PIM-based Digital Controller for System with Process Delay

by

Muhammad Zuhair bin Norhisam

15550

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical & Electronic)

SEPTEMBER 2014

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

PIM-based Digital Controller for System with Process Delay

by

Muhammad Zuhair bin Norhisam

123456

A project dissertation submitted to the
Electrical & Electronic Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONIC)

Approved by,

____

(Suhaila Badarol Hisham)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
September 2014

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 references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MUHAMMAD ZUHAIR BIN NORHISAM

Abstract

Discretization of a controller in a process plant is a norm through a variety of means. A typical approach of industry involves converting an analog time controller to discrete time controller using PID adjustment method. Delayed system however, suffers degradation in its transient performance at certain level and form including but not limited to; higher overshoot and slower settling time. As such, process' tuning is always mandatory as a means of compensation. Plant Input Mapping (PIM) based method aims to reduce the performance's degradation with minimal tuning involved. In this project, an epsilon operator is used and preferred compared to z operator. Both the plant and the controller will be discretized using Step Invariant Model (SIM) and Matched Pole Zero (MPZ) technique respectively. Experiment has been done to compare the performance of discrete PID and PIM method. First order dead time (FODT) transfer function of the plant is calculated using statistical modelling method and thus, the values for continuous time (CT) PID is calculated using the open loop tuning method. Simulation of all process are generated through Simulink's model. It is observed that the performance of PIM could rival the DT PID's however, there is some limitation in PIM technique that makes the technique undesirable. Further modification of technique is done and proven to be significantly better than DT PID and closely simulates its continuous system counterpart.

Acknowledgement

Even a full page of words could not describe how much thankful and grateful I am to my supervisor who guides me throughout the duration.

Sincerely, thank you.

Table of Contents

1: Introduction

1.1: Background

Process plant are continuous-time (CT) in nature. For a typical closed-loop system, controllers, in which they may vary from a simple Proportional, Integral and Derivative (PID) or state space controller, are often used to stabilize and optimize the output of the system. They act by calculating the error of the process versus the reference point and manipulating the controlled variable to produce the desired outcome. These controllers are designed based on continuous plant normally before they are being discretized and turned into a discrete time controller (DT) to be implemented [1]. This method is considerably cost effective than having a whole discrete plant from the start using a Z-Transform technique which is termed "direct design". This method is referred as "emulation of continuous controller" and it involves the classical approach in determining CT controller such as root locus method as the first step and PID adjustment method is used later on to discretize the CT controller. In summary;

