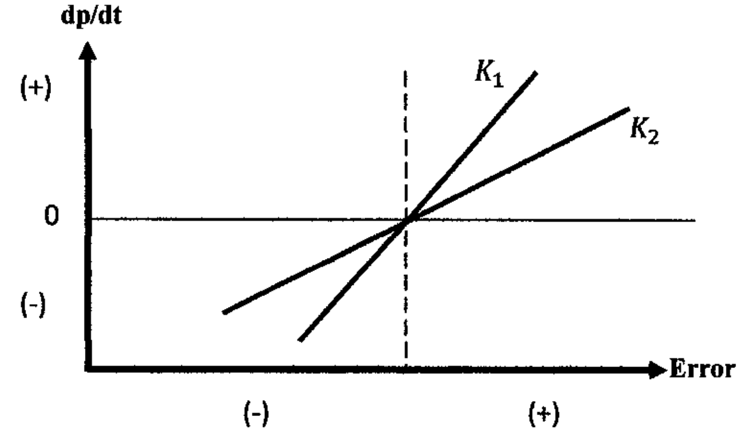
Where K_i = intergral constant

p(0) = controller output when integral action starts i.e. at t=0

In integral action, the value of the controller output p(t) is changed at a rate which is proportional to the actuating error signal e(t) which can be represented as:

$$\frac{dp(t)}{dt} = K_i e(t)$$

So, when there is zero error, the integral action would also be zero which means the controller output would not be changed.



Graph 2 shows the relationship of controller output as error increases with K_1 is higher gain than K_2

The integral gain, K_i is often represented by inverse which is called integral time or reset action, T_i= 1/ K_i. It is expressed in minutes instead of seconds.

Derivative (D) control mode is responsible for the rate at which error is changing that is the derivative of the error. The controller output depends on the time rate of change of actual errors whereby it is also called rate action mode or anticipatory action mode [6]. The mathematical equation for this mode is:

$$p(t) = K_d \frac{de(t)}{dt}$$

Where K_d = derivative gain constant

The $K_d$will indicates by how much % of the controller output must change for every % per sec rate of change of the error where it generally represented in minutes. If the derivative gain is increased, the rate of change of error would also increase thus produces sudden changes in controller output.

However, normally in process control, a single controller mode of P or I or D would not be used due to the disadvantages of proportional, P and derivative, D control modes that is it will not achieve zero offset. Offset or steady state error occurs when the output deviates from the set point. It will also depend on the reaction rate of the controller where higher reaction rate produces larger offset error. So, the composite control modes are used in order to gain advantages and eliminate some limitations of each mode. The various composite control modes are:

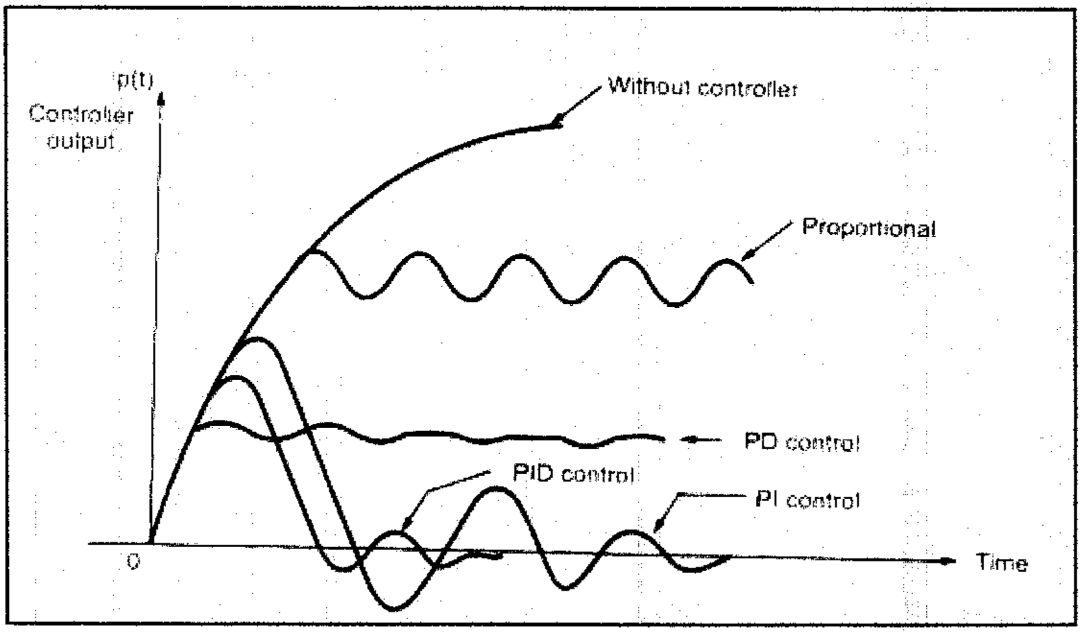
1- Proportional + Integral Mode (PI)
2- Proportional + Derivative Mode (PD)
3- Proportional + Integral + Derivative Mode (PID)

The mathematical expression for the three mode controller PID would be:

$$p(t) = K_p e(t) + K_p K_i \int_0^t e(t)dt + K_p K_d \frac{de(t)}{dt} + p_i(0),$$

where all the terms were defined in the previous equations.

Each of them will be used together because of the different advantages for each of them. For example, PI control mode completely removes the zero offset problems of proportional mode. However, the PI control mode will have relatively slow changes in the load because it did not have derivative action in the process control. For PD control mode, it will reduce the overshoot and the rise time so that the response of the control mode would be faster. However, in this control mode, it will not eliminate zero offset problems. For PID control mode, it is more complex than other composite control modes where it combines the proportional, integral and derivative action but it can virtually be used for any processes condition. PID controller modes would eliminates the zero offset problems, faster response and smaller dead time in the process control. So, it depends on the situations and conditions of the particular process control to use any of the composite controller modes.



Graph 3 shows the response of various control modes to a unit step load change.

2.2 Intelligent Controllers

Nowadays, the modern controllers for process control system are used with intelligent control system to replace conventional controllers (PID controller) due to its simple structure; it is difficult to overcome the effect of uncertainties [7]. Intelligent system or machine intelligent is defined by Yong – Zai Lu [8] as the systems or machine designed should have some human like behaviors such as parallels information processing, adaption to environment, associative memory, learning, generalization and self organization. For intelligent control, it can define as the application of intelligent system methodologies in system control such that the control system is furnished with some intelligent behaviors as stated above.

Some intelligent control systems that are used in current industrials process controls are fuzzy control, neural network and general algorithm controller.

2.2.1 Fuzzy Logic Controller

Fuzzy logic theory was first introduced by Zadeh (1965) and since that, it has become one of the attractive techniques in developing an intelligent control system [9]. Fuzzy logic which is based on the concept of fuzzy sets provides a means for representing uncertainties. These fuzzy sets are defined through the concept of membership function, μ where it can have value between 0 and 1. If the values are between 0 and 1, it is known as partial membership in the fuzzy set whereas for 0 or 1 denote a complete membership or non- membership [10]. Fuzzy logic theory also generally follow the heuristic method which is if –then rule.

IF <condition> THEN < consequence>

From the if-then rule, two main types of fuzzy models are introduced such as Mamdani model and the Takagi- Sugeno (TS) model.

Mamdani Model has a fuzzy variable defined by a membership functions in its consequents [11].

IF x is A_i THEN y is B_i , i = 1,2,...K,

where A_i is the antecedent and B_i is the consequence linguistic terms ('too small', 'small', 'medium', ' large', 'too large') represented by fuzzy sets and K is the number of rules in the model

Meanwhile, Takagi-Sugeno (TS) model are expressed by a linear combination of weighted input variables.

IF x is A_i THEN $y_i = a_i^T x + b_i$, i = 1,2,..., K,

where a_i is the consequent parameter vector, b_i is a scalar offset and i = 1,...,K. Antecedents describe fuzzy regions in the input space in which the consequent functions are valid. Output y is computed as the weighted average of the individual rule's contributions:

$$y = \frac{\sum_{i=1}^{K} \beta_i(x)\left(a_i^T x + b_i\right)}{\sum_{i=1}^{K} \beta_i(x)}$$

where β_i(x) is the degree of fulfillment of the i th rule .

The differences of between Mamdani and TS fuzzy models is that for Mamdani, the outputs are fuzzy sets while for the TS models, the output is singleton or expected to be defuzzified before fed to the PID controller. Basic components of fuzzy logic system are divided into three main parts which is input interface, linguistic description (fuzzy rule base and fuzzy sets and operators), and output interface [12].

From the fuzzy logic theory shown above, the Fuzzy Logic Controller (FLC) can be designed and implemented in the control system of process plant. The figure below had shown the basic structures of FLC in control system.

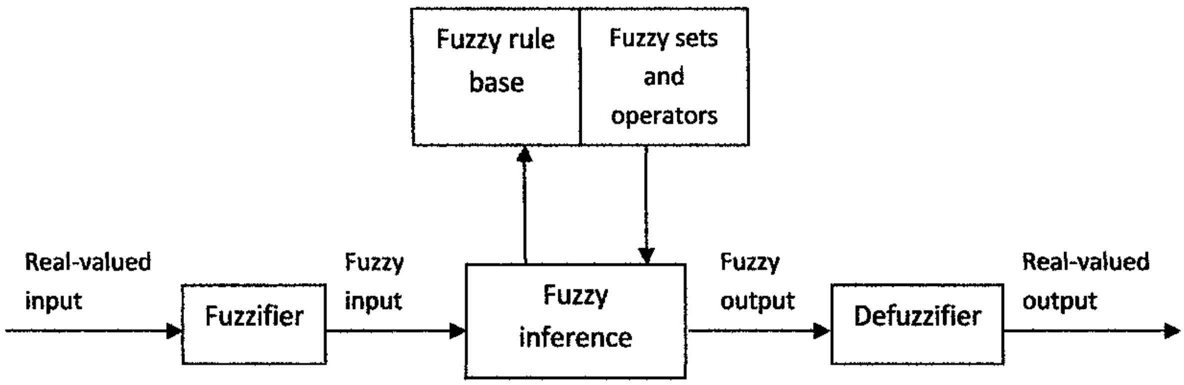


Figure 2: The basic components of a fuzzy logic control system.

Fuzzifier or fuzzification process is the process of mapping inputs to the Fuzzy Logic Controller into fuzzy set membership values [13]. For fuzzification process, the decision need to be made for the number of inputs, size of universes of discourse, and number and shape of the fuzzy sets. The larger number of the fuzzy sets would ensure better control performance as more accuracy for smaller error. Then, each fuzzy sets will be given a linguistic label for identity i.e. Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM) and Positive Large (PL).

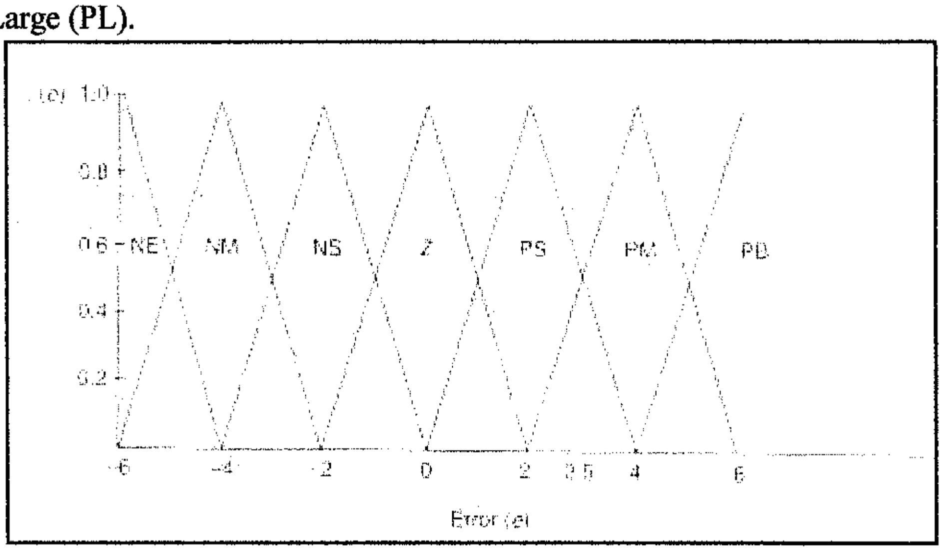


Figure 3: Seven set fuzzy input windows for error, e.

Fuzzy rule base that used the concept of IF-THEN shall consist of the antecedent and consequent linguistic rules in the form of:

IF error, e is A AND del PV is B THEN μ is C

This rule base will be constructed using a priori knowledge from either one of the sources [13]:

- Physical laws that govern that plant dynamics
- Data from existing controllers
- Imprecise heuristic knowledge obtained from experienced experts.

Then, from the fuzzy rule, the equation of fuzzy logic can be written in Boolean OR and AND function as:

$$\mu_c(u) = \max[\min(\mu_A(e), \mu_B(del\ PV))]$$

Table 1: Example of tabular structure of linguistic fuzzy rule base.

delPV \ e	NL	NM	NS	Z	PS	PM	PL
NL	NB	NB	NB	NM	Z	PM	PB
NM	NB	NB	NB	NM	PS	PM	PB
NS	NB	NB	NM	NS	PS	PM	PB
Z	NB	NM	NS	Z	PS	PM	PB
PS	NB	NM	NS	PS	PM	PB	PB
PM	NB	NM	NS	PM	PB	PB	PB
PB	NB	NM	Z	PM	PB	PB	PB

Defuzzification process is the process of mapping or converting the fuzzy output signal into a non- fuzzy control signal. For defuzzification process, the most well known method that is being used is Center of Area (CoA) or center of gravity method [12, 13]. The algorithm for that method is:

$$Defuzzification = \frac{\int u.\mu(u)du}{\int \mu(u)du}$$

Where u = output vector for each linguistic fuzzy rules

μ = membership function, MF

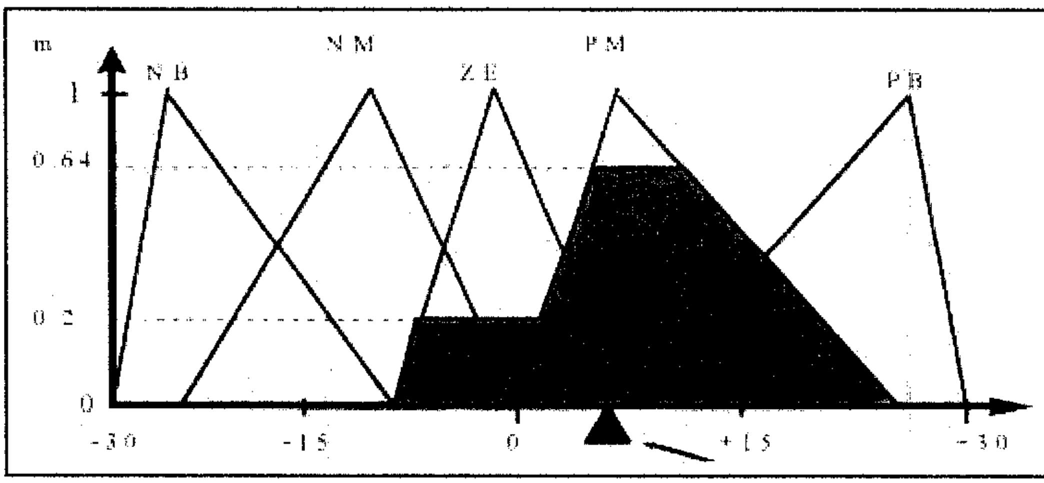


Figure 4: Center of Area (CoA) defuzzification.

2.3 Future Controller system

Conventional controllers such as PID controllers work well if the plant or process is reasonably linear. Each of the modes in the PID controller can be adjusted or tuned by using tuning methods such as Ziegler- Nichols (closed loop tuning), Cohen Coon (open loop tuning) and Ciancone Correlations method to give the best performance for the feedback control system. The three terms controller is widely used in nowadays industrial process control because of its simplicity and for linear system. However, if the final control elements used is nonlinear or the mathematical process modeling is too complex due to poor knowledge of process and imprecise known parameters (uncertainties), PID controllers may not be the best controller to be used. It can cause instability to the process control system. So, the best solution will be to use human control experience for manual adjustments. Intelligent control system is the attempt of adapting the human intelligence into the machine.

By using the intelligent control system such as Fuzzy Logic Control (FLC) will ensure the representation of uncertainty in input parameters that is associated with imprecision, vagueness and lack of information [13]. This will guarantees faster parallel operation because it does not need for calculation to complete it processing task. It also increases robustness in the control system as FLC processed each rule independently for minimal effects on its final output. Meanwhile, the Artificial Neural Network (ANN) controller would ensure that the process control to have self learning capability by experience. It also has the ability to generalize from given training data to unseen data [14].

For feedback control system, it is very important to achieve stability in the process control where linear system usually gives good results regarding the local stability of the system [15]. However, for conventional FLC, it has been reported that it has certain issues with the stability thus raising concern of its reliability [16]. The fix values of Scaling Factor (SF) and simple Membership Function (MF) might not always be sufficient to generate the necessary control actions [17].

Therefore, for the stability of fuzzy control system, some of the traditional stability analysis methods for nonlinear control systems have been applied by combining both of them [18]. One of them is by combining PI with Fuzzy Logic Control or PD with Fuzzy Logic Control. Each of them has its own advantages for example, PD – Fuzzy controller is suitable to be used for several class of system [19] and it produces quicker response with less oscillation. However, PI – Fuzzy controller with an integrator is more common and practical for linear first order system [15].

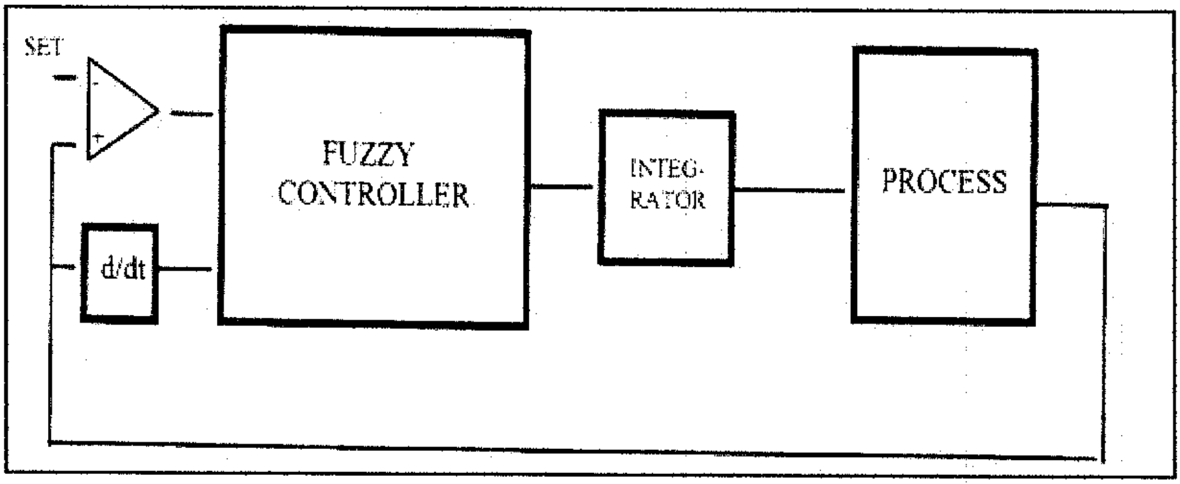


Figure 5:PI-Fuzzy Controller with integrator.

In [18], the author tried to design a robust self tuning PI- Fuzzy controller from the conventional PI- Fuzzy controller that will be more efficient for higher order of the system and marginally stable systems with different values of dead time. A robust self – tuning PI- Fuzzy controller shown in the paper has outperformed the conventional PI- Fuzzy controller in the case of the robustness and wide range of different processes. This may be the future trend for process controller as more improvement for the fuzzy controllers with hybridization with the conventional controllers (PID controller). From the hybridization of PID and Fuzzy Logic controller, many researchers has come up with proposal of adaptive fuzzy logic PID controller for more enhancement to the performance of controllers.