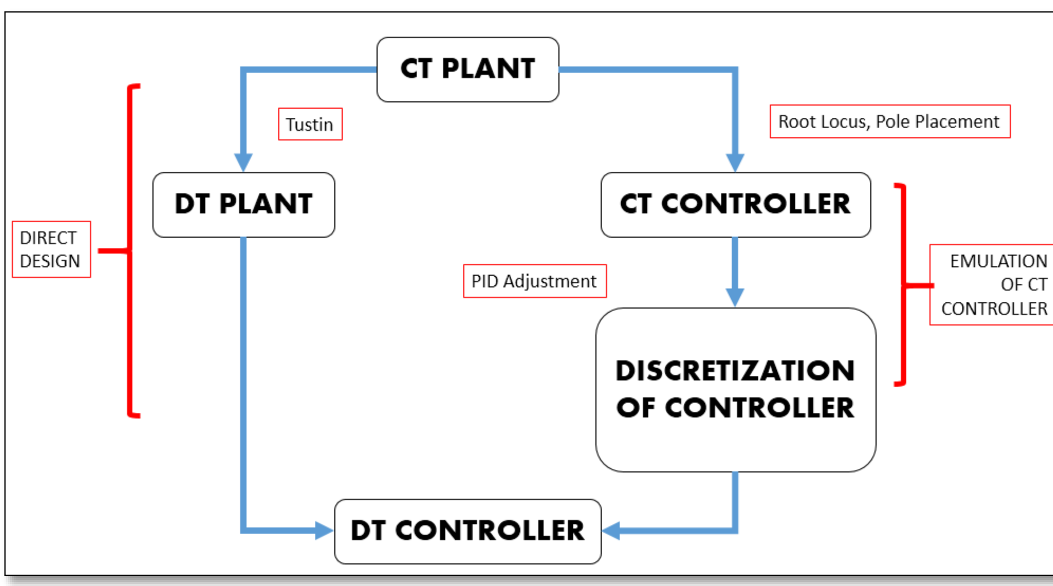


Figure 1 : Typical concepts for discretizing controller

Generally, dead time contributes a detrimental effect as a continuous process is being discretized. While dead time is a superset of process delay, computational delay and network delay, engineers often treat calculation delay as insignificant threat compared to process delay. Process delay is the effect of reaction process. Certain chemical process has slow reaction, moreover in those that deal with temperature and pressure. It is in fact the major contributor of dead time. Calculation delay usually caused while processing huge algorithm during the process. It may also a result from an improper sampling technique. While recent digitalizing techniques would incorporate some algorithm to counter this issue, they are, however, vary in term of outcome's performance. This project would focus on the performance of Plant Input Mapping (PIM) based technique to normalize the computational delay compared to other techniques such as normal adjustment of half the sampling time to PID.

PID needs to be adjusted in order to enhance its transient discrete performance. Typically, half of the sampling time is added to delay term before the controller is being discretized. This may boost the performance however a further fine tuning technique is needed to obtain a better performance.

PIM approach integrates delta transform in its calculation unlike its counterpart whereby a Laplace or Z transform would be used in usual practice. The intended transfer function is termed as Plant Input Transfer Function (PITF) with epsilon (ε) operator as its basis. The equation is a complex polynomial form with multiple unknowns, often solved by the aid of Diophantine equation, the reverse of Sylvester Matrix Equation. The PIM approach is a partial discretization of whole plant. Only the plant process is left out in the discretization technique and the form would differ for *n*-number of blocks. This approach would take Match-Pole-Zero (MPZ) method for the discretization of PITF and Step Invariant Model (SIM) model for the plant part into account for calculation.

1.2: Problem Statement

The discretization process of continuous controller degrades the transient performance. More oscillation, longer settling time, and higher overshoot are part of the addressed issues observed in the discrete implementation of the controller. This could be fixed by few iterations of fine tuning although the result could not be assured. Since this method is being used commonly in industry, many would share the same issue while discretizing the continuous process.

1.3: Objectives and Scope of Study

Throughout the duration, both minor and major project's requisite that lead to the final conclusion has been identified as the objectives as listed below;

- To design PIM controller based on CT Plant.
- To simulate PIM controller using Matlab and Simulink.
- To compare the performance of PIM controller with DT PID and ultimately compare to CT PID.

Although both direct discretization and emulation method has been discussed, this project will only consider emulation for the research's purposes.

2: Literature Review and/or Theory

The Account of Half Sampling Time

Half of sampling interval is referred to many as standard practice in discretizing PID. H.H Ray in his book cites it is a custom in MRI process to add the half of sampling time to normalize the graph against system lag [4]. It is again cited in Gopal's book as standard procedure to maintain the consistency of discrete graph again continuous stream [5].

Delta Transform

Delta transform is a method in which a discrete time-domain function is converted and defined in ε operator, much as Z or Laplace transform. Theoretically, it is expressed as;

$$F(\varepsilon, T)= \Delta\, [f(k,T)]=\Sigma\, f(k,T)(T\varepsilon+1)^{-k}T$$

For reference, the table of typical conversion from time domain to epsilon domain is provided in section **7.2.1 Appendices**.

The delta transform is unique in that it poles and zero does not bounded in a conventional left side of the graph but it has a radius circle, r and the value varies according to its sampling interval parameter as per below;

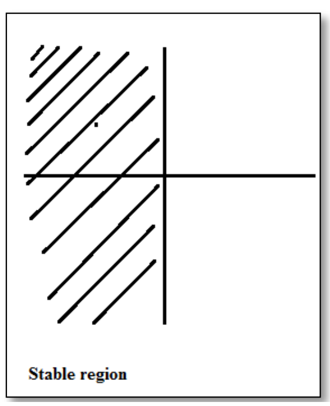


Figure 2 : Typical region of stability

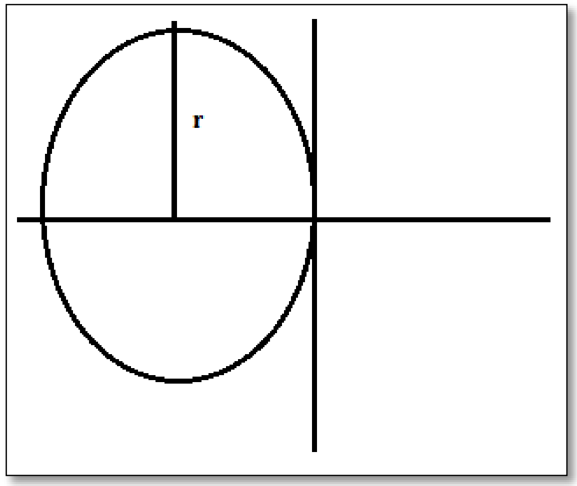


Figure 3: Delta transform's stability region

By manipulating the parameter that correlate with the radius of the region, it is possible to extend the region unlimited across the graph. This ensure the stability of the function to a greater length and at the same time it has the advantage over z transform in term of numerical properties [2].

While delta transform is unique, it can be applied effortlessly since it can be associated with z-operator by below relationship where T is the sampling interval;

$$\varepsilon = \frac{z-1}{T}$$

Plant Input Transfer Function

Using ε operator, a continuous loop can be expressed in its discrete form called plant input transfer function. The discretization process only occurs in the control and feedback block while leaving the process plant block in its original state.

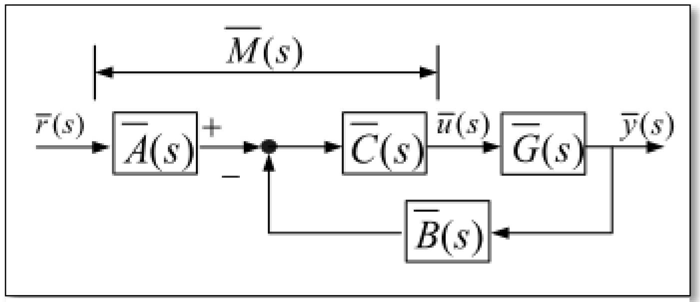


Figure 4: PITF concept

In this example, G(ε) block would remain analog while the rest block will be redesigned as discrete block. Since the aim of PITF is to achieve the partial discretization, the transfer function would be U(ε)/r(ε) instead of having the usual output, y(ε) as the nominator. Although G(ε) is not conceptually discretized, the block is still a part of the error equation and the whole process would still use the transfer function from the block to achieve PITF.

If the blocks are expressed in polynomial form;

$$\overline{A}(s)=\frac{\overline{n}_A(s)}{\overline{d}_A(s)},\quad \overline{B}(s)=\frac{\overline{n}_B(s)}{\overline{d}_B(s)},$$

$$\overline{C}(s)=\frac{\overline{n}_C(s)}{\overline{d}_C(s)},\quad \overline{G}(s)=\frac{\overline{n}_G(s)}{\overline{d}_G(s)}.$$

Figure 5: Blocks in polynomial form

then, PITF can be expressed as;

$$M(\varepsilon)=\frac{n_A(\varepsilon)d_G(\varepsilon)}{n_B(\varepsilon)n_G(\varepsilon)+d_C(\varepsilon)d_G(\varepsilon)}$$

Figure 6: Actual PITF equation

The calculation of PITF is approached by the combination of MPZ method and SIM method while the multiple unknowns are solved by using Diophantine equation [3].

Matched Pole-Zero and Step Invariant Model

MPZ and SIM is a method to convert a transfer function block into a ε domain. They use delta operation as the underlying concept. SIM adapts a sampling interval into equation by the factor, $T\varepsilon + 1$. In almost cases, this will yield extra order into the equation and later can be simplified if possible. Unlike SIM, MPZ does not account for the sampling interval and is used for calculation of the feedback and control block of the PITF.