2.3.1 *Adaptive Fuzzy Logic PID controller (AFPIDC)*

AFPIDC is one of the hybrid controllers which combine the conventional PID controller and Fuzzy Logic Controller (FLC). It is designed to be adaptive in order to enhance the performance of the hybrid controllers in the control system. There are many ways of designing AFPIDC but one of them is shown in [20]. The FLC will produce three outputs as additional inputs for the three operating units of P, I and D of conventional PID controllers.

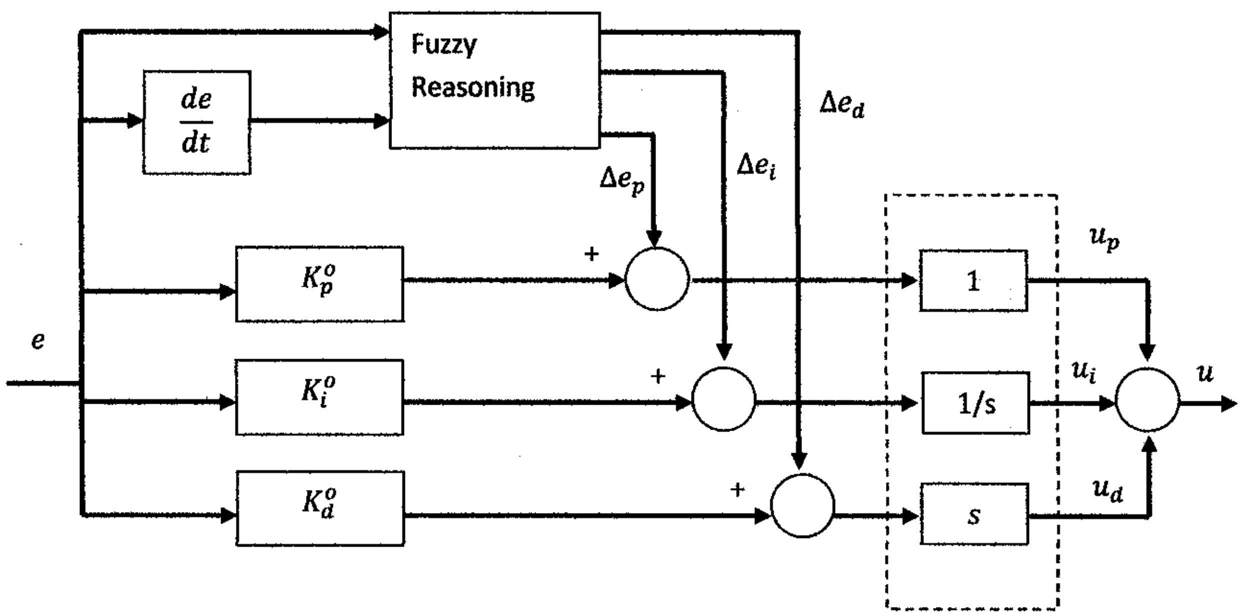


Figure 6: Block diagrams of AFPIDC proposed in [20].

From figure 6, FLC will be used to produce output signals of $\Delta e_p(t)$, $\Delta e_i(t)$ and $\Delta e_d(t)$. However, the FLC still using two inputs which is error, e and changes of error, $\Delta e(t)$. The outputs of the FLC considered as adaptive PID gain where the output will be

$$\Delta u(t) = \Delta e_p(t) + \int_0^t \Delta e_i(\tau)d\tau + \frac{d[\Delta e_d(t)]}{dt}$$

Then, it will be added with the initial PID gain which is

$$u^0(t) = K_p^0 e(t) + K_i^0 \int_0^t e(\tau)d\tau + K_d^0 de(t)/dt$$

The output of the addition of $\Delta u(t)$ and $u^0(t)$ will be feed in to operating inputs PID controller which will produced the output of the AFPIDC to the final element.

$$u(t) = \Delta u(t) + u^0(t)$$

$$= \Delta e_p(t) + \int_0^t \Delta e_i(\tau)d\tau + \frac{d[\Delta e_d(t)]}{dt} + K_p^0 e(t) + K_i^0 \int_0^t e(\tau)d\tau + K_d^0 de(t)/dt$$

$$u(t) = [K_p^0 + \Delta K_p]e(t) + \int_0^t [K_i^0 + \Delta K_i(\tau)]e(\tau)d\tau + \frac{d[K_d^0 + \Delta K_d(t)]e(t)}{dt}$$

where, $\begin{Bmatrix} \Delta e_p(t) = \Delta K_p \\ \Delta e_i(t) = \Delta K_i \\ \Delta e_d(t) = \Delta K_d \end{Bmatrix}$

From the fuzzy rules in [21], the tuning of each three outputs of FLC ($\Delta e_p(t)$, $\Delta e_i(t)$ and $\Delta e_d(t)$) will be as following:

i. If |e| is larger, then Δe_p should be larger, and then Δe_d should be smaller so that the system responds quickly. For intergral action, it should be limited ($\Delta e_i = 0$) to avoid system appearing large overshoot.
ii. If |e| is moderate, then Δe_p should be smaller, the value of Δe_d is more important to obtain a small overshoot.
iii. If |e| is smaller, then Δe_p and Δe_i should be larger to make the system have better steady state performance. When |Δe| is larger, Δe_d should be smaller and vice versa.

CHAPTER 3
METHODOLOGY

3.1 Project Flowchart

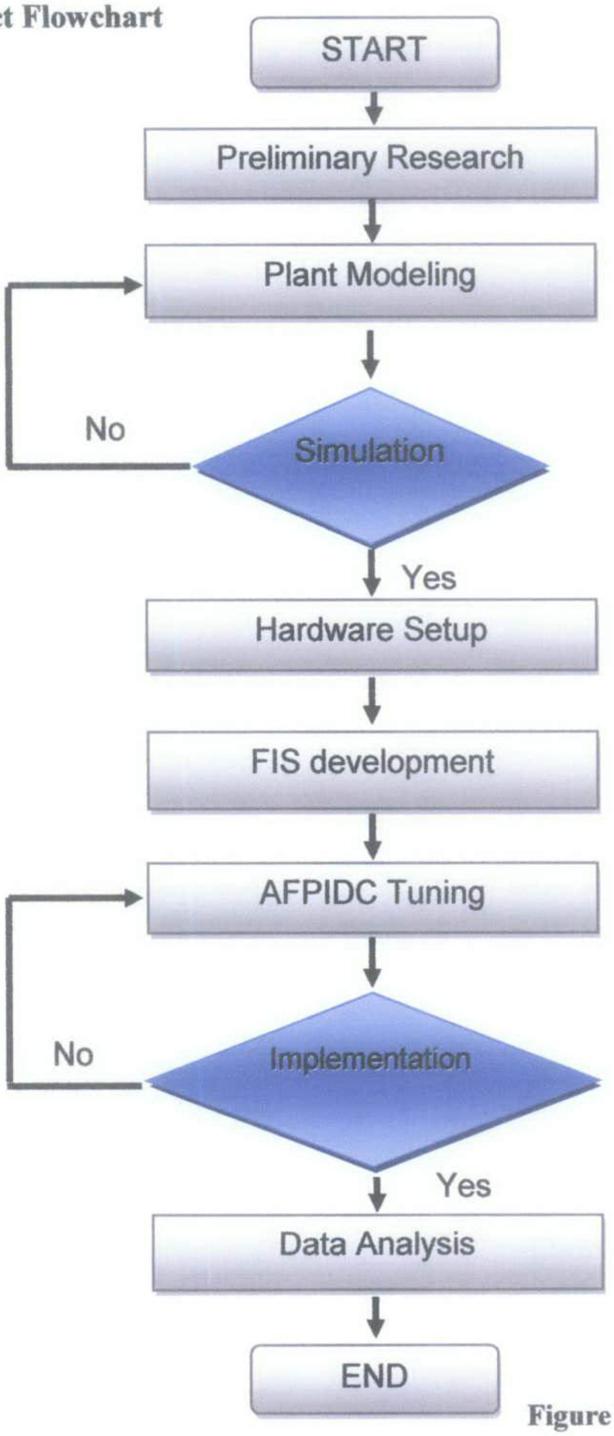


Figure 7: Flow chart of the project

3.2 Project Activities

Activity	Description
Preliminary Research	Process of gathering information regarding of the project from and tools used in this project such as hardware, software and model verifications. Result from this activity is the literature survey from certified scholars, thesis and journal papers and text books for PID, FLC and AFPIDC. Other necessary information such as MATLAB/Simulink manuals is also obtained.
Plant Modelling	Performed via Empirical Modelling method. Using the pilot plant mentioned earlier, the result of this activity is the generation of transfer function of FODT. Then, the mathematical expression is also used to compute the plant model and parameters for PID controller.
Simulation	Simulation of the plant involving FIS that is developed using ANFIS method and empirical model of the process plant. From this activity, the characteristic and behaviour of the AFPIDC and how it interact with the plant.
Hardware/ Experimental setup	The connection of the pilot plant with remote PC through Data Acquisition (DAQ) card will be established. Termination of cables, jumpers and resisitors are done to ensure the interfacing.
FIS Development	The Fuzzy Inference System (FIS) is developed and designed from the literature by designing the membership functions, input fuzzification and output defuzzification techniques.
AFPIDC Tuning	For the best performance of the controller, it has to be tuned. The parameters that can be used for tuning is the membership function or input/ output scaling.
Implementation	Implement the designed controller to the pilot plant. Then, the controller is subjected to random set point changes and disturbances. The DCS HMI will be developed using Matlab/Simulink.
Data Analysis	Analysis the result obtained from the designed controller to the process plant and compares it with the conventional FLC.

3.3 Gant Chart

For details of the Gantt chart for this project, please refer to *Appendix A- Gantt Chart.*

3.4 Milestone

Based on the Gantt Chat and Project Activities, the milestone for this project is:

1) Preliminary Research (from Week 1 – 6)
    - Result - Literature for the PID, Fuzzy Logic and Adaptive Fuzzy PID Controller, information for using controller function in MATLAB/Simulink.
2) Plant Modeling (from Week 6– 7)
    - Result – Empirical modeling of the pilot plant and generation of pilot plant transfer function which will be FODT.
3) Simulation (from Week 7 – 9)
    - Result – From the transfer function of process plant, simulate the control system for controller (PID, FLC and AFPIDC), final element, process plant and sensor using MATLAB/Simulink. Then, the output of the simulation is analyzed to see the performance of each controller using the transfer function obtained earlier.
4) Hardware / Experimental Setup (from Week 8-9)
    - Result – Setting up the connection between the remote PC and pilot plant through DAQ card. Establish the connection in the MATLAB/Simulink so that real time data trending can be shown on the remote PC (HMI).
5) FIS Development (Week 9)
    - Result – Design the FIS for the FLC based on the fuzzy linguistic rules and membership function by MATLAB/Simulink
6) AFPIDC Tuning and Implementation (from Week 10 – 13)
    - Result – Design the AFPIDC for the control system with all the block diagram (final element, process plant transfer function, sensor) using MATLAB/Simulink and then implemented it on the pilot plant.
7) Data Analysis (from Week 14 to Week 18)
    - Result – Based on result of the implementation of AFPIDC, the comparison of each controller will be made.

3.5 Tools and Equipments Required

Some of the hardware and software needed for the completion of the project for *PcA SimExpert Mobile Pilot Plant SE231B-21 Flow Control and Calibration Process Unit* will be as follows:

- A Laptop PC
- Data Acquisition (DAQ) card (USB-1208FS)
- MATLAB with Real Time workshop, Control system, Fuzzy Logic Toolboxes, Simulink and ANFIS
- Electronics component (250 ohm resistors) and jumper cables.

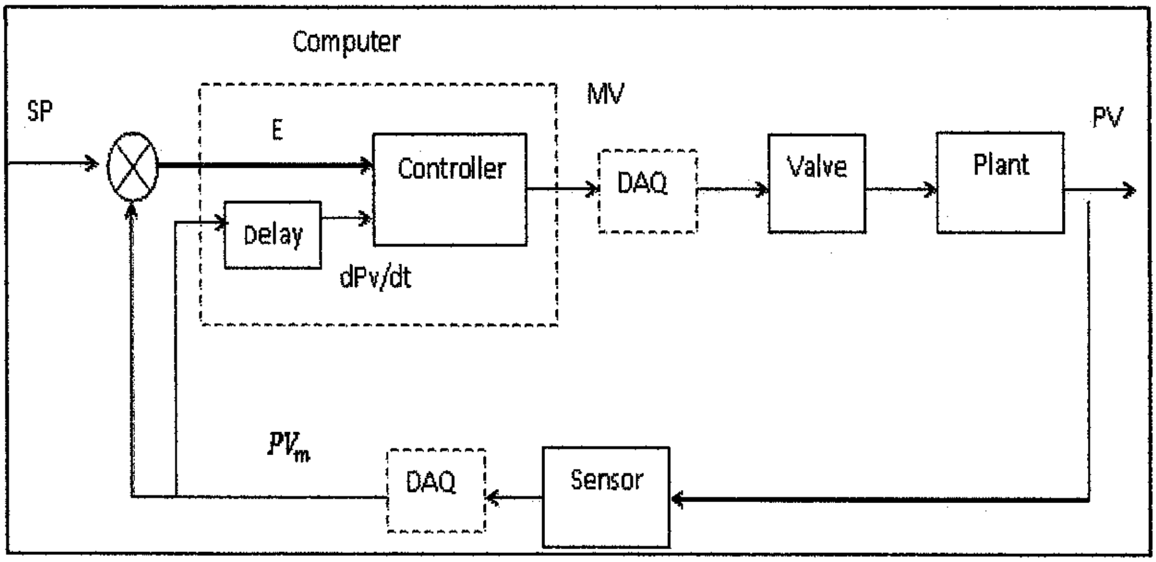


Figure 8: Connection of the remote PC and pilot plant through DAQ card.

CHAPTER 4
PRELIMINARY RESULTS AND DISCUSSION

4.1. Plant Process and Instrumentation Diagram (P&ID)

The Process and Instrumentation Diagram (P&ID) for the *PcA SimExpert Mobile Pilot Plant SE231B-21 Flow Control and Calibration Process Unit* are as shown in Figure 9 below.

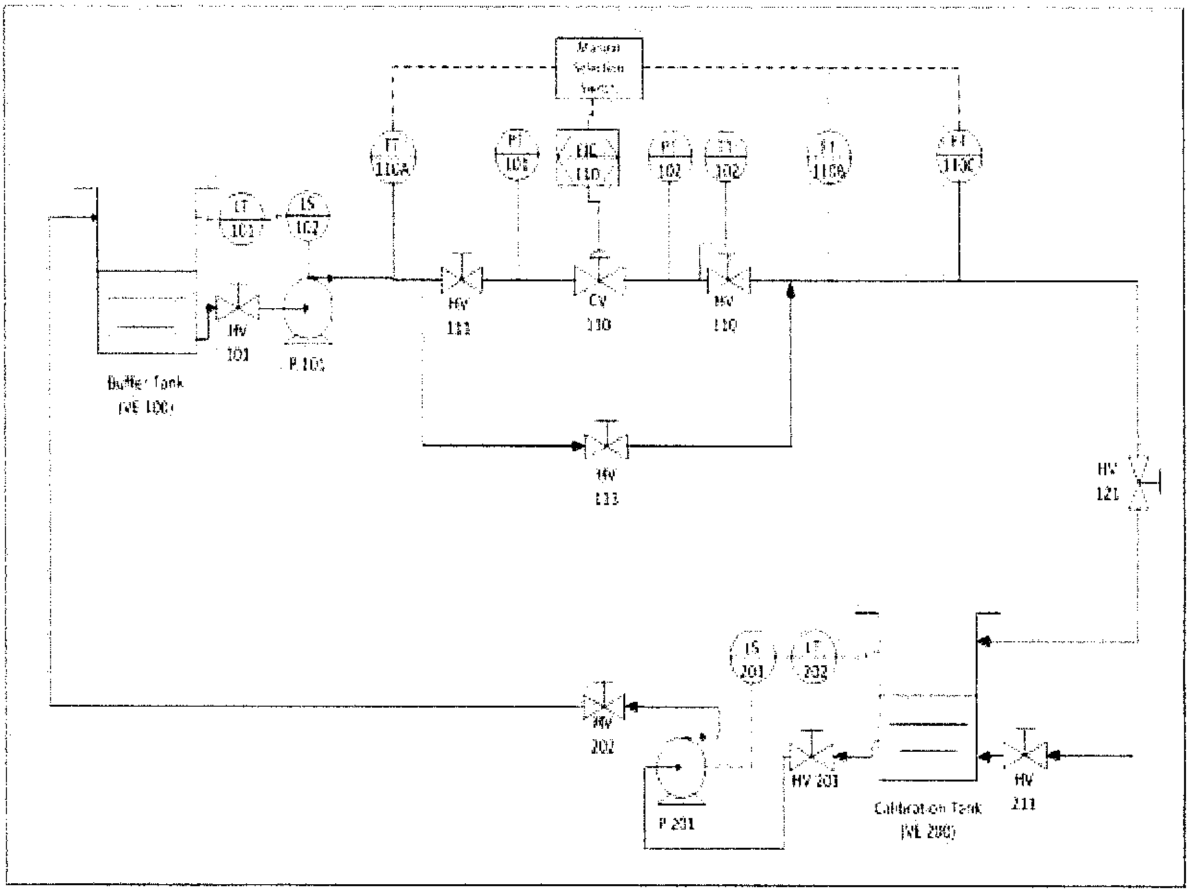


Figure 9: P&ID for the Flow Control Pilot Plant

4.2. Plant Process Description

From the P&ID shown in Figure 9 above, the basic operation of the pilot plant is to transfer the fluid (water) from the Buffer Tank (VE 100) to the Calibration Tank (VE 200), while controlling the flow of fluid in between of these two tanks. The flow must be controlled in such a way that the level of both tanks does not overflow and maintained at a steady predetermined level. The feedback loop of this pilot plant is controlled by FIC – 110 which currently using PID controller devices. The FIC – 110 will receive signal from one of the three flow transmitters in this pilot plant (FT – 110A, FT -110 B, FT – 110C) and fed back to the controller. These three flow transmitters, Coriolis Flow meter (FT- 110A), Vortex Flow meter (FT – 110B) and Orifice Flow meter (FT – 110C), can be used interchangeably by selecting either one of them using the Manual Selection Switch located in front of the pilot plant panel. Then, the output from the FIC – 110 will be sent to the control valve, CV – 110 where it has a causal relationship with the flow of fluid from the VE 100 to the VE 200 tank.

Besides this feedback loop, there are also other feedback loops that are available in this pilot plant which is the level switch (LS – 101) at the VE – 100 tank connected to the pump (P – 101) and level switch (LS – 201) at the VE – 200 tank connected to the pump (P – 201). If both level switches detected that the level of the fluid in the tank is low or high, it will automatically send the signal to the both pump to immediately shut down. So, basically, these two feedback loops is to ensure the fluid level do not overflow in one of the tanks and to protect the pumps from damage. The pressure transmitters (PT – 101 and PT – 102) are used to determine the before and after pressures of the line that goes through CV – 110. There is also temperature transmitter (TT – 102) to measure the temperature of the fluid in the tank and can be used to determine the density of the fluid. There are several hand valves (HV – 101, HV – 111, HV – 113, HV – 110, HV – 121, HV – 211, HV – 201, HV – 202) installed for blocking and allowing flow of the fluid manually in these lines.

4.3. Plant Modeling Techniques

There are basically three plant modeling techniques can be used which are Mathematical modeling, Empirical modeling and Statistical modeling. However, for this pilot plant, it is better to use Mathematical and Empirical Modeling.