PID Tuning

PID tuning is a process to obtain the optimal values for P, I and D. It is done in step where by the gain P is discovered first and subsequently, using the same value of P to optimize the value of I and D [8]. The system's response towards each gain is calculated and observed to select the prime combination that has the fast rise time and settling time. This is however would be balanced out by the overshoot of the response, creating a coherent oscillation that should diminish by a certain ratio, often calculated as 25%.

Ziegler, in his research, has developed method to tune PID either in its closed loop form or open loop state. For the project's purpose, open loop tuning would be selected as the better approach. His method uses the plant model parameters to induce the value of P, I and D [8].

Simply shown;

$$G_m(s) = \frac{k_m e^{-ds}}{\tau_m s + 1}$$

Controller	k_c	τ_I	τ_D
P	$\frac{1}{K_m} \cdot \frac{\tau_m}{d}$	-	-
PI	$\frac{0.9}{Km} \cdot \frac{\tau_m}{d}$	$\frac{d}{0.3}$	-
PID	$\frac{1.2}{Km} \cdot \frac{\tau_{pn}}{d}$	2d	0.5d

Figure 7: Ziegler Nichols Open Loop Method

Another tuning method that is used is Cohen Coon open loop tuning. This method differs from Ziegler Nichols method with its more complex formula. However, it is noted that the performance generally have slow rise time to avoid high overshoot. The formula for this method is shown below;

Cohen-Coon Tuning Rules

	K_c	τ_{Int}	τ_{Der}
P	$\frac{1}{rK}\left(1+\frac{r}{3}\right)$		
PI	$\frac{1}{rK}\left(0.9+\frac{r}{12}\right)$	$\tau_{del}\frac{30+3r}{9+20r}$	
PID	$\frac{1}{rK}\left(\frac{4}{3}+\frac{r}{4}\right)$	$\tau_{del}\frac{32+6r}{13+8r}$	$\tau_{del}\frac{4}{11+2r}$

Figure 8: Cohen Coon Open Loop Method

3: Methodology/Project Work

The project has a flow chart as below;

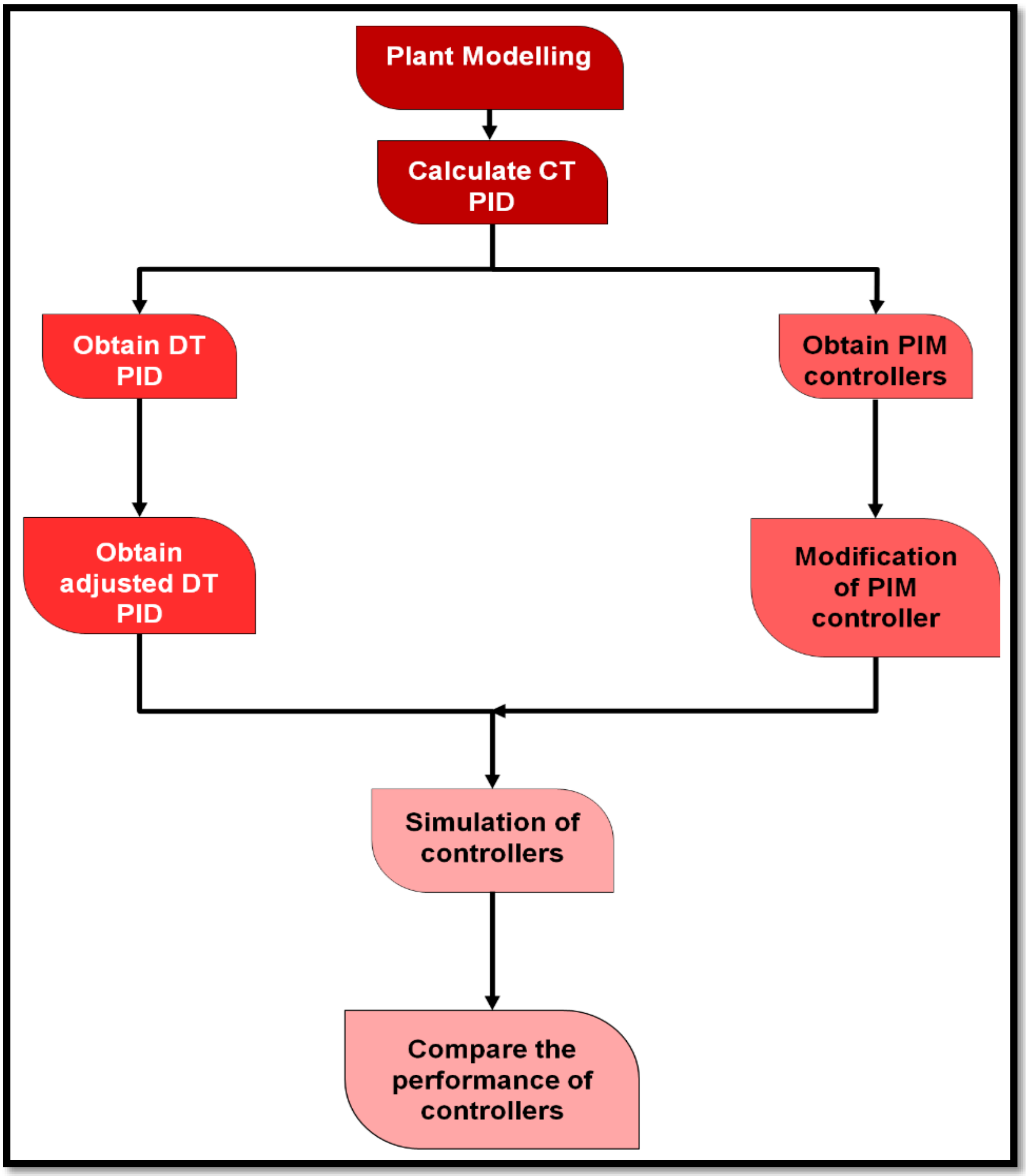


Figure 9: Project's flow chart

3.1: Plant Modelling

Liquid Plant – Simple Heat Exchanger Temperature Control (TIC312) of Universiti Teknologi Petronas' (UTP) block 23 lab is used as the basis of the project. A layout of the plan is provided in **7.3.1 Appendices** section.

This plant simulates the heat exchanging process where there would be two fluids entering the shell and tube of the heat exchanger to reach equivalent temperature. In this case, fluid from tank VE330 is used to heat up the cool fluid from vessel VE300. The heated up fluid from VE300 would then be cooled down for recycling purposes.

Temperature controller, TIC342 is used to control the temperature of fluid from VE330 by comparing it against the set point and controlling the boiling process. The flow controller, FIC312 is used to control the opening of the valve towards tube side of the heat exchanger.

These loops' output would then be plotted against time to display plant's performance. A plant modelling using statistical method is done in order to get the plant's parameters of first order with dead time (FODT) model as below where Kp is the gain of the system, θ denotes the dead time of the system and τ refers to the rise time.

$$\frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1} \qquad \dots\dots\dots\dots\dots (eqn1)$$

In statistical approach, the data for both manipulated variable and process value for the whole sampling period are taken into account. The output data would then be adjusted accordingly with a set of Gamma, σ which represents multiple of time delay. Then they are plotted with a linear regression model. An FODT equation is obtained from the plot using correlations as below;

$$(Y'_{i+1}) = a(Y'_{i}) + b(X'_{1-\sigma}) \qquad \dots\dots\dots\dots\dots (eqn2)$$

$$a = e^{-\Delta T/\tau} \quad \dots\dots\dots\dots\dots(eqn3)$$

$$b = K_p(1 - e^{-\frac{\Delta T}{\tau}}) \quad \dots\dots\dots\dots\dots(eqn4)$$

3.2: CT PID Controller Parameters

CT PID is calculated manually using Ziegler Nichols and Cohen Coon correlations. The set of all modes of controllers are then compared between both technique to determine which correlation and mode provides the best initial values for the system. This calculation is based on the open loop tuning of the system.