4.3.1. Mathematical Modeling

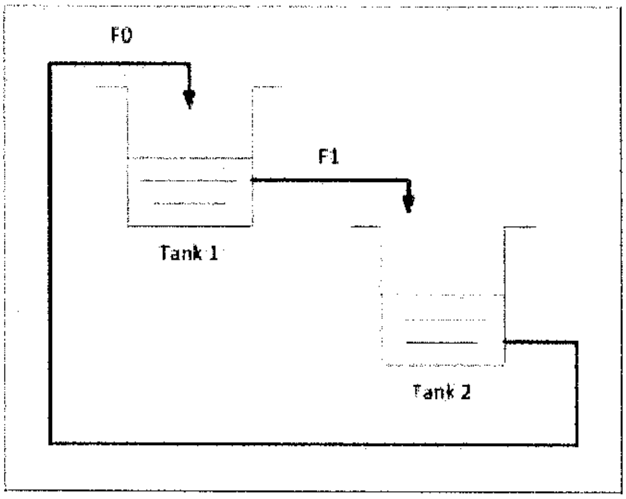


Figure 10: Simplified Two Tank System

The following steps must be followed in order to do the Mathematical Modelling for any process based on [20]:

1. Define goals
   a) Specific Design Decisions
   b) Numerical Values
   c) Functional Relationships
   d) Required Accuracy

2. Prepare information
   a) Sketch process and identify system
   b) Identify variable of interest
   c) State assumptions and data

3. Formulate model
   a) Conservation balances
   b) Constitutive equations

c) Rationalize (combine equations and collect terms)
d) Check degrees of freedom
e) Dimensionless form

4. Determine solution
   a) Analytical
   b) Numerical

5. Analyze results
   a) Check results for correctness
      i. Limiting and approximate answers
      ii. Accuracy of numerical method
   b) Interpret results
      i. Plot solution
      ii. Characteristics behaviour like oscillations or extreme
      iii. Relate results to data and assumption
      iv. Evaluate sensitivity
      v. Answer "what-if" question

6. Validate model
   a) Select key values for validation
   b) Compare with experimental results
   c) Compare with results from more complex model

For this particular pilot plant modeling, the density of the fluid ρ is considered to be constant and since the tank is cylindrical, the cross section of the tank doesn't change with the height of the tank. Available data for the plant are as follows:

- Cross section of the tank, $A = 551.55cm^2$
- Diameter of the tank is 26.5 cm
- Height of the liquid at 100% level is 83cm
- Initial steady state condition is the tanks level , L

The level depends on the total amount of liquid in the tank so the total conservation equation selected is an overall material balance of the system:

$$\frac{d(mass)}{dt} = \rho F_i(t) - \rho F_o(t)$$

From the equation above, the level of the liquid in the tank is equals to the difference in the inflow to thank and outflow from the tank. So, the equation will be:

$$A\frac{dL(t)}{dt} = F_i(t) - F_o(t)$$

Where $F_0 = C_d a \sqrt{2gL} = k_{FO}L^{0.5}$

$k_{FO} = C_d a\sqrt{2g} = 39.23$

F_0 is the flow rate of the liquid out of the tank in cm^3/sec

C_d is the discharge coefficient of the tank outlet = 0.7

a is the area of the tank = 1.266 cm^2

g is the gravitational constant = 9.8 m/s^2 = 980 cm/s^2

Combining all equations, the system can be described by a single first order differential equation:

$$F_i - k_{FO}L^{0.5} = A\frac{dL}{dt}$$

Subtracting the linearized balance at steady state conditions and noting that input is a constant step, then:

$$\Delta F_i(t) - \left(0.5k_{FO}L_s^{-0.5}\right)L' = A\frac{dL'}{dt}$$

The linearized Differential Equation can be rearranged and solve as:

$$\frac{\tau dL'}{dt} + L' = \frac{\tau}{A}\Delta F_i \ where\ \tau = \frac{A}{19.26{L_s}^{-0.5}}$$

and $K_p = \frac{\tau}{A} = \frac{1}{19.26{L_s}^{-0.5}}$

From the above equation, it is transformed using Laplace transform to obtain a transfer function whereby the input is the inflow and the output is the liquid level.

$$L'(s) = \frac{K_p}{\tau s + 1} F_i'(s)$$

However, it will still depend on the initial level of the liquid as it will affect the flow rate of the outlet.

4.3.2. Empirical Modeling

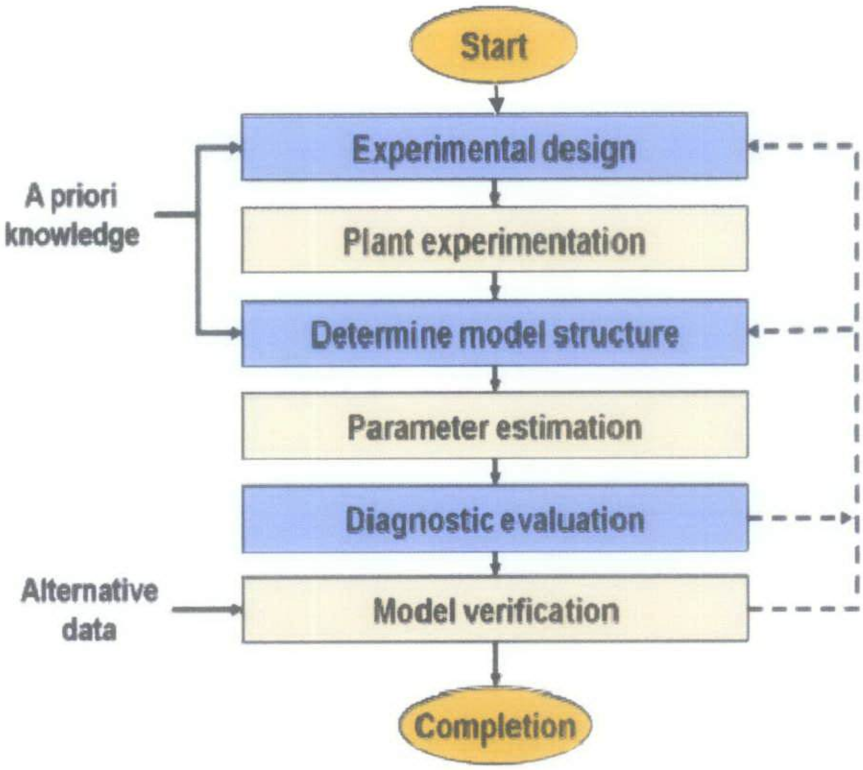


Figure 11: Empirical Modeling Technique Flow Chart

From figure 11, it shows the steps must be taken in order to model the plant using Empirical Modeling Technique for the process plant [21]. There are basically two methods to obtain Empirical modeling which are:

a) **Method I** – uses slope from the plant's data to obtain the time constant, steady state process gain and fraction dead time.
b) **Method II** – uses the time at 28% and 63% of the slope by using simple mathematical calculation to obtain the time constant, steady state process gain and fraction dead time.

In order to implement the Empirical Modeling in modeling the pilot plant, the Process Reaction Curve (PRC) must be obtain through experiment on the process plant. Then, Cohen Coon open loop tuning method is used based on the PRC obtained in order to calculate the initial PID parameters for the process plant. So, using Method II, the opening of the control valve (MV) is changed from 50% to 75%.

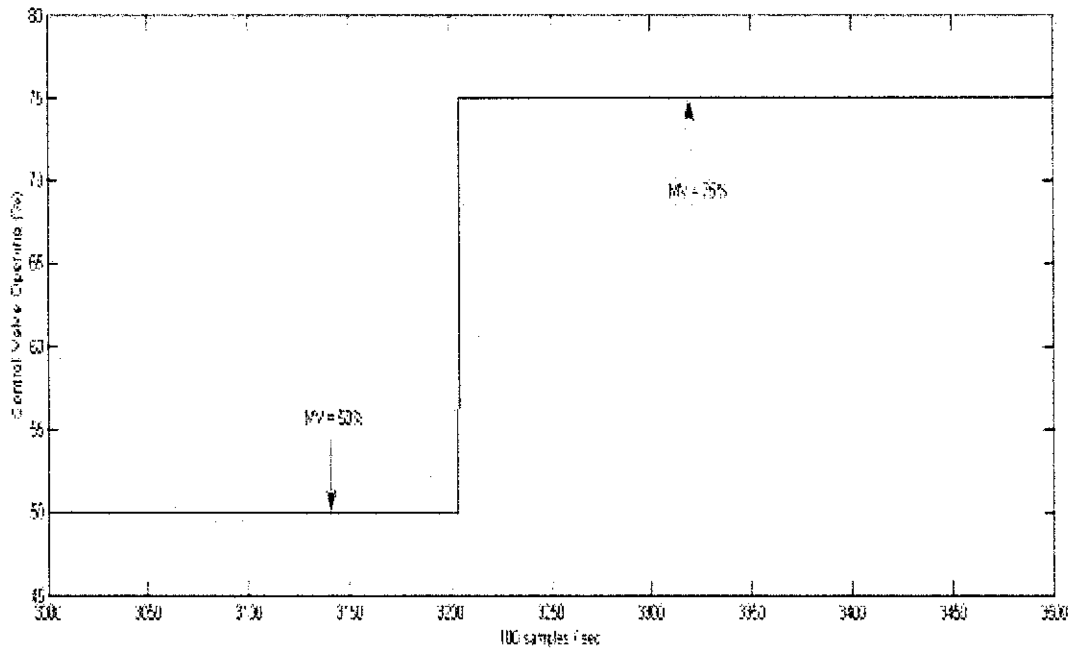


Figure 12: MV changes (Percentage of control valve opening)

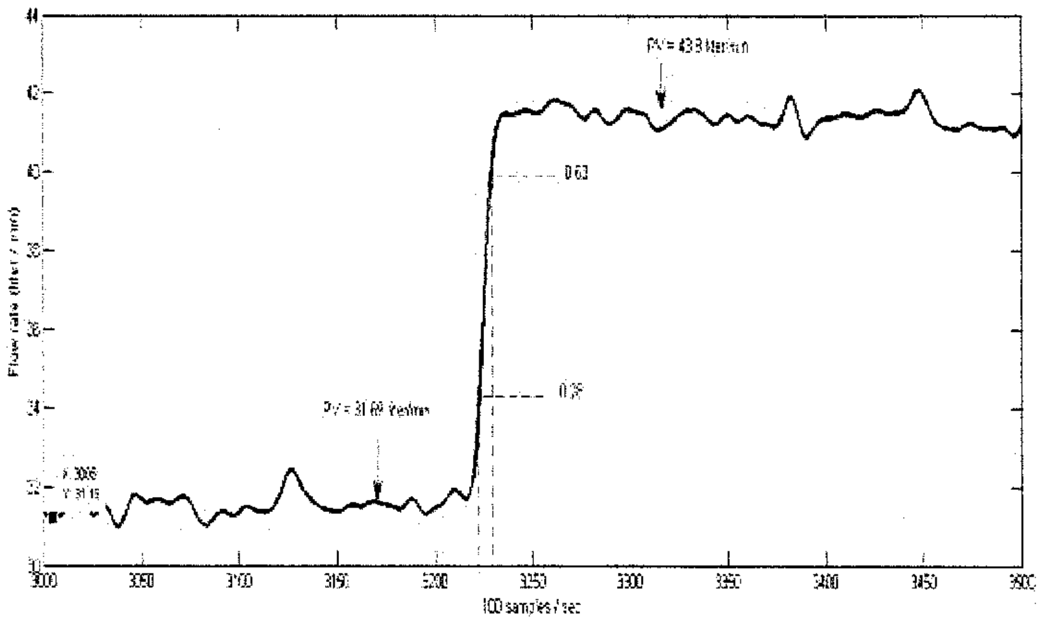


Figure 13: Process Reaction Curve (PRC) (changes in PV - Orifice Flow Meter)

The PRC above is acceptable, based on following rules to determine a good PRC from the experiment:

- The input (MV) magnitude is large enough to give an output (PV) signal to noise ratio greater than 5 which in this case the MV changes are 25.
- The duration for the experiment is more than $\theta + 4\tau$.
- The input (MV) change is a perfect step change.
- The model obtained matches with the First Order with Dead Time process model.
- No significance disturbances that give large effect to the accuracy of PRC.
- When decreasing back the input (MV) to initial condition, output also change to initial condition as can be seen in Figure 14.

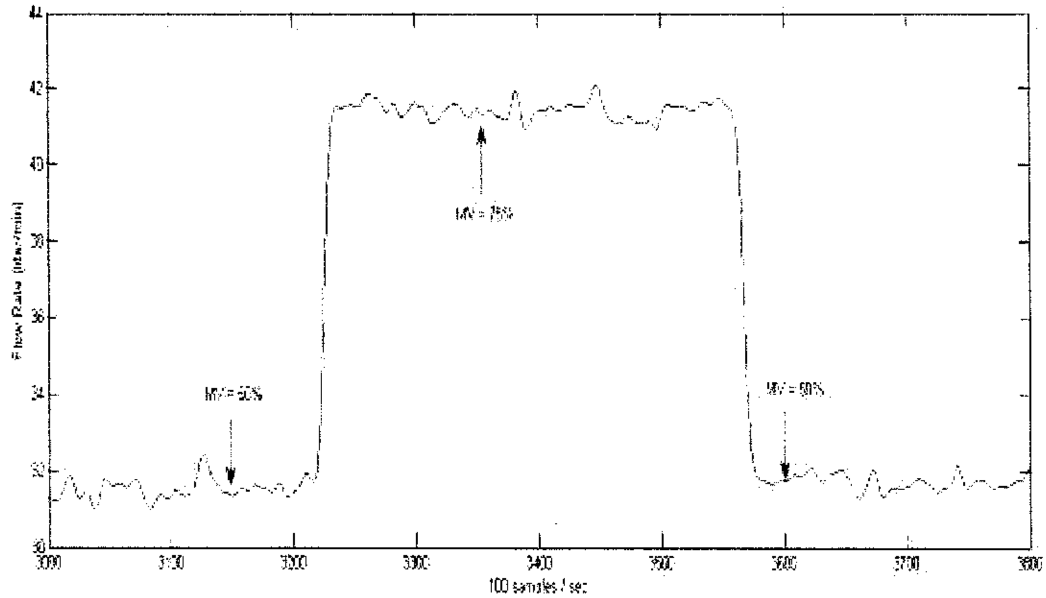


Figure 14 : PV back to initial value after MV decrease to initial

Then, from the PRC obtained in Figure 13, the parameters that can be obtained are:

$$\sigma = \delta = 75 - 50 = 25$$

$$\Delta = 43.8\ l/\min - 31.69\ l/min$$

$$= 12.11\ l/min$$

$$K_p = \frac{\Delta}{\delta} = \frac{12.11}{25} = 0.4844$$

$$0.63\Delta = 0.63(12.11) + 31.69 = 39.32\ l/min$$

$$t_{0.63\Delta} = (3229 - 3204)/100 = 0.25s$$

$$0.28\Delta = 0.28\,(12.11) + \ 31.69 = 35.08\ l/min$$

$$t_{0.28\Delta} = (3223 - 3204)/100 = 0.19s$$

$$\tau = 1.5\ (t_{0.63\Delta} - t_{0.28\Delta}) = 1.5(0.25 - 0.19) = 0.09s$$

$$\theta = t_{0.63\Delta} - \tau = 0.25 - 0.09 = 0.16s$$

Table 2: Results for Process Reaction Curve

Measurement	Value
Change in perturbation / MV , σ	25%
Change in output / PV, Δ	12.11 *l/min*
Maximum Slope, S	-
Apparent Dead Time, θ	0.16 sec
Calculations	**Value**
Steady state process gain, $K_p = \Delta/\delta$	0.4844
Apparent Dead time, $\tau = \Delta/s$ Or $\tau = 1.5\ (t_{0.63\Delta} - t_{0.28\Delta})$	0.09
Fraction dead time, $R = \theta/\tau$	1.7778

Table 3: PID Tuning Constant

Tuning Parameters	P - only	PI	PID	PD
Proportional Gain, K_p	0.6881	0.1720	2.0644	1.7956
Integral Gain, T_i	-	0.569	0.7508	-
Derivative Gain, T_D	-	-	0.0440	0.0143

The PID tuning constants from Table 3 were obtained using Cohen Coon open loop correlations. From the process reaction curve (PRC), the First Order with Dead Time (FODT) model was obtained:

$$G_p(s) = \frac{0.4844e^{-0.16s}}{0.09s + 1}$$

4.4. Preliminary Simulation Results

Using the FODT transfer function obtain for the process plant and PID parameters from the Table 2 and 3 above, several simulations has been conducted using MATLAB and Simulink software.

Simulation control model of the flow process plant for PID controller

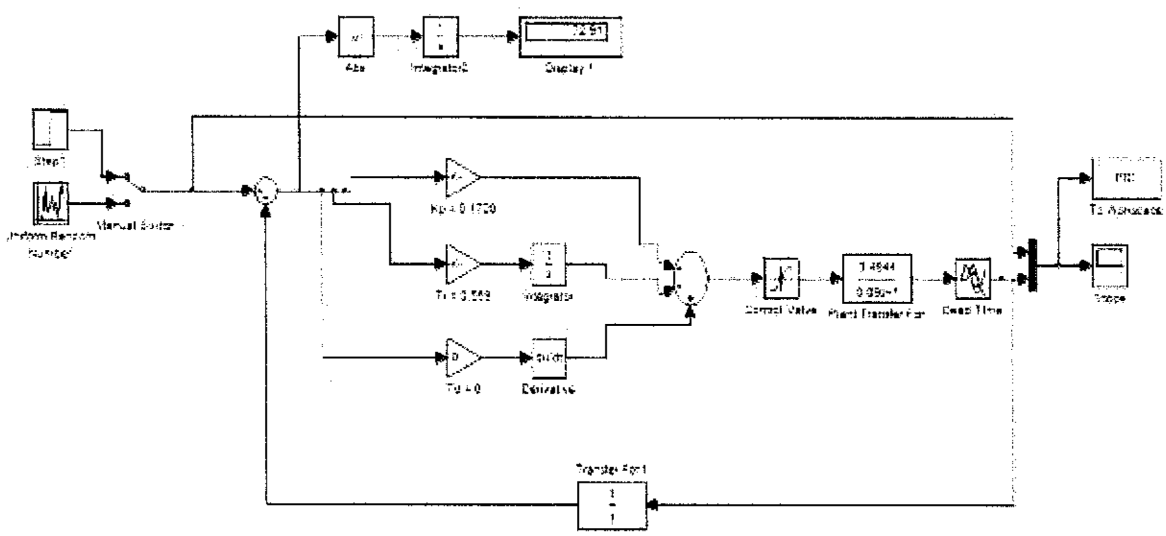


Figure 15 : Plant Model using PID controller

Results of Simulation for PID Controller

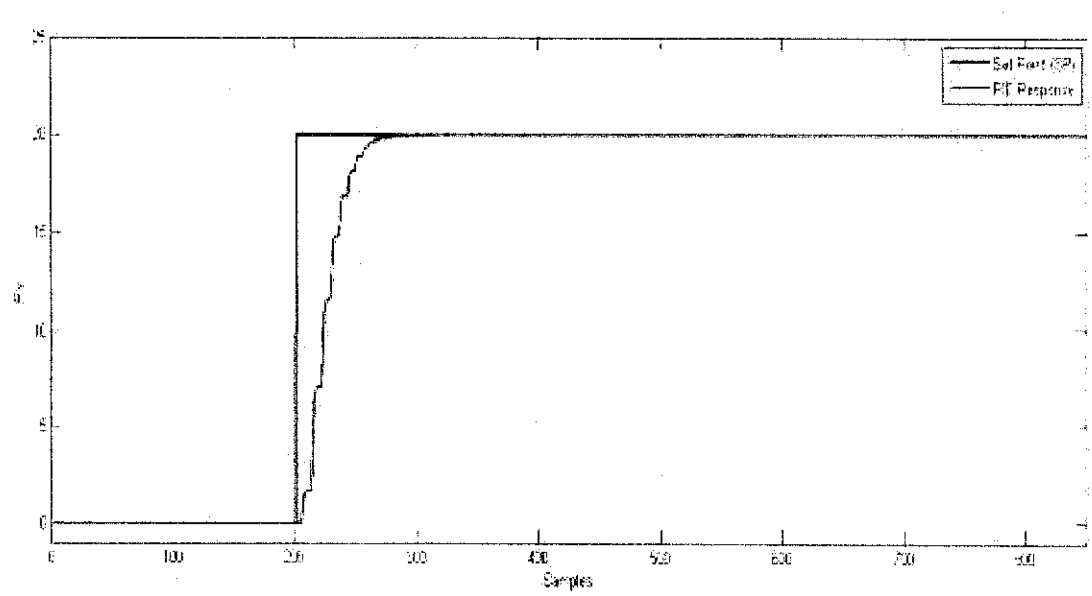


Figure 16: Controller response PID for Step Input change

4.5 Fuzzy Logic Controller Development

Fuzzy Logic Controller used in this project has been designed and developed by Elangeshwaran, P. in [23]. However, it has been implemented onto the same pilot plant using Coriolis Flow Meter however for this project, Orifice Flow Meter is used. The relationship between inputs and outputs is shown in Table 4 below. It has a total of 81 rules which has better control performances and accuracy at smaller errors.

Table 4: Relationship between Input and Output

E \ PV	NB	NI	NM	NS	Z	PS	PM	PI	PB
NB	NB	NB	NB	NB	NB	NB	NB	NB	NB
NI	NM	NM	NS	NI	NI	NI	NS	NM	NM
NM	Z	NM	NM	NM	NM	NM	NM	NM	Z
NS	PS	Z	NS	NI	NS	NI	NS	Z	PS
Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
PS	NS	Z	PS	PI	PS	PI	PS	Z	PS
PM	Z	PM	PM	PM	PM	PM	PM	PM	Z
PI	PM	PM	PS	PI	PI	PI	PS	PM	PM
PB	PB	PB	PB	PB	PB	PB	PB	PB	PB

Based on Table 4, the rules are set based on four rules of thumbs which are:

1) If E is large and delPV is small then valve opening/closing must be quick
2) If E is large and delPV is large then valve opening/closing must be be moderate
3) If E is small and delPV is small then valve opening/closing must be small
4) If E is small and delPV is large then valve opening/closing must be small in the opposite direction.

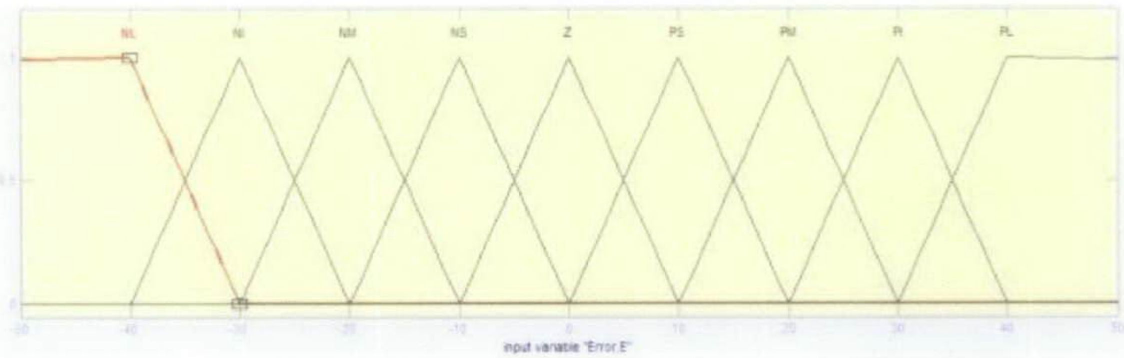


Figure 17 : Membership Function for Error,E

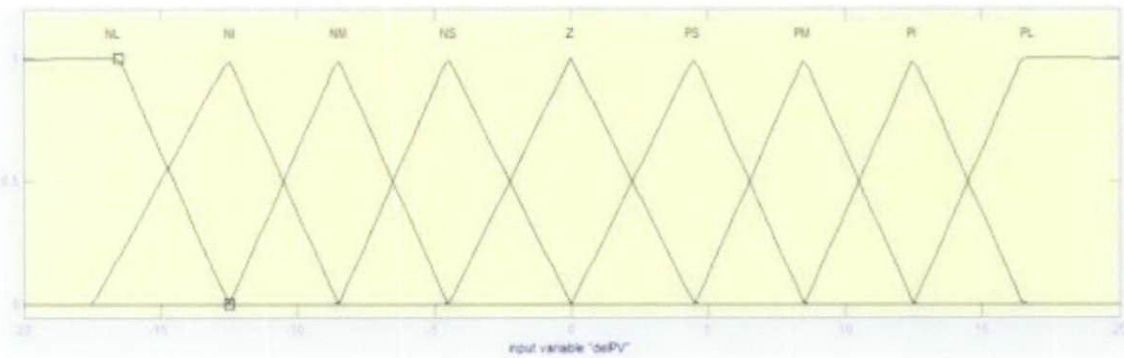


Figure 18 : Membership Function for del PV

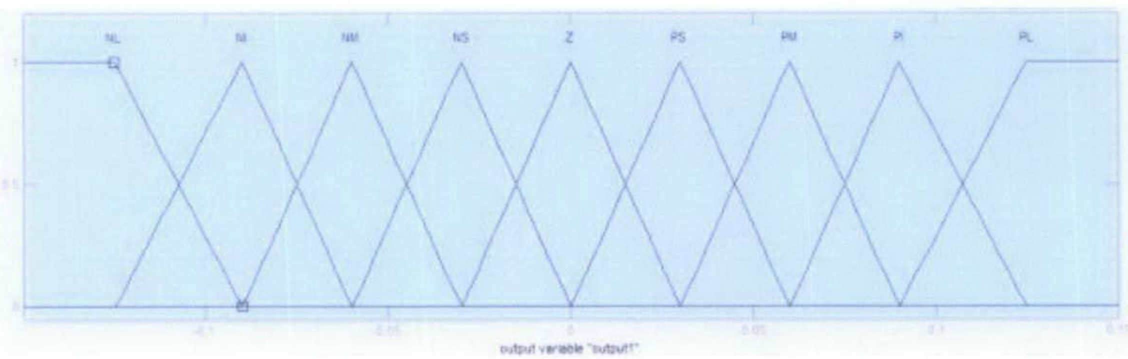


Figure 19 : Membership Function for Output

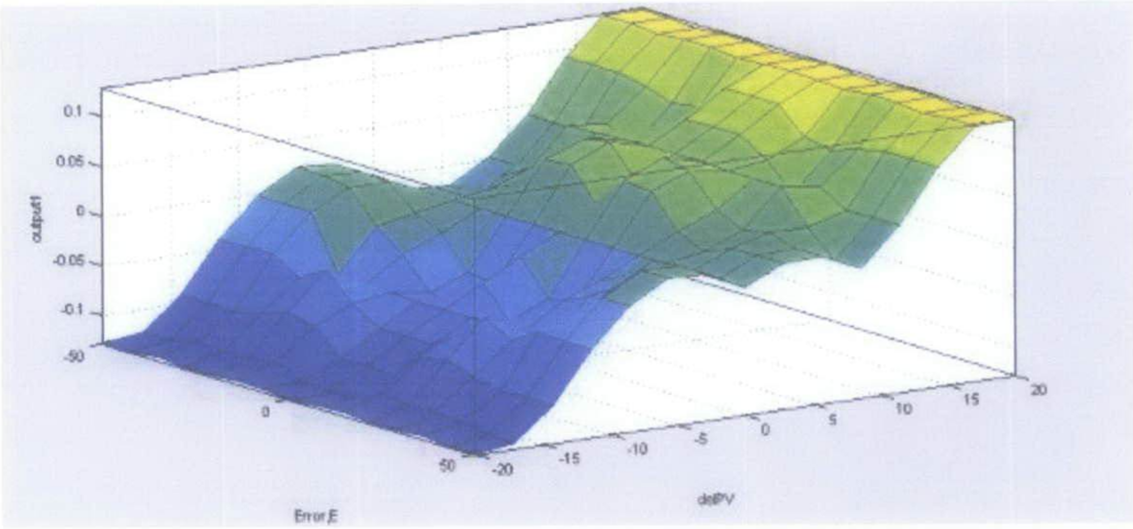


Figure 20 : Surface View of FIS

4.6 Adaptive Fuzzy – PID Controller Development and Implementation

4.6.1 Designing Fuzzy Inference System for AFPIDC

In order to design Fuzzy Logic Reasoning for the AFPIDC, the Fuzzy Logic Toolbox in MATLAB/Simulink can be used for this project. The controller will then be used to control the flow of liquid in the plant, so there are basically two inputs to the FIS i.e. error, *E* and rate of change of error, *delE*. Then, the Fuzzy Logic Reasoning, FLC will produce three signals or outputs as shown in Figure 6. The three outputs [$\Delta e_p(t)$, $\Delta e_i(t)$,$\Delta e_d(t)$] will be added with the conventional PID gains which are [K_p^0, K_i^0, K_d^0]. The output of the controller is the control signals to the control valve which is the manipulated variable, *mv*. From the inputs and outputs for the AFPIDC, the relationship can be seen in Table 5, 6 and 7. The input – output relationships are used to develop the IF-THEN rules for the fuzzy inference system (FIS). There are 49 rules created using Fuzzy Logic reasoning for each of the FIS. However, the rules can be made larger as it will ensure better control performance with better accuracy even when there are smaller errors.

According to table 5, 6 and 7, the fuzzy control rules are built according to [20] in tuning the [$\Delta e_p(t)$,$\Delta e_i(t)$,$\Delta e_d(t)$] uses the following three rules of thumb:

1) If |E| is larger, then Δe_p should be larger meanwhile Δe_i should be limited to 0 or smaller to avoid the system appearing large overshoot.
2) If |E| is moderate, then Δe_p and Δe_i should be moderate.
3) If |E| is smaller, then Δe_p should be smaller and Δe_ishould be larger for better steady state performance.

Mean while for *del E,* usually Δe_d will be changed for faster or slower response of controller but flow response is a fastest response than other applications, so Δe_d is left to be zero to avoid unstable response of controller.

Table 5 : Fuzzy Control Rule for Δe_p

delE \ E	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	Z	PS	PS	Z
NM	PB	PB	PM	Z	PS	Z	Z
NS	PM	PM	NM	Z	Z	NS	NM
Z	PM	PS	PS	Z	NS	NM	NM
PS	PS	PS	Z	Z	NS	NM	NM
PM	Z	Z	NS	Z	NM	NM	NB
PB	Z	NS	NS	Z	NM	NB	NB

Table 6 : Fuzzy Control Rule for Δe_i

delE \ E	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	NM	Z	Z
NM	NB	NB	NM	NM	NS	Z	Z
NS	NM	NM	NS	NS	Z	PS	PS
Z	NM	NS	NS	Z	PS	PS	PM
PS	NS	NS	Z	PS	PS	PM	PM
PM	Z	Z	PS	PM	PM	PB	PB
PB	Z	Z	PS	PM	PB	PB	PB

Table 7 : Fuzzy Control Rule for Δe_d

delE \ E	NB	NM	NS	Z	PS	PM	PB
NB	Z	Z	Z	Z	Z	Z	Z
NM	Z	Z	Z	Z	Z	Z	Z
NS	Z	Z	Z	Z	Z	Z	Z
Z	Z	Z	Z	Z	Z	Z	Z
PS	Z	Z	Z	Z	Z	Z	Z
PM	Z	Z	Z	Z	Z	Z	Z
PB	Z	Z	Z	Z	Z	Z	Z

The inputs, *E* and *del E*, and output, *MV* has seven membership functions i.e. negative big (NB), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM) and positive big (PB). Figure 22 shows the membership function of the two inputs and output designed in MATLAB/Simulink for the Fuzzy Inference engine. From the Figure 23 and 25, the range for the first input, *E,* is [-50 50] following the range of operation for Orifice Flow meter in the plant which is set from 15 *l/min* to 45 *l/min*. The valve operation is non- linear in nature as can be seen from Figure 21 will give best linearity from 29% to 70% of opening. The error would not exceed ± 20 *l/min,* because the largest set point change can only be ± 20 *l/min.* For the second input, *del E,* the range is chosen at [-20 20] because during in the empirical modeling stage, the change in error is found to be averaging around ± 20 *l/min* when there is a step change in the set point. Meanwhile for the output, the range is [-0.2 0.2] which can be seen in Figure 24.

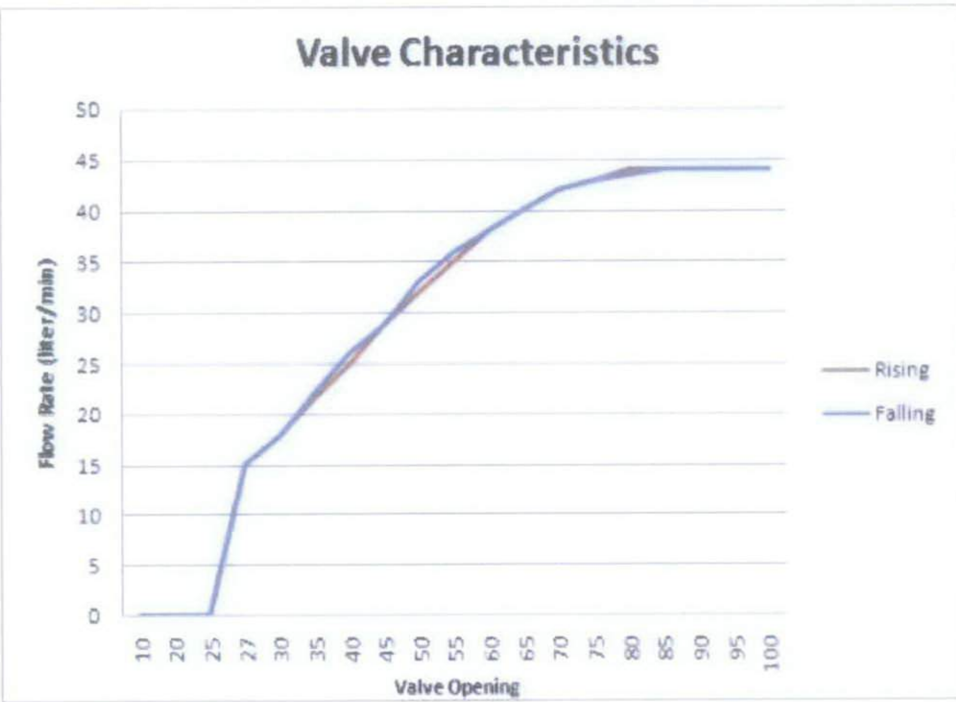


Figure 21: Valve characteristics for Orifice Flow meter

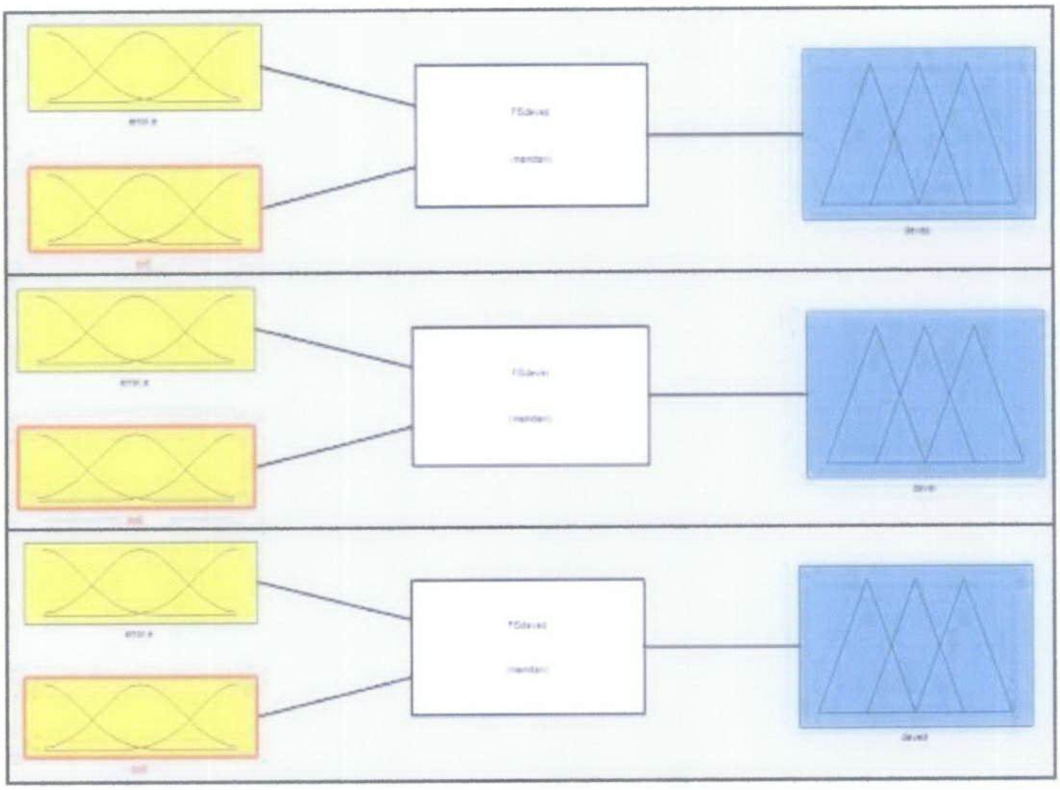


Figure 22: Inputs, Fuzzy Inference Engine and Output for three signals devP, devI and devD