The formula for both methods are summarized as below;

Table 1: Ziegler Nichols Open Loop Tuning

P	$\frac{1}{K_p}(\frac{\tau}{\theta})$	-	-
PI	$\frac{0.9}{K_p}(\frac{\tau}{\theta})$	$\frac{\theta}{0.3}$	-
PID	$\frac{1.2}{K_p}(\frac{\tau}{\theta})$	$2\,\theta$	$0.5\,\theta$

Table 2: Cohen Coon Open Loop Tuning

P	$\frac{1}{rK_p}(1+\frac{r}{3})$	-	-
PI	$\frac{1}{rK_p}(0.9+\frac{r}{12})$	$\tau\frac{30+3r}{9+20r}$	-
PID	$\frac{1.2}{K_p}(\frac{4}{3}+\frac{r}{4})$	$\tau\frac{32+6r}{13+8r}$	$\tau\frac{4}{11+2r}$
Where r = $\frac{\theta}{\tau}$			

3.3: Discrete PID and Adjustment

Discretization of PID is simulated using Simulink's Discrete PID block. Corresponding sampling time as used in DCS is applied during the simulation to reflect the actual working system. This discrete controller are then paired with the CT plant and a closed loop system is obtained. Next, half of the sampling time, T is added into time delay generating an adjusted time delay.

$$\theta' = \theta + \frac{T}{2} \qquad \dots\dots\dots\dots\dots(eqn5)$$

Using this value, a new set of discrete PID is simulated and compared with original DT PID.

3.4: PIM-Based Controller

PIM controller is obtained by solving the Diophantine equation. This Diophantine equation is restated in a matrix form called Sylvester Matrix. It is the result of the discrete plant using SIM and the discrete PITF equation using MPZ as described in section **2.1:**

Theory. The PIM based controller parameters are calculated using MATLAB by the aid of Delta Toolbox. Polynomials are often handled in their state space form as opposed to classical transfer function to avoid numerical errors.

3.5: Modification of PIM controller

Modification of PIM controller is achieved by modifying the input to the Diophantine matrix while maintaining its theoretical calculation. This can be done by substituting the delay term in plant's FODT equation by using Pade approximation. Since both input to the Diophantine equation have a delay term incorporated in the equations, a modification have been done in two ways;

1) Delay terms in both plant transfer function and PITF equation are substituted using Pade approximant.

   This method is termed as **PIM with Full Pade Approximation (PIMfp)** since every delay terms are taken into account.

2) Only the delay term in PITF equation is approximated while maintaining the original term in plant transfer function.

   This method is termed as **PIM with Partial Pade Approximation (PIMpp)** since delay terms in PITF is taken into account.

These two methods could be attempted with Pade approximation of different degree of order as discussed in section **2.1: Theory.** The table below summarized the method of PIM controller's modification;

Table 3: Modification of PIM

	Input to SIM	**Input to PITF**
PIM	Plant transfer function, G(s).	PITF equation, M(s).
PIMfp	Plant transfer function with Pade approximation, Gp (s).	PITF equation with Pade approximation, Mp (s).
PIMpp	Plant transfer function, G(s).	PITF equation with Pade approximant, Mp (s).

3.6: Simulation and Comparison

Both type of controllers are simulated using Simulink. Both discrete controllers are compared to each other as well as to their analog counterpart. The best controller will be judged based on the settling time, rise time and overshoot percentage for comparison purposes. Table below describes the desired outcome;

Table 4: Selection Criteria

Criteria	Desired Outcome
Overshoot Percentage (OS%)	< 25%
Settling Time (Ts)	<60
Rise Time (Tr)	<60

4: Result and Discussion

As discussed in **3: Methodology** section, experiments were done using Liquid Plant – Simple Heat Exchanger Temperature Control and the data were collected from two controller, FIC631 which monitors flow process and TIC634, which in charges of temperature processes.