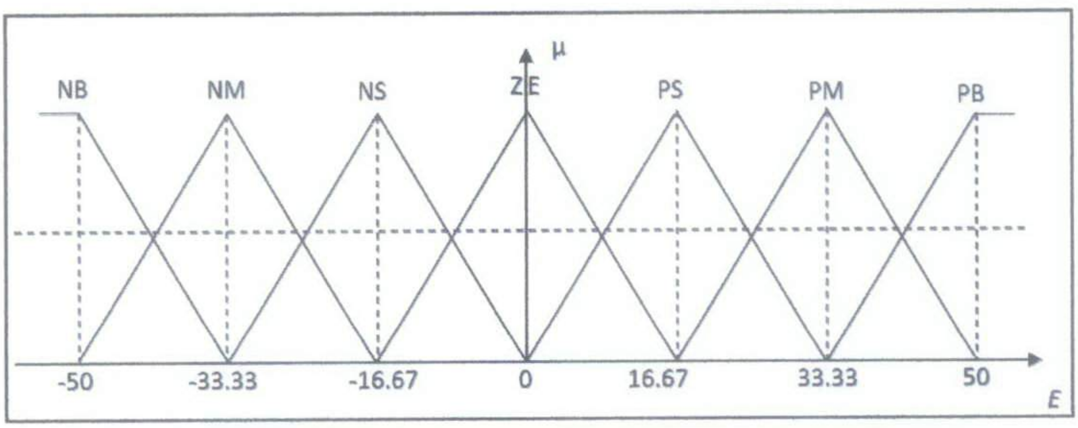


Figure 23: MF's for error, E

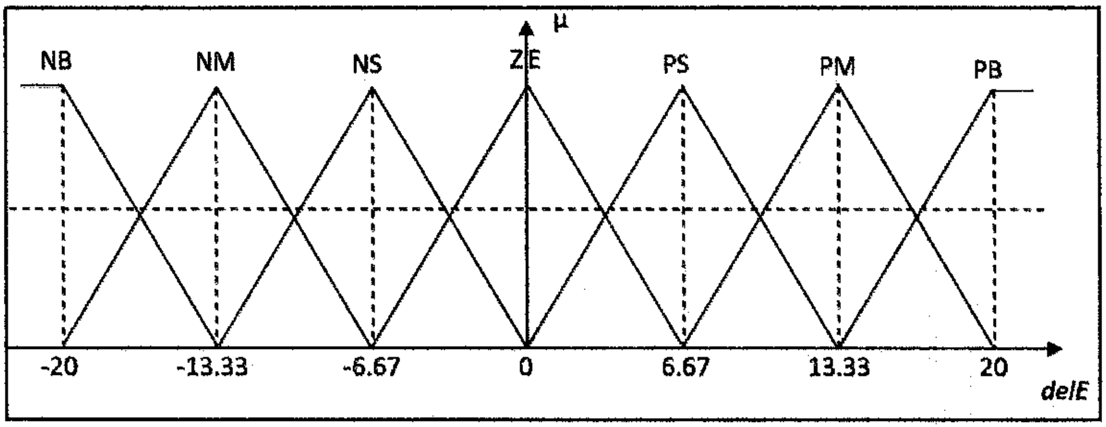


Figure 25: MF's for del E

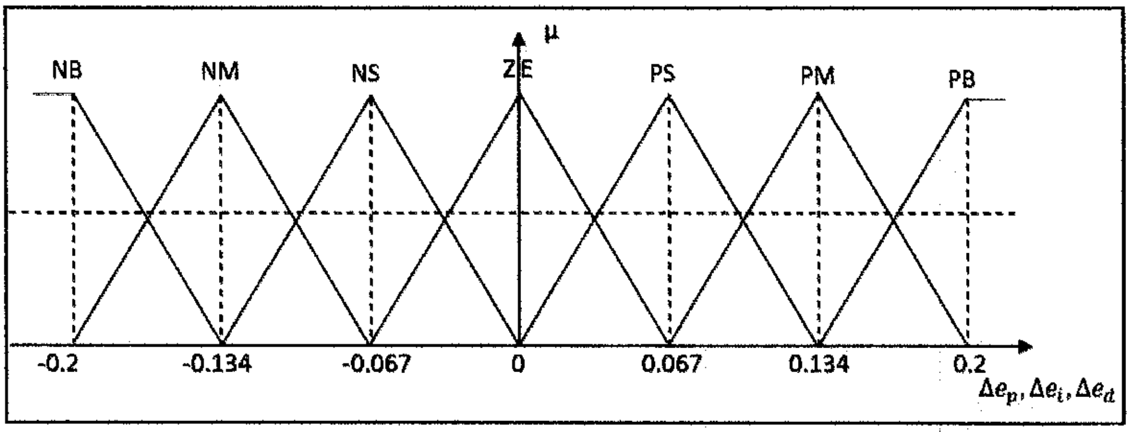


Figure 24: MF's for Output

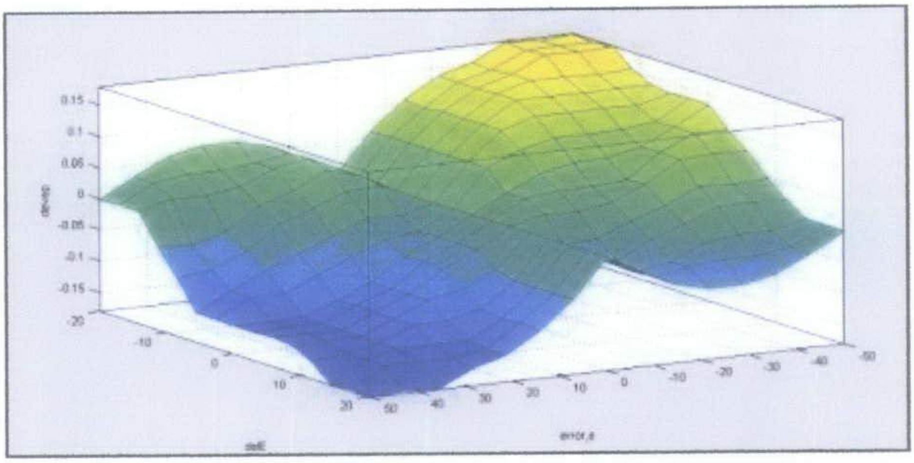


Figure 26: Surface view of FIS for devP

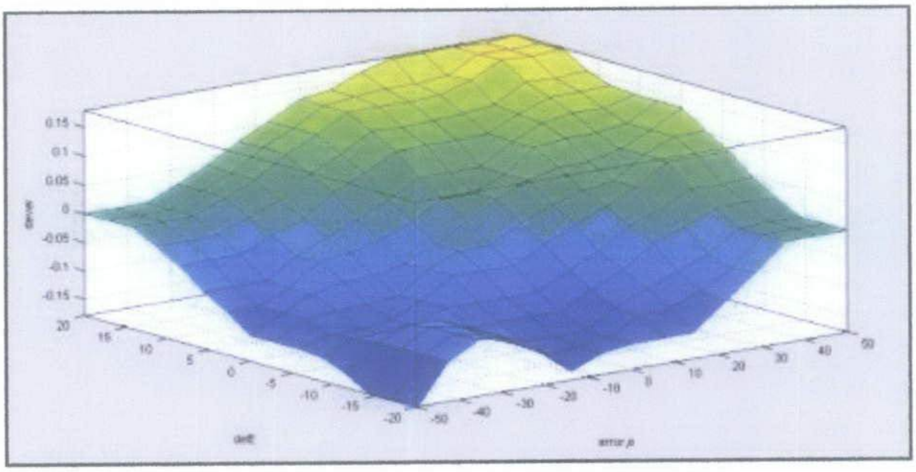


Figure 27: Surface view of FIS for devI

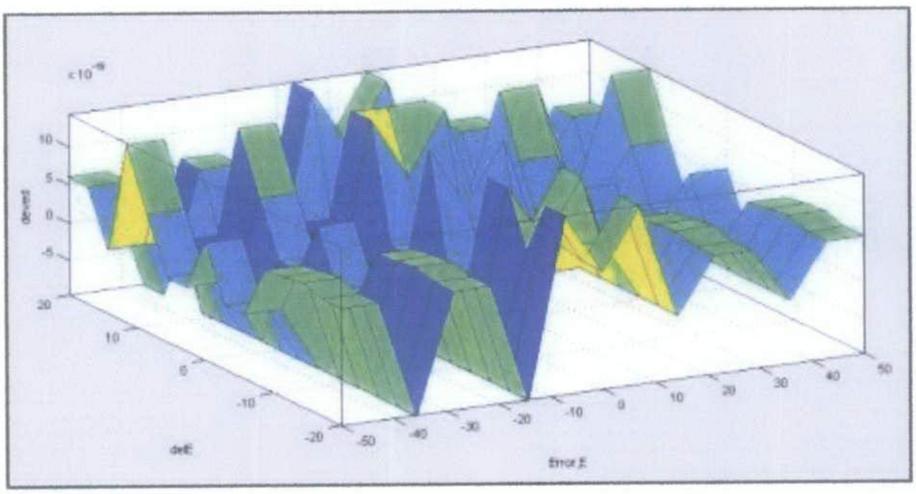


Figure 28: Surface view for devD

4.7 Controller Implementation on Pilot Plant

The aim for this section is to show that the AFPIDC developed can be used to control the operation of a pilot plant and compare its control performance to both conventional PID and Fuzzy Logic Controller. The implementation is done via two data acquisition card USB1208FS (by Measurement Computing) connected between laptop to Control Valve as output and from Orifice Flow Meter as input. The flow control operation will be done by using MATLAB/Simulink with Data Acquistion, Fuzzy Logic and PID Controller toolboxes.

4.7.1 Setting for Real Time Application using USB1208FS DAQ card for MATLAB/Simulink

In order to have real time control application using USB1208FS DAQ cards, firstly, on the MATLAB/Simulink, the settings for the Configuration Parameters (press CTRL + E) are as shown in Figure 29.

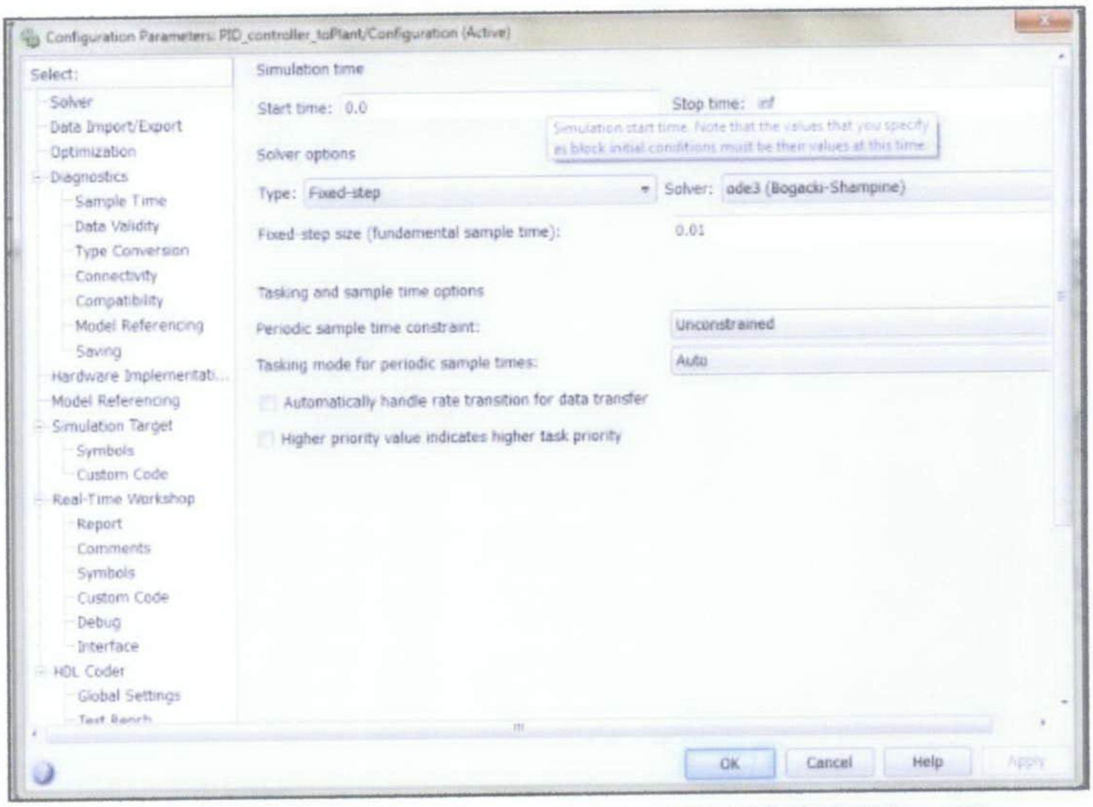


Figure 29: Configuration Parameters for USB1208FS DAQ card in MATLAB/Simulink

Then, in the MATLAB/Simulink, from the Data Acquisition Toolbox, the Analog Output and Analog Input are set up as in Figure 30 and 31. The Analog Output will give signal to the DAQ card that is connected to the Control Valve and for the Analog Input, it will receive signal from the DAQ card that is connected to the Flow Meter and will be converted to volume flow rate units which is liter per minute.

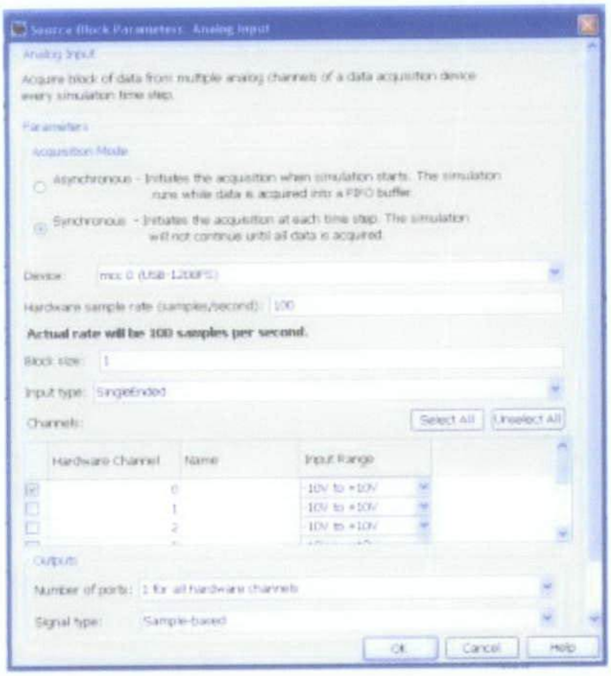


Figure 31: Analog Input toolbox

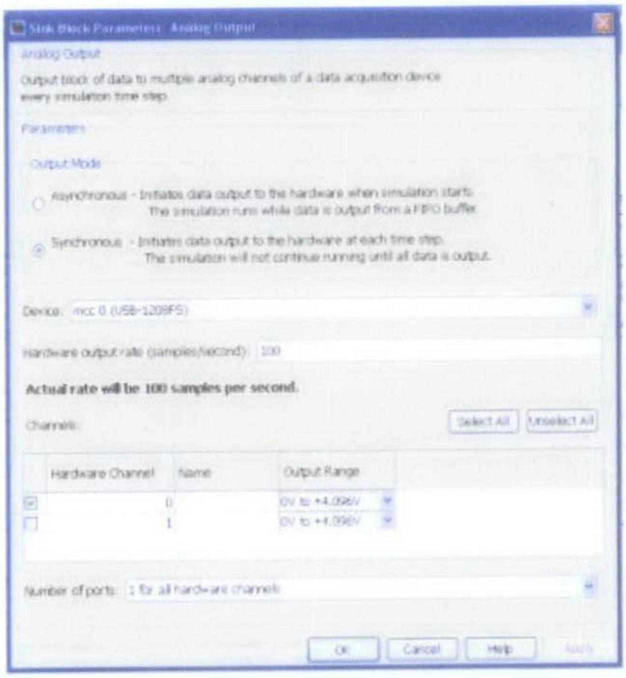


Figure 30: Analog Output Toolbox

Some of the features of the Simulink model used for Flow Control application are:

- DCS- like HMI feature with scopes to show real time data for PV, MV and SP
- User friendly approach where the operator can input the set point and change from auto to manual easily
- Engineering Units for SP and PV are used in liter per min and MV with percentage of opening
- Data can be collected over a very long period of time and saved to the workspace or file.
- Gaussian filter used to filter noise in the signal obtained from Flow Meter.
- For AFPIDC, the controller includes a memory block to act as an accumulator that adds up the controller output over time.

4.7.2 Simulink Model for Controller Implementation

For this project, the Simulink model for three types of controllers which are PID, Fuzzy Logic and Adaptive Fuzzy PID Controller are built as shown as in Figure 32 and 33. The model will show how the PID, FLC and AFPIDC will be used to control the pilot plant. For details of each controller Simulink models refer to *Appendix C –Simulink Model Subsytems.*

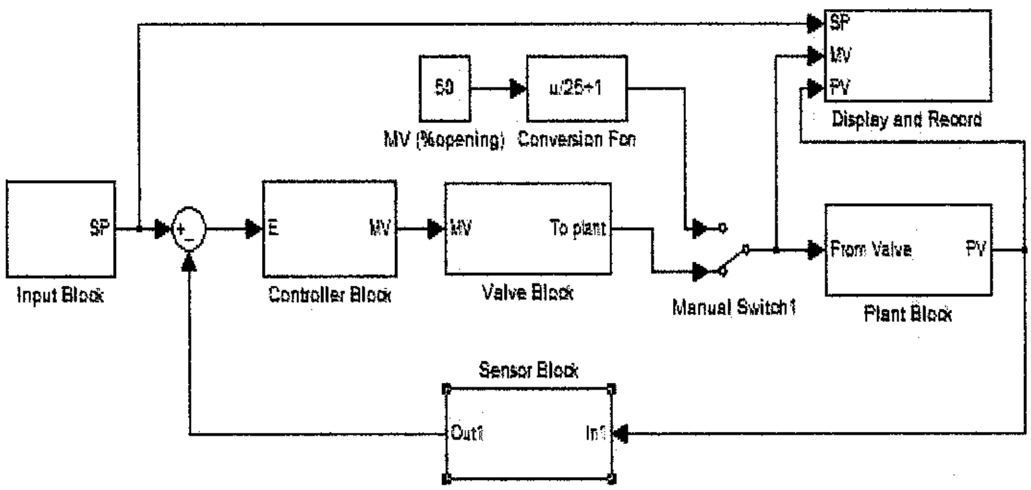


Figure 32: Simulink model for PID and AFPIDC Controller Implementation

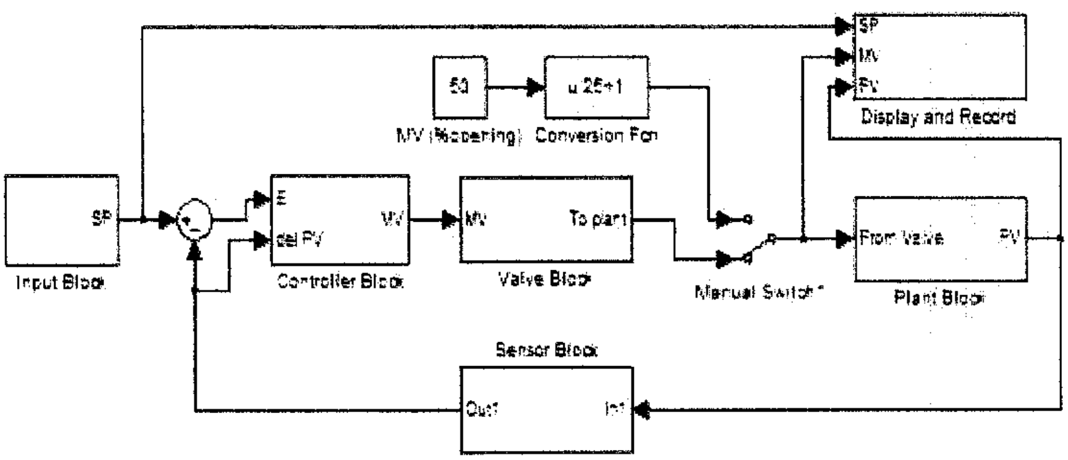


Figure 33: Simulink model for Fuzzy Controller Implementation

A. Results and Findings

To evaluate the robustness, these controllers have been subjected to random set point changes using the "uniform random number" block in Simulink. Since the operating range of the plant is set from 15 *l/min* to 45 *l/min.* with an interval of 20 seconds. This shall give the controller ample time to settle to steady state with each set point changes. Below are the results for three controllers for random set point changes.