4.1: The Modelling of the Continuous Plant

Standard procedure of obtaining process reaction curve (PRC) using statistical method is used. In this method, a set of data of output (process variable, PV) and its input (manipulated variable, MV) is obtained using open loop increment of MV. Coefficient of "a" and "b" are determined by using equation 3 and 4 respectively. Linear regression is used to model the plant transfer function;

$$\theta = (\Gamma \text{ x sampling time}) \quad \ldots\ldots\ldots\ldots\ldots (eqn6)$$

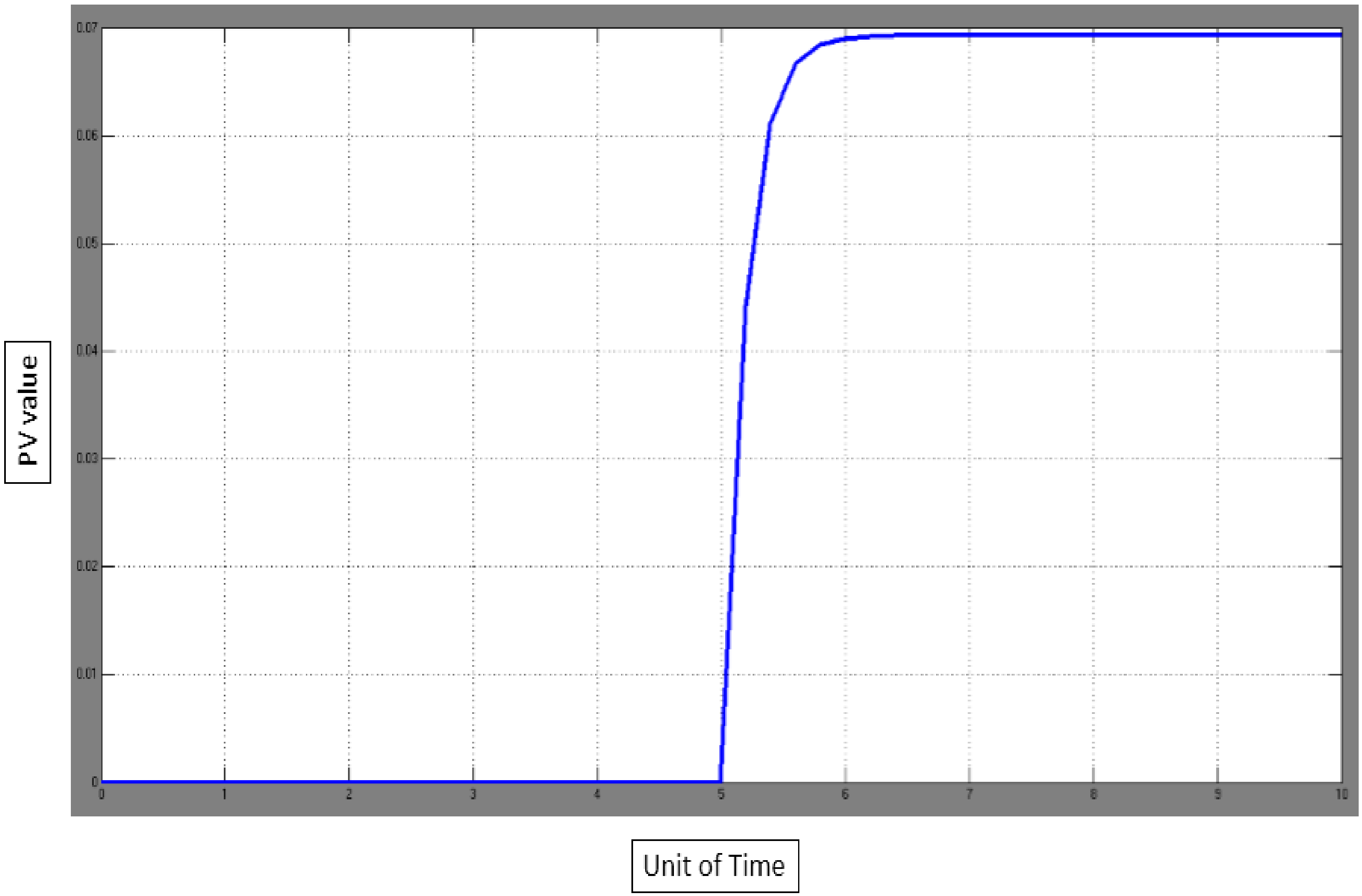
$$\tau = (\text{- sampling time})/\log(a) \quad \ldots\ldots\ldots\ldots\ldots (eqn7)$$

$$Kp = b/(1\text{-}e^{-(\text{sampling time})/\ \tau} \quad \ldots\ldots\ldots\ldots\ldots (eqn8)$$