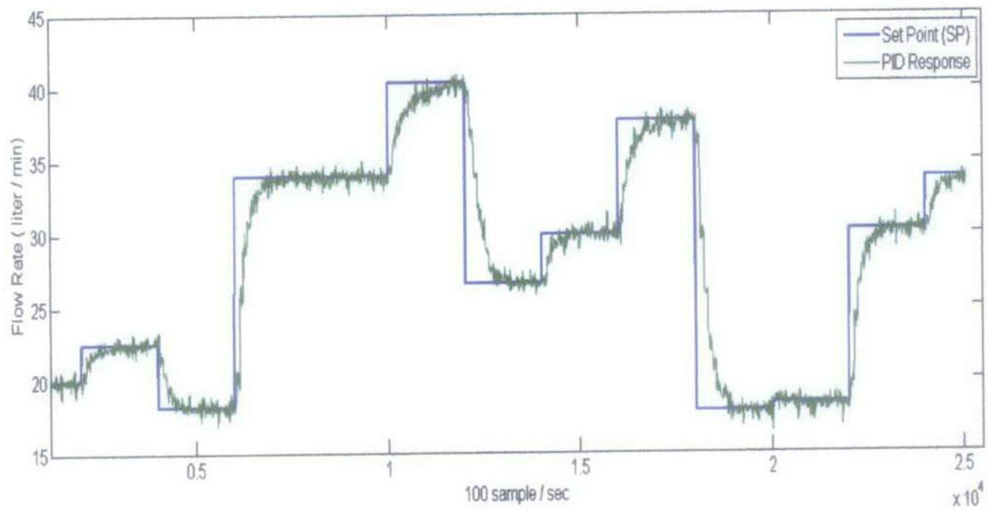


Figure 34 : PID Response with Random Set Point changes

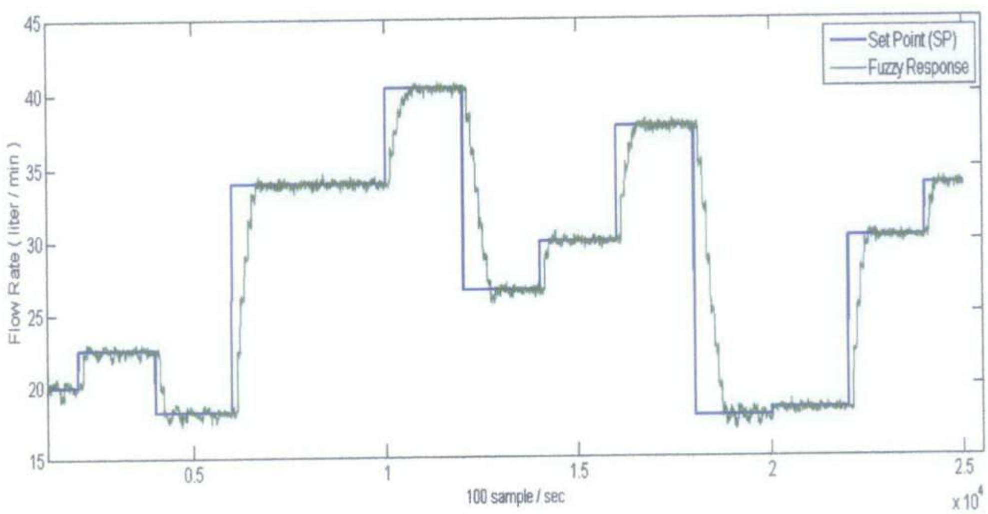


Figure 35 : Fuzzy Response with Random Set Point changes

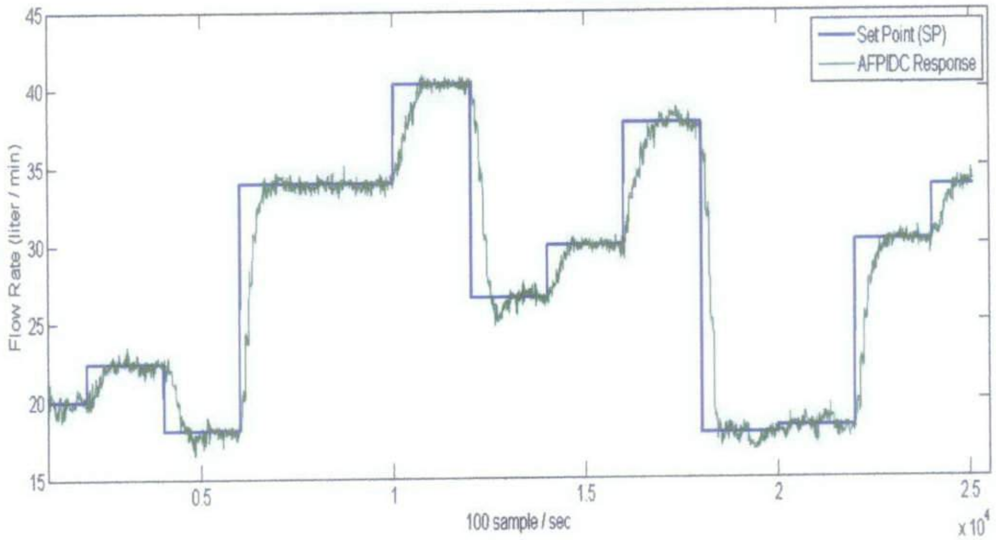


Figure 36 : AFPIDC Response with Random Set Point Changes

From the result of each controller's response to random set point change, it is obvious that the best controller is Fuzzy Logic controller followed by AFPIDC and PID controller. Although there are some overshoot can be seen for AFPIDC, however it is justifiable as the overshoot is only 2.5%. Generally, Fuzzy Logic response will give quicker response in terms of rise times and settling times with zero steady state error is better than PID controller. So, with the help of Fuzzy reasoning used to auto tune the initial PID gain in AFPIDC, it can be seen that it gives better control performances than PID controller with the same initial gain in Figure 34.

A more detailed study of the control performance is explained using the step change set point as input. All three of the controllers are subjected to the same conditions and a set point change from 25 *l/min* to 40 *l/min* is introduced to the system. The responses of all three controllers are shown in the Figure 37, 38 and 39 below.

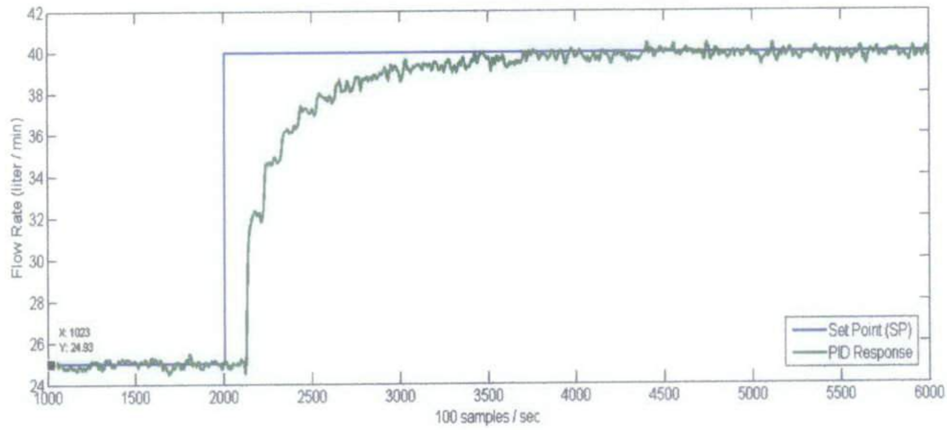


Figure 37 : PID Response to a Step change

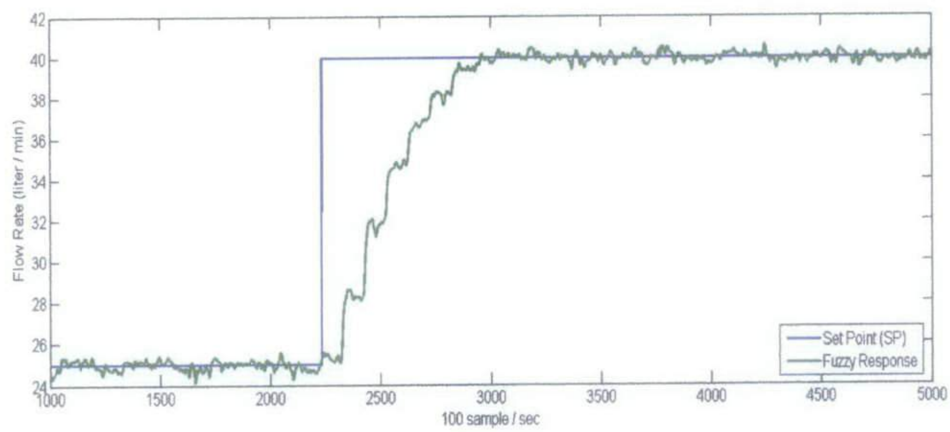


Figure 38 : Fuzzy Response to a Step change

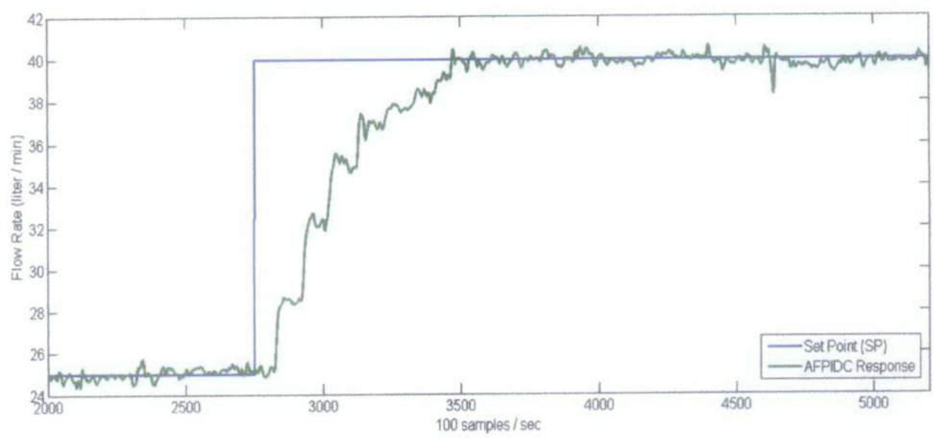


Figure 39 : AFPIDC Response to a Step change

From the result shown in Figure 37, 38 and 39, the control performance can be studied where the AFPIDC gives a better performance than PID controller. This is due to the AFPIDC gives a better settling time, smaller integral of absolute error (IAE), smaller MV overshoot.

Table 8: Controller Performance Comparison

Control Performance	PID	FUZZY	AFPIDC
IAE	Intermediate	Very Small	Small
Rise Time	14.21 s	4.23 s	7.17 s
Settling Time	14.21 s	4.23 s	7.17 s
Decay Ratio	0.00	0.00	0.00
Steady State Error	Acceptable	0.00%	Acceptable
% Overshoot	0.00%	0.00%	0.00%

However, Fuzzy Logic controller is the best controller between the three controllers implemented onto the pilot plant.

The reasons why Fuzzy has better control performances than AFPIDC are because:

- AFPIDC still has the PID characteristics as the output of summation between adaptive PID gain [$\Delta e_p(t)$,$\Delta e_i(t)$,$\Delta e_d(t)$] with initial PID gain will undergone to the proportional, integrate and derivative function.
- Fuzzy Reasoning is used to auto tune the PID gain not as main output of a controller to the control valve.

But from this result analysis, it is proven that by using the Fuzzy Logic Reasoning, it can automatically tuned the PID parameter to give better control performance than conventional PID controller. This is one of the advantages of Adaptive Fuzzy PID Controller (AFPIDC) that combines two controllers which is PID controller with the Fuzzy Logic Controller. It will provide faster response and automatically tuned the PID parameter gains to improve conventional PID controller.

B. Bubble Noise Effect

One of the disturbances that normally affect the flow meters is the Bubble Noise disturbances. Bubble Noise is caused by the superposition of single pulses generated by individual bubbles which are randomly distributed in the liquid [23] and [24]. Bubbles will be produced in a process line by inserting air using a small tube that has several small holes drilled on the tube in Buffer Tank VE 100 [refer Figure 9]. The air feed on the bottom of the tank will produced a bubble and some of them will escaped along the pipe line of the process flow between Tank VE 100 to VE 200. These bubbles will cause the flow measurement from the flow meter to be fluctuated and erroneous rapidly.

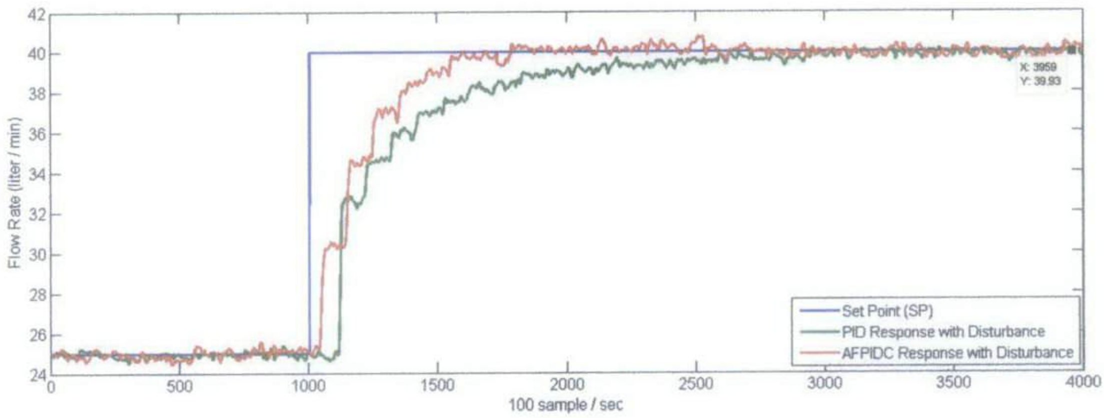


Figure 40: PID and AFPIDC Controller Performances with Disturbance

Table 9: Controller Performance with Disturbance

Control Performance	PID	AFPIDC
IAE	Intermediate	Small
Rise Time	16.90 s	7.81 s
Settling Time	16.90 s	7.81 s
Decay Ratio	0.00	0.00
Steady State Error	Acceptable	Acceptable
% Overshoot	0.00%	0.00%

Figure 40 shows the controller performances under bubbling noise disturbances for PID controller and AFPIDC. AFPIDC response exhibits no overshoot and provides faster response than PID controller under bubbling noise disturbances based on results in Table 4. The rise time and settling time for PID controller has increased for 2.7 seconds meanwhile for AFPIDC, they have only increased 0.64 seconds. Thus, AFPIDC provides stable controller response than PID controller with disturbances in the process plant.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Adaptive Fuzzy PID Controller (AFPIDC) is a new hybrid controller that uses the Fuzzy Logic reasoning to automatically tune the PID gain so that the controller will have better control performances than the conventional PID controller. The advantages of a Fuzzy based controller over PID controller are derived from the implementation results onto the pilot plant given random and step input changes. Better control performances, robustness and overall stability can be expected from the AFPIDC with the usage of Fuzzy Logic reasoning to automatically tune the initial PID gain. However, Fuzzy controller alone gives better control performances than AFPIDC as AFPIDC still have the characteristics of PID controller due to its circuit design and FIS tuning.

The development of this project is also parallel with industrial needs and applications; therefore AFPIDC can be implemented on to the real industry process to be used together with conventional PID controller. Furthermore, the DCS-HMI developed is simpler, easier and more economical to be used compared to the standard DCS architecture. The objectives of this project which are to develop an Adaptive Fuzzy PID Controller (AFPIDC), implement onto the *PcA SimExpert Mobile Pilot Plant SE231B-21 Flow Control and Calibration Process Unit*, develop a DCS- HMI system and to compare the controller performances between PID, Fuzzy and AFPIC controllers have been achieve successfully.

5.2 Recommendations

In the future, other methods of hybrid controller (PID with Fuzzy controller) can be explored such as both outputs are summed together to act as one controller (PID- Fuzzy Controller). Other controllers such as Artificial Neural Network (ANN), Model Predictive (MPC) and NeuroFuzzy controllers can improve the control performances for flow control applications. Last but not least, since for this project, the flow meter used for all three controllers is Orifice Flow Meter, then the performance of all three controllers can be implemented using Vortex Flow Meter and compare with other flow meters.

REFERENCES

[1] Roland S. Burns, "Advanced Control Engineers," Historical Review of Control Engineering, Butterworth Heinemann, 2001, pp 1-3.

[2] Curtis D. Johnson, "Process Control Instrumentation Technology", 7th Edition, Prentice Hall, 2003, pp 6 - 7

[3] C.H Chou, H.C. Lu, "Real time fuzzy controller design for hydraulic servo system", International Journal Computer Industries 22, 1993, pp 129 - 142

[4] S. Bennet, "A history of control engineering 1930 - 1955", IEE Control Engineering Series 47, 1993, pp 31 - 39

[5] J.G. Ziegler, N.B Nichols, "Optimum settings for automatic controllers", Trans American Society of Mechanical Engineers, 64, 1942, pp 759 - 768

[6] Uday A. Bakshi, Mayuresh V. Bakshi, " Modern Control Theory", Technical Publications Pune, 2009, pp 5 - 6.

[7] Jia Lei, Lv Hongli, Wenjian Cai, "Model Predictive Control Based on Fuzzy Linearization Technique For HVAC Systems Temperature Control," Industrial Electronics and Applications, 2006 1ST IEEE Conference, pp 1-5, 24-26 .

[8] Yong Zai Lu," Industrial Intelligent Control, Fundamentals and Applications", John Wiley & Sons publications, 1996, pp 27 - 28.

[9] Zadeh, L.A., "Knowledge Representation in Fuzzy Logic," IEEE Transactions on Knowledge and Data Engineering, Vol. 1, No 1, Mar 1989, pages 89 – 100.

[10] Robert Babuska, " An overview of nonlinear identification and control with fuzzy systems", IEE Control Engineering Series 70,2005 , pp 1 - 36.

[11] A. Filippidis, L.c. Jain and C.W. de Silva, "Intelligent Control Techniques", Intelligent Adaptive Control: Industrial Applications, CRC Press, 1999, pp 1 - 24.

[12] X. Hong, C.J Harris, " Neuro - Fuzzy model contruction, design and estimation", Intelligent Control Systems using Computational Intelligence Techniques, The Institution of Electrical Engineers, 2005, pp 219 - 252.

[13] Roland S. Burns,"Advanced Control Engineering", Intelligent Control System Design, Butterworth Heinemann 2001, pp 325 - 379.

[14] Fakhreddine O. Karray, Clarence W. de Silva,"Learning and Adaption in Complex Dynamic Systems", Intelligent Adaptive Control: Industrial Applications, CRC Press, 1999, pp 25 - 40.

[15] Javier Aracil, Francisco Gordillo," Describing function method for stability analysis of PD and PI Fuzzy Controllers", Fuzzy Sets and Systems Volume 143, Issue 2, 16 April 2004, pp 233-249

[16] Wei Li, "Design of a hybrid fuzzy logic proportional plus conventional integral-derivative controller," Fuzzy Systems, IEEE Transactions on , vol.6, no.4, pp.449-463

[17] Rajani K Mudi, Nikhil R. Pal, " A self tuning Fuzzy PI controller", Fuzzy Sets and Systems 115, 2000, pp 327 - 338.

[18] D. Driankov, H. Hellendoorm and M. Reinfrank. "An Introduction to Fuzzy Control", Springer, Berlin (1993) pp 197 – 231.

[19] H.A. Malki, H. Li, G. Chen, New design and stability analysis of fuzzy proportional-derivative control systems, IEEE Trans. Fuzzy Systems 2 (4) ,1994, pp 245 - 254.

[20] Yuanhui Yang, Wailing Yang, Mingchun Wu, Qiwen Yang, Yuncan Xue,"A new type of adaptive fuzzy PID controller," Intelligent Control and Automation (WCICA), 2010 8th World Congress on, pp 5306-5310

[21] Tang, K.S., Kim Fung Man, Guanrong Chen, Kwong, S.,"An optimal fuzzy PID controller," Industrial Electronics, IEEE Transactions on, vol.48, no.4, pp 757-765

[22] Marlin, T.E., " Process Control: Designing Processes and Control Systems for Dynamic Performance", Mc Graw Hill International, 200, pp 889 – 894

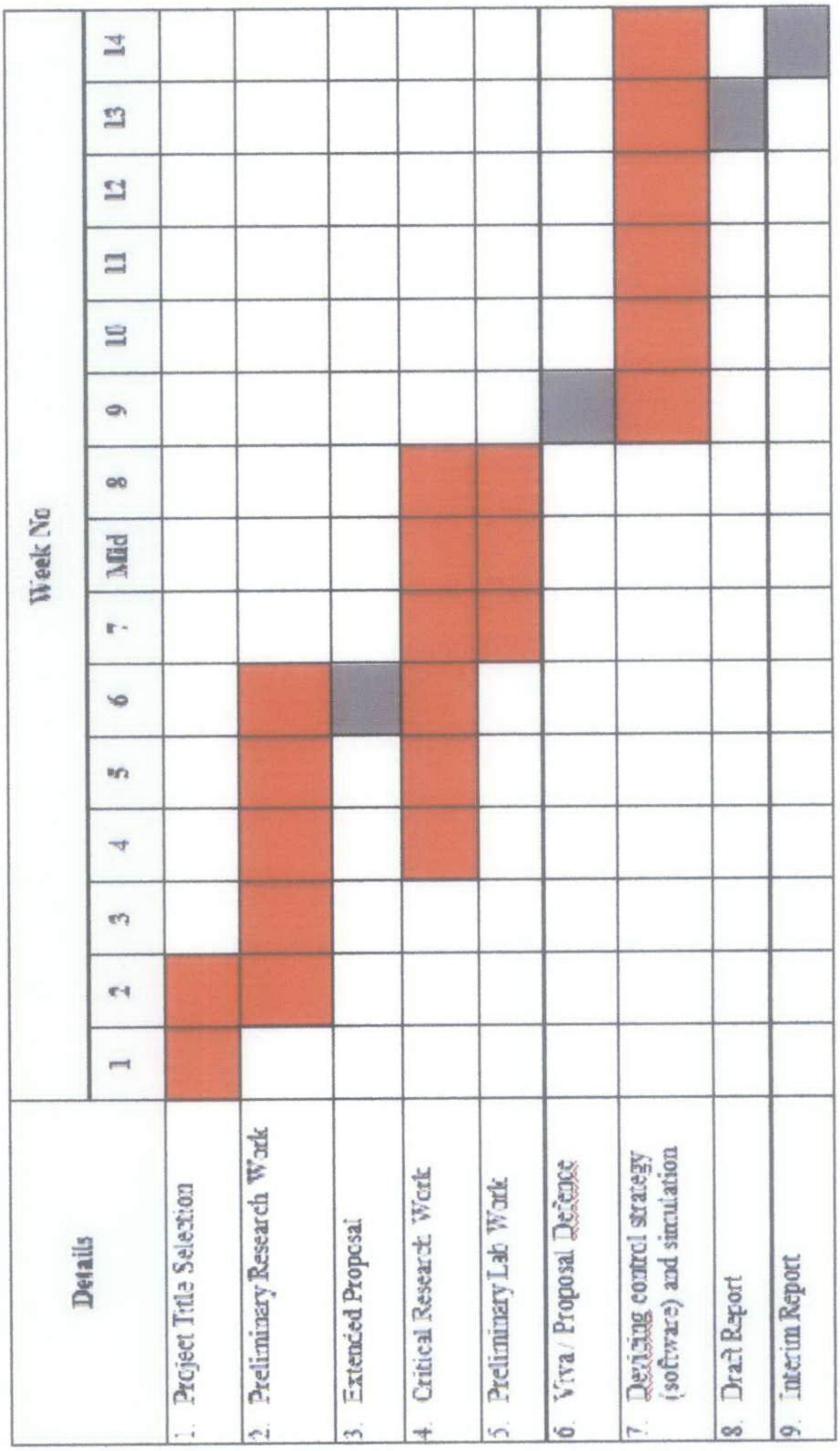
[23] Pathmanathan, E.; Ibrahim, R.; , "Development and implementation of Fuzzy Logic Controller for Flow Control Application," *Intelligent and Advanced Systems (ICIAS), 2010 International Conference on* , vol., no., pp.1-6, 15-17 June 2010

[24] Fuge R., Horst. L, "Bubble Noise at Magnetic Flow Meters", Nuclear Engineering and Design, vol 49-3,Sept 1978 pp 205-212

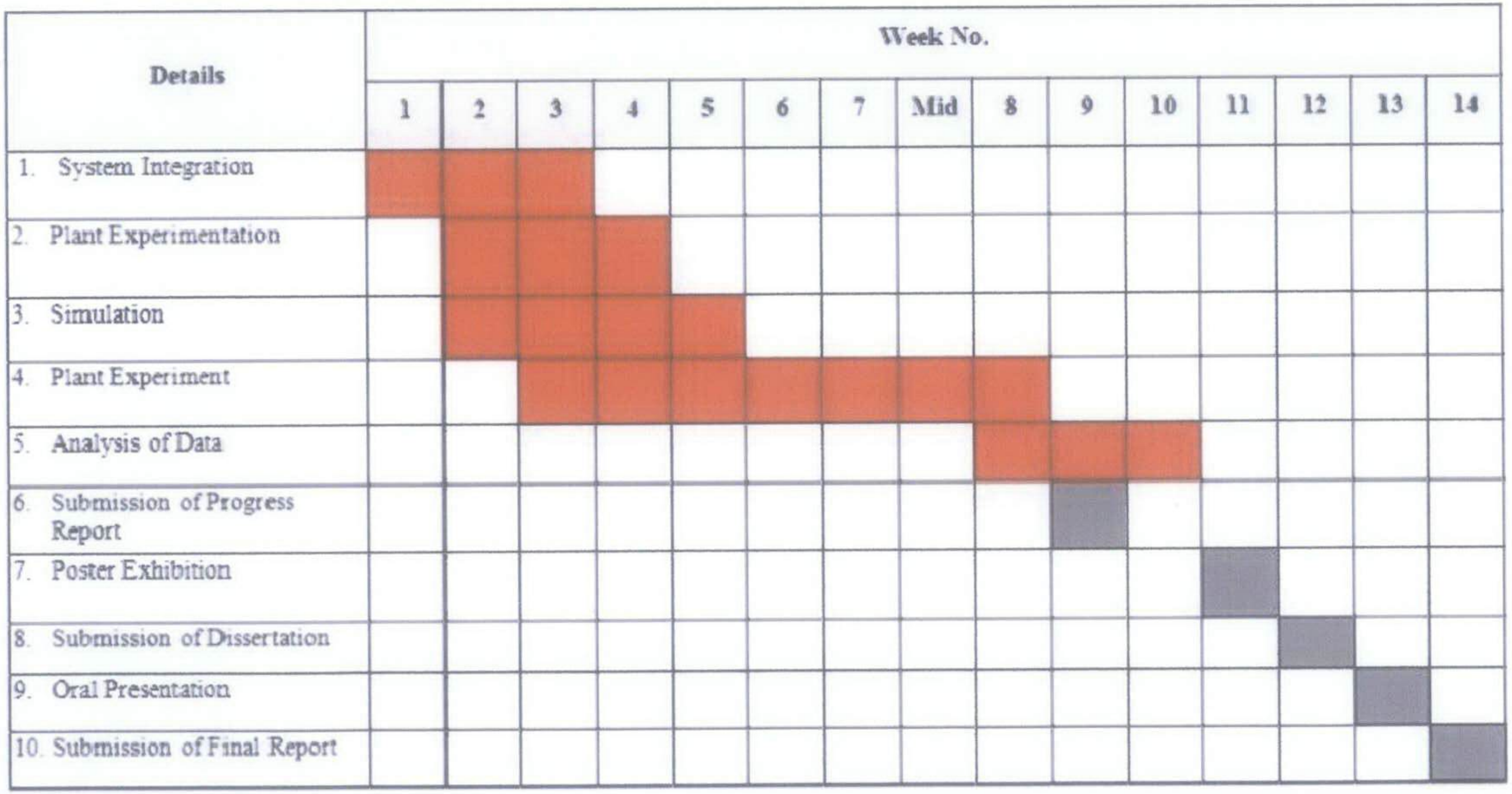
APPENDICES

APPENDIX A – Project Gantt Chart

Gantt chart for FYP 1



Gantt chart for FYP 2

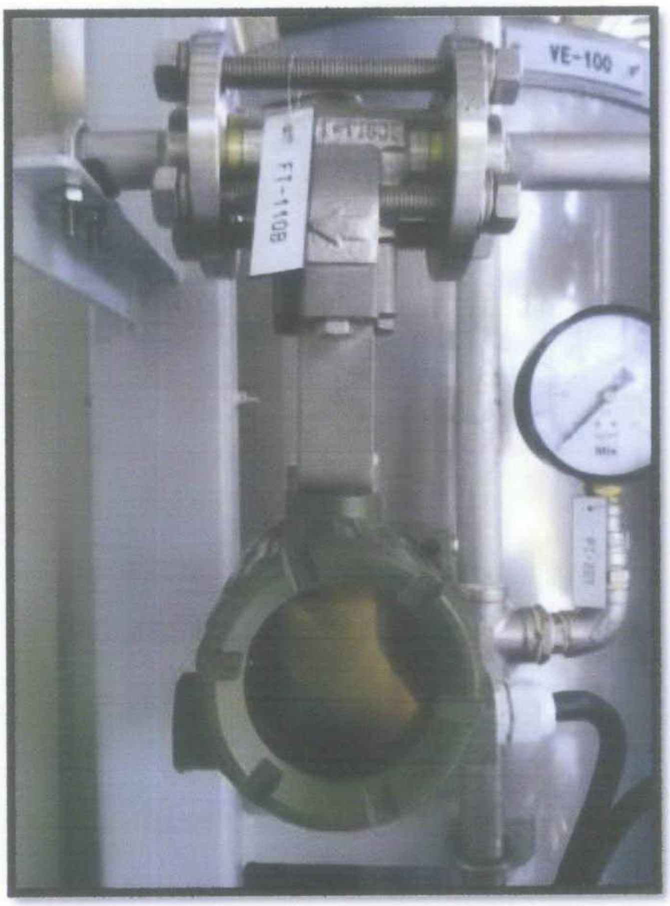


APPENDIX B - Plant Devices and Instruments

Pilot Plant

Line of flow of water from two tanks

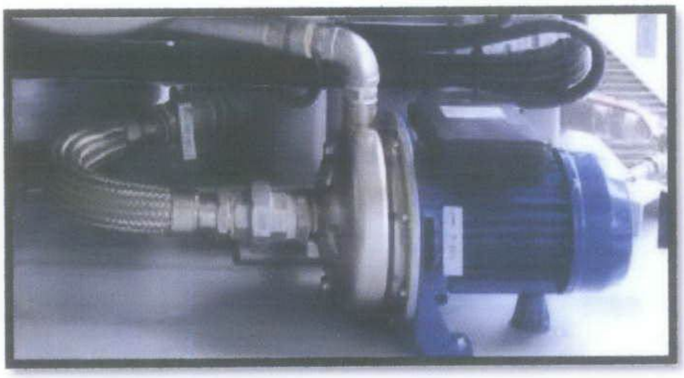
FT 110A – Coriolis Flow Meter



FT 110B – Vortex Flow Meter

FT 110C – Orifice Flow Meter

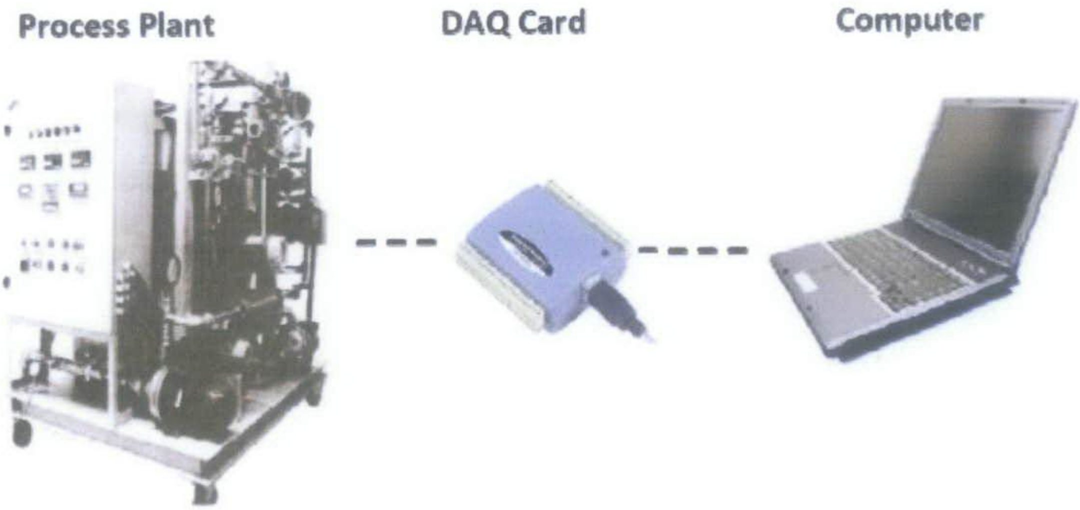
Control Valve

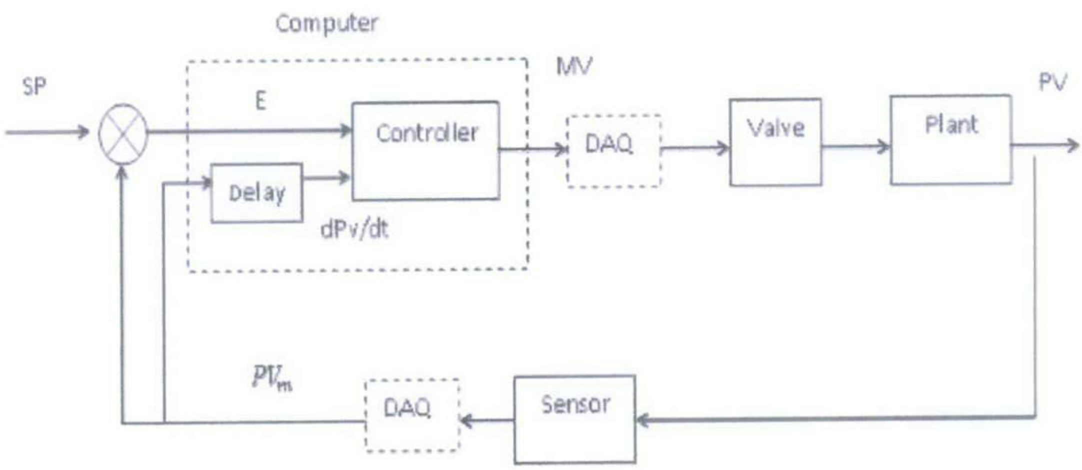


Liquid Transfer Pump

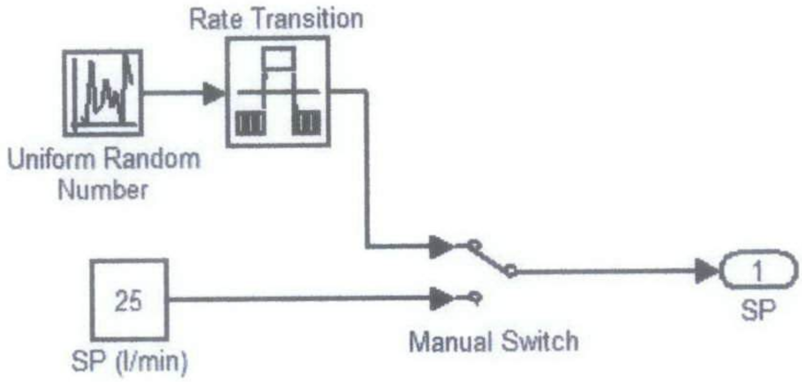
Front of Local Control Panel

APPENDIX C – Project Implementation Set Up

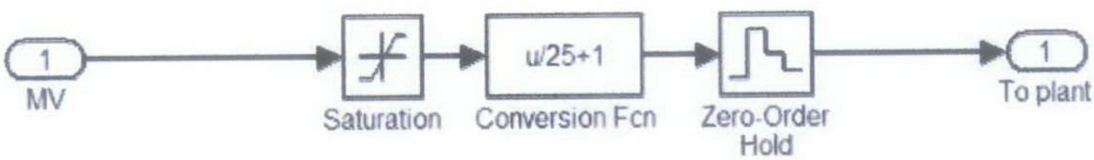




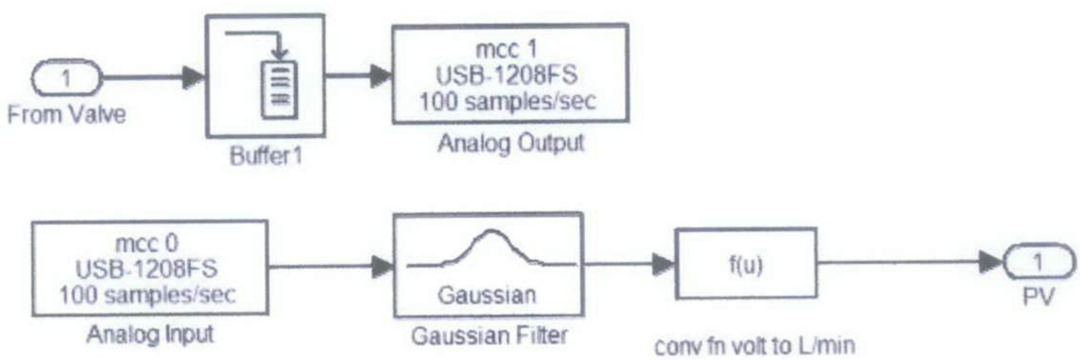
APPENDIX D – Simulink Model Subsystems



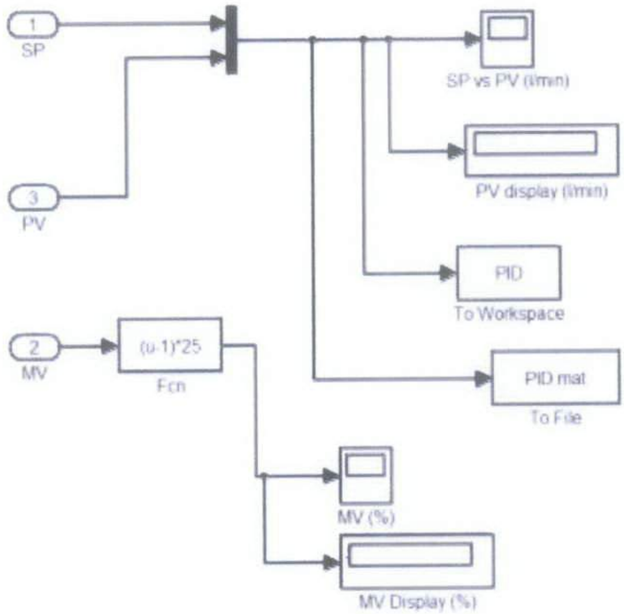
Input Subsystem



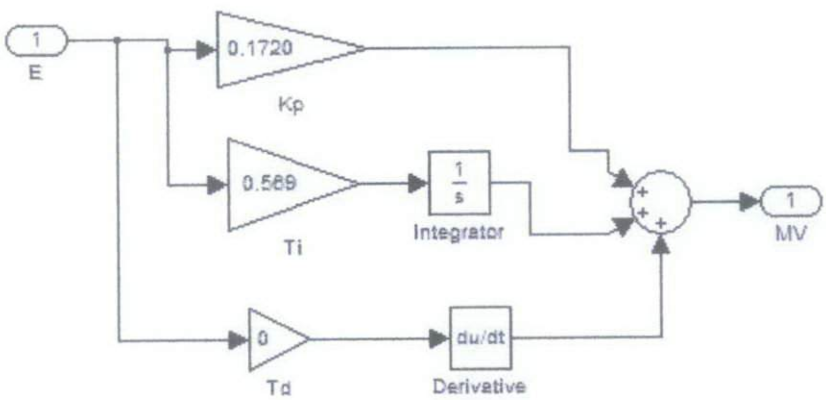
Valve Subsystem



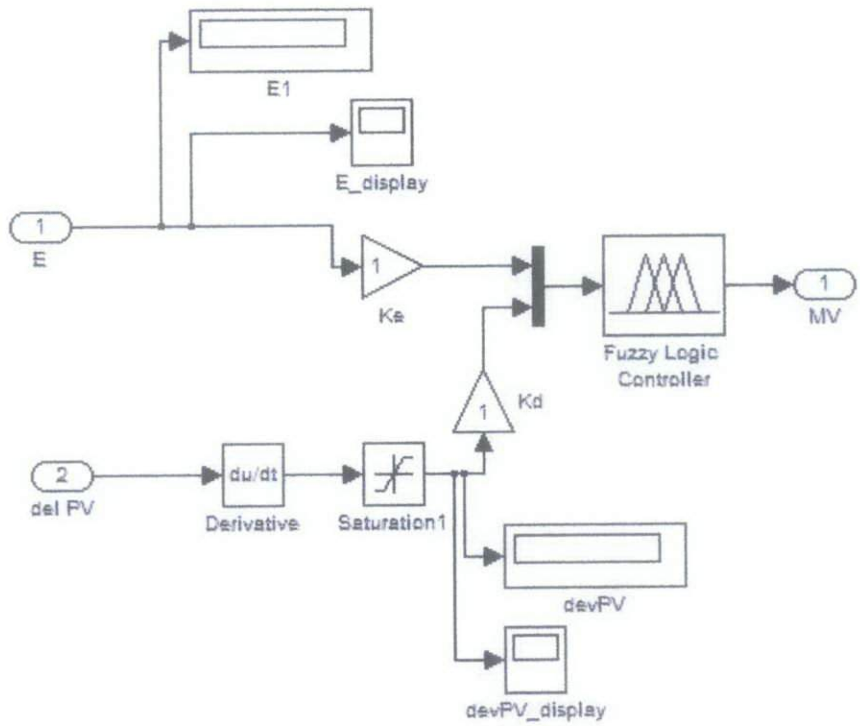
Plant Subsystem



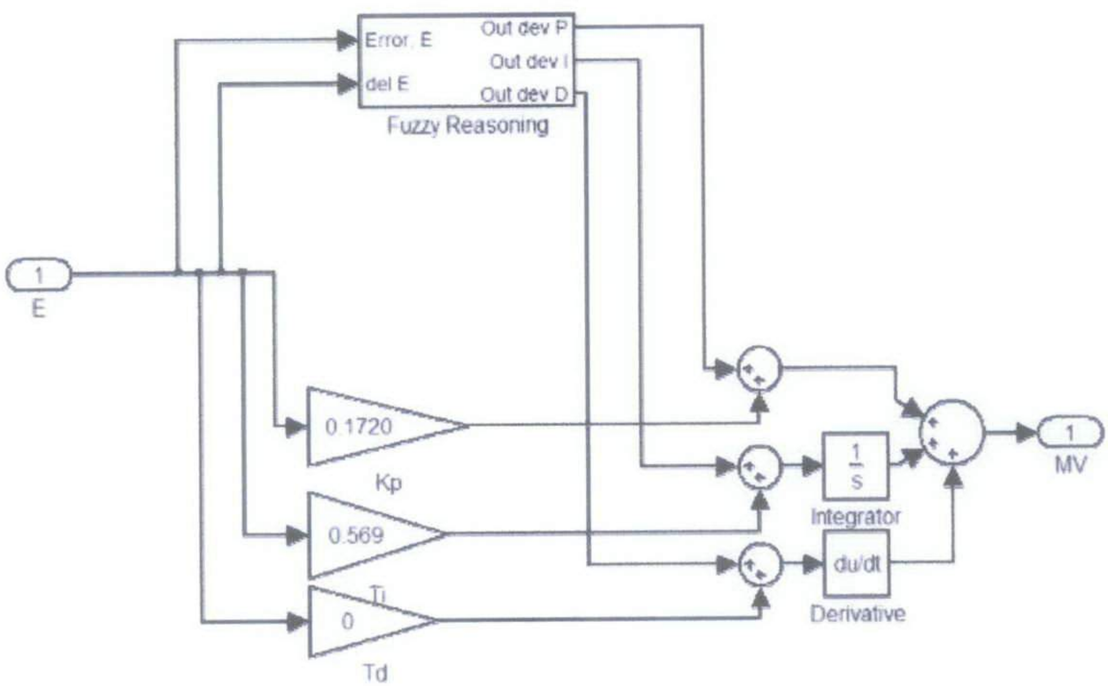
Display and Record Subsystem



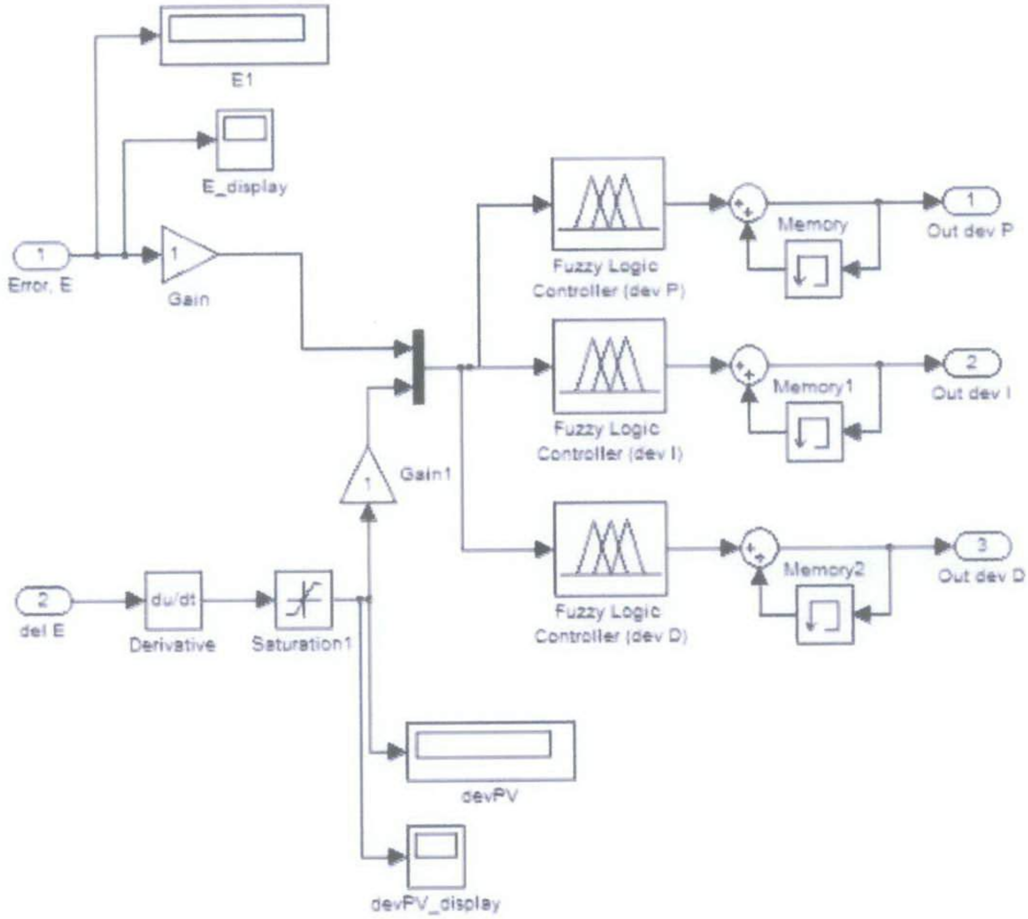
PID Controller Subsystem



Fuzzy Controller Subsystem

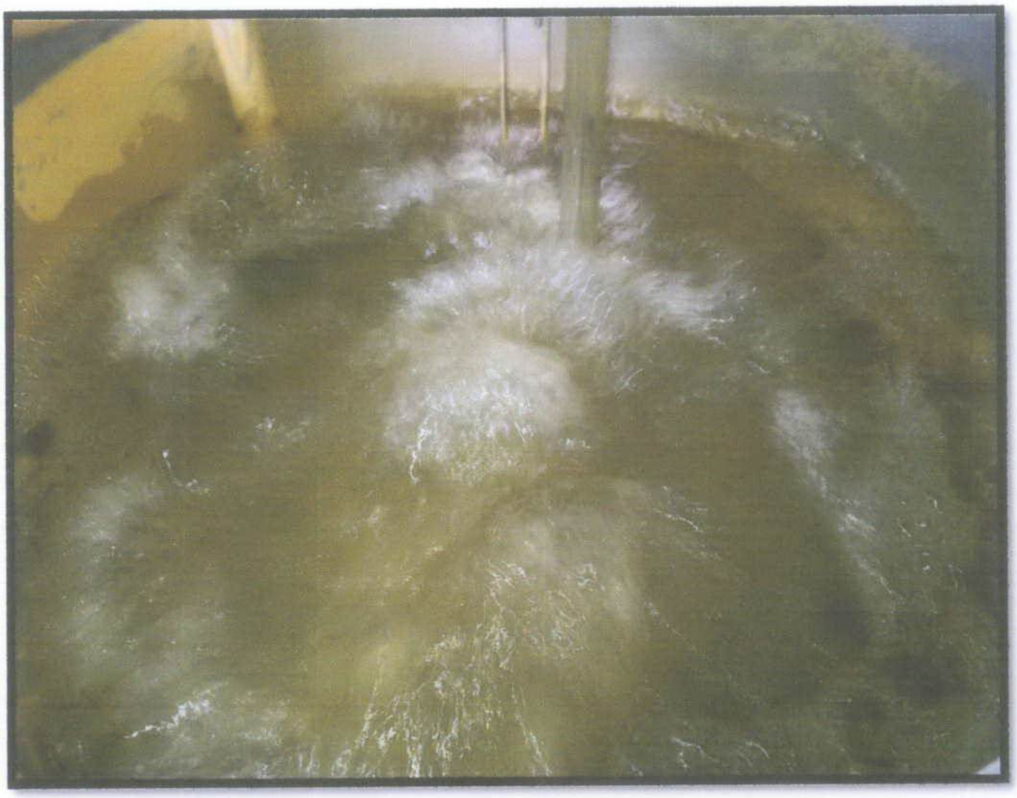


AFPIDC Controller Subsystem



Fuzzy Reasoning Subsytem

APPENDIX E – BUBBLING EFFECT DISTURBANCES SET UP



Bubbles Produces at the bottom of Tank VE100

APPENDIX F – USB1208FS Data Acquisition Card Specifications

USB1208FS Measurement Computing DAQ Card

Analog input

Table 1. Analog input specifications

Parameter	Conditions	Specification
A/D converter type		Successive approximation type
Input voltage range for linear operation, single-ended mode	CHx to GND	±10 volts (V) max
Input common-mode voltage range for linear operation, differential mode	CHx to GND	-10 V min, +20 V max
Absolute maximum input voltage	*CHx to GND*	*±28 V max*
Input impedance		*122KOhm*
Input current (Note 1)	Vin = +10 V	70 microamperes (µA) typ
	Vin = 0 V	-12 µA typ
	Vin = -10 V	-94 µA typ
Number of channels		8 single-ended / 4 differential, software selectable
Input ranges, single-ended mode		±10 V, G=2
Input ranges, differential mode		±20 V, G=1 ±10 V, G=2 ±5 V, G=4 ±4 V, G=5 ±2.5 V, G=8 ±2.0 V, G=10 ±1.25 V, G=16 ±1.0 V, G=20 Software selectable
Throughput (Note 2)	Software paced	250 samples per second (S/s) typ, PC-dependent
	Continuous scan	50 kilosamples per second (kS/s)
Channel gain queue	Up to 16 elements	Software configurable channel, range, and gain.
Resolution (Note 3)	Differential	12 bits, no missing codes
	Single-ended	11 bits
CAL accuracy	CAL = 2.5 V	±36.25 mV max
Integral linearity error		±1 least significant bit (LSB) typ
Differential linearity error		±0.5 LSB typ
Repeatability		±1 LSB typ
CAL current	Source	5 milliamperes (mA) max
	Sink	20 µA min, 100 µA typ
Trigger source	Software selectable	External digital: TRIG_IN
Pacer source	Software selectable	Internal External (SYNC), rising edge triggered Programmed IO

Note 1: Input current is a function of applied voltage on the analog input channels. For a given input voltage, V_{in}, the input leakage is approximately equal to $(8.181 * V_{in} - 12)$ µA.

Analog output

Table 8. Analog output specifications

Parameter	Conditions	Specification
Resolution		12-bits, 1 in 4096
Output range		*0 – 4.096 V, 1 mV per LSB.*
Number of channels		2
Throughput (Note 4)	Software paced	250 S/s single channel typical, PC dependent
	Single channel, continuous scan	10 kS/s
	Dual channel, continuous scan, simultaneous update	5 kS/s
Power on and reset voltage		Initializes to 000h code
Output drive	Each D/A OUT	15 mA
Slew rate		0.8V/microsecond (µs) typ

Note 4: Maximum throughput scanning to PC memory is machine dependent. The rates specified are for Windows XP only. Maximum rates on operating systems that predate XP may be less and must be determined through testing on your machine.

Digital input/output

Table 11. Digital I/O specifications

Digital type	CMOS
Number of I/O	16 (Port A0 through A7, Port B0 through B7)
Configuration	2 banks of 8
Pull up/pull-down configuration	All pins pulled up to Vs via 47K resistors (default). Positions available for pull down to ground. Hardware selectable via zero ohm (Ω) resistors as a factory option.
Input high voltage	2.0 V min, 5.5 V absolute max
Input low voltage	0.8 V max, –0.5 V absolute min
Output high voltage (IOH = -2.5 mA)	3.8 V min
Output low voltage (IOL = 2.5 mA)	0.7 V max
Power on and reset state	Input

APPENDIX G – USB1208FS Block Diagram and Pinouts

Main connector and pin out

Table 21. Main connector specifications

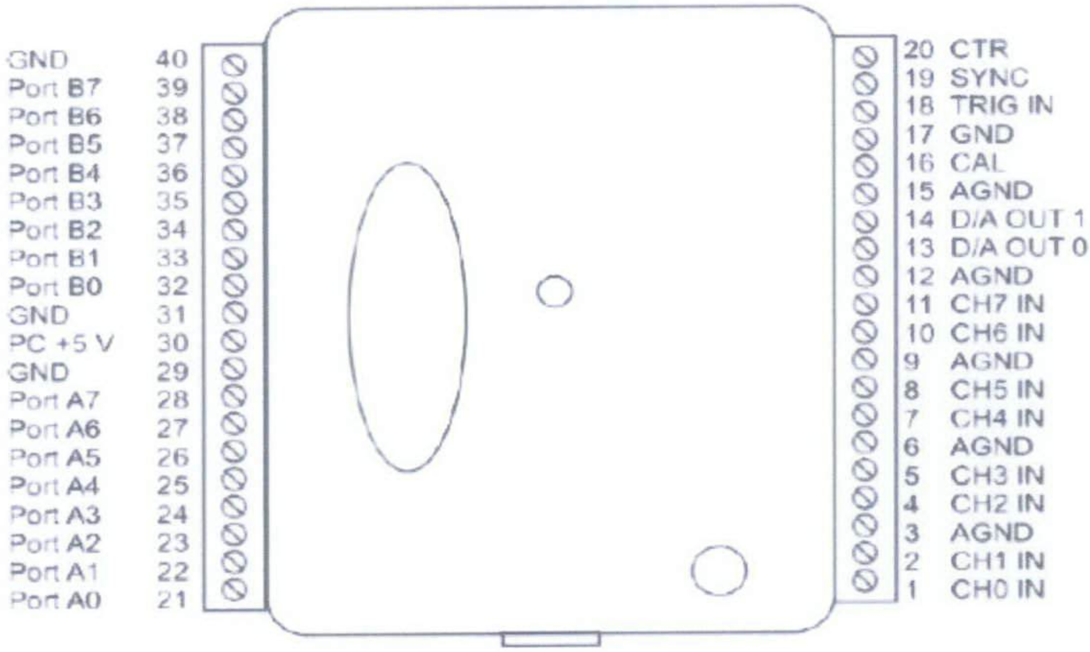
Connector type	Screw terminal
Wire gauge range	16 AWG to 30 AWG

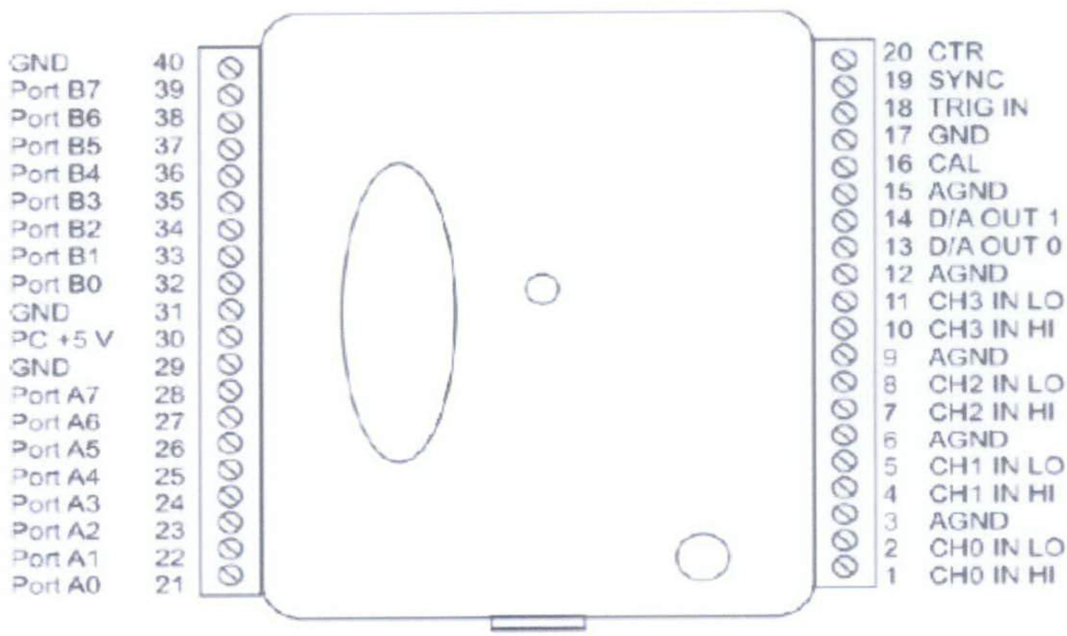
4-channel differential mode

Pin	Signal Name	Pin	Signal Name
1	CH0 IN HI	21	Port A0
2	CH0 IN LO	22	Port A1
3	AGND	23	Port A2
4	CH1 IN HI	24	Port A3
5	CH1 IN LO	25	Port A4
6	AGND	26	Port A5
7	CH2 IN HI	27	Port A6
8	CH2 IN LO	28	Port A7
9	AGND	29	GND
10	CH3 IN HI	30	PC+5V
11	CH3 IN LO	31	GND
12	AGND	32	Port B0
13	D/A OUT 0	33	Port B1
14	D/A OUT 1	34	Port B2
15	AGND	35	Port B3
16	CAL	36	Port B4
17	GND	37	Port B5
18	TRIG_IN	38	Port B6
19	SYNC	39	Port B7
20	CTR	40	GND

8-channel single-ended mode

Pin	Signal Name	Pin	Signal Name
1	CH0 IN	21	Port A0
2	CH1 IN	22	Port A1
3	AGND	23	Port A2
4	CH2 IN	24	Port A3
5	CH3 IN	25	Port A4
6	AGND	26	Port A5
7	CH4 IN	27	Port A6
8	CH5 IN	28	Port A7
9	AGND	29	GND
10	CH6 IN	30	PC+5V
11	CH7 IN	31	GND
12	AGND	32	Port B0
13	D/A OUT 0	33	Port B1
14	D/A OUT 1	34	Port B2
15	AGND	35	Port B3
16	CAL	36	Port B4
17	GND	37	Port B5
18	TRIG_IN	38	Port B6
19	SYNC	39	Port B7
20	CTR	40	GND


GND 40
Port B7 39
Port B6 38
Port B5 37
Port B4 36
Port B3 35
Port B2 34
Port B1 33
Port B0 32
GND 31
PC +5 V 30
GND 29
Port A7 28
Port A6 27
Port A5 26
Port A4 25
Port A3 24
Port A2 23
Port A1 22
Port A0 21
20 CTR
19 SYNC
18 TRIG IN
17 GND
16 CAL
15 AGND
14 D/A OUT 1
13 D/A OUT 0
12 AGND
11 CH7 IN
10 CH6 IN
9 AGND
8 CH5 IN
7 CH4 IN
6 AGND
5 CH3 IN
4 CH2 IN
3 AGND
2 CH1 IN
1 CH0 IN


GND 40
Port B7 39
Port B6 38
Port B5 37
Port B4 36
Port B3 35
Port B2 34
Port B1 33
Port B0 32
GND 31
PC +5 V 30
GND 29
Port A7 28
Port A6 27
Port A5 26
Port A4 25
Port A3 24
Port A2 23
Port A1 22
Port A0 21
20 CTR
19 SYNC
18 TRIG IN
17 GND
16 CAL
15 AGND
14 D/A OUT 1
13 D/A OUT 0
12 AGND
11 CH3 IN LO
10 CH3 IN HI
9 AGND
8 CH2 IN LO
7 CH2 IN HI
6 AGND
5 CH1 IN LO
4 CH1 IN HI
3 AGND
2 CH0 IN LO
1 CH0 IN HI

APPENDIX H – Cohen Coon Open Loop Correlations

Control Modes	Parameters
P - only	$K_c = \left(\frac{1}{RK_p}\right)(1 + \frac{R}{3})$
P + I only	$K_c = \left(\frac{1}{RK_p}\right)(\frac{9}{10} + \frac{R}{12})$
	$T_i = \frac{\theta(30 + 3R)}{9 + 20R}$
P + I + D	$K_c = \left(\frac{1}{RK_p}\right)(\frac{4}{3} + \frac{R}{4})$
	$T_i = \frac{\theta(32 + 6R)}{13 + 8R}$
	$T_D = \theta\frac{4}{11 + 2R}$
P + D only	$K_c = \left(\frac{1}{RK_p}\right)(\frac{5}{4} + \frac{R}{6})$
	$T_D = \theta\frac{6 + 2R}{22 + 3R}$