Based on this method, the transfer function of the process is estimated as;

$$G(s) = \frac{K_P e^{-\theta s}}{\tau s+1} = \frac{0.069e^{-4s}}{0.182s+1}$$

Graph of CT Plant



In the simulation, a set point of 1 is specified. This set point represents the flow of liquid in m^3/s if perceived through the actual system. In the simulation, an open loop plant without a controller gave a reading of 0.07 after 5 unit of time as opposed to the desired value, 1 before the system stabilized. The needs of controller is obvious to boost the performance of the system.

4.2: Calculation of CT PID

CT PID values are calculated using Cohen Coon open loop tuning method and Ziegler Nichols open loop tuning method. The formula for both are stated in section **3.2: Methodology**. Both methods' performance are compared and the best mode will be selected as the process' controller mode.

The PID values for flow process are tabled as below;

Table 5: Cohen Coon's PID parameters

Cohen Coon Tuning Parameters	**P-only**	**PI**	**PID**
Proportional gain, K_C	5.4665	1.7932	4.4827
Integral time, T_I (minutes/repeat)	-	0.8554	3.4711
Derivative time, T_D (minutes/repeat)	-	-	0.2910

Table 6: Ziegler Nichol's PID parameters

Ziegler Nichols Tuning Parameters	**P-only**	**PI**	**PID**
Proportional gain, K_C	0.6563	0.5907	0.7875
Integral time, T_I (minutes/repeat)	-	13.3333	8
Derivative time, T_D (minutes/repeat)	-	-	2

Using the modes values from the CT PID calculations, simulation of the system performance has been conducted using MATLab. Based on the graph from the simulation, the observation on the performance of each controller mode has been done to analyze the affect and has been discussed which mode is the best for the system.

The Simulink model for the closed loop system is shown below;

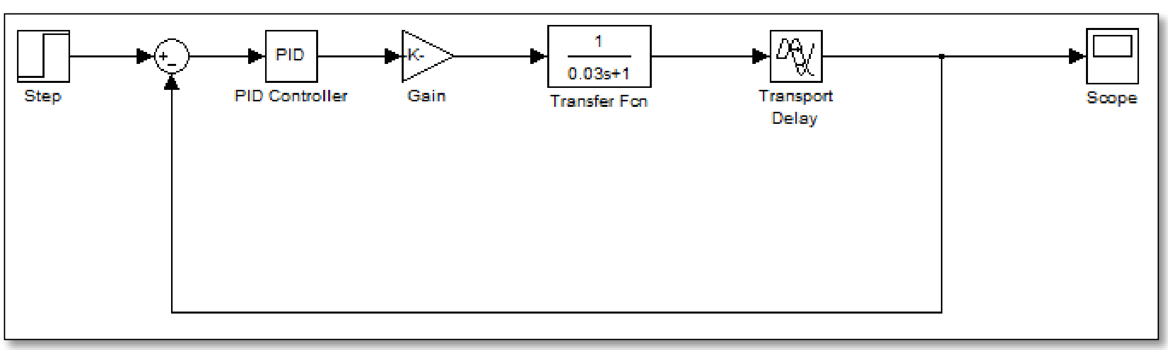


Figure 10: Closed Loop Model

The performance between the modes of controller (P, PI, and PID) for the system have been compared and analyzed. The controller should achieve zero offset whenever integral mode is used where zero offset means that the final steady state value of the control variable is equal to set point or at least reach ±5% of SP. In this simulation, only PI and PID controller could achieve zero offset while P-only controller has the largest offset due to temperature behavior.

Both methods are simulated and PID derived from Cohen Coon’s method consistently perform better compared to PID tuning derived from Ziegler-Nichols’ method for flow process.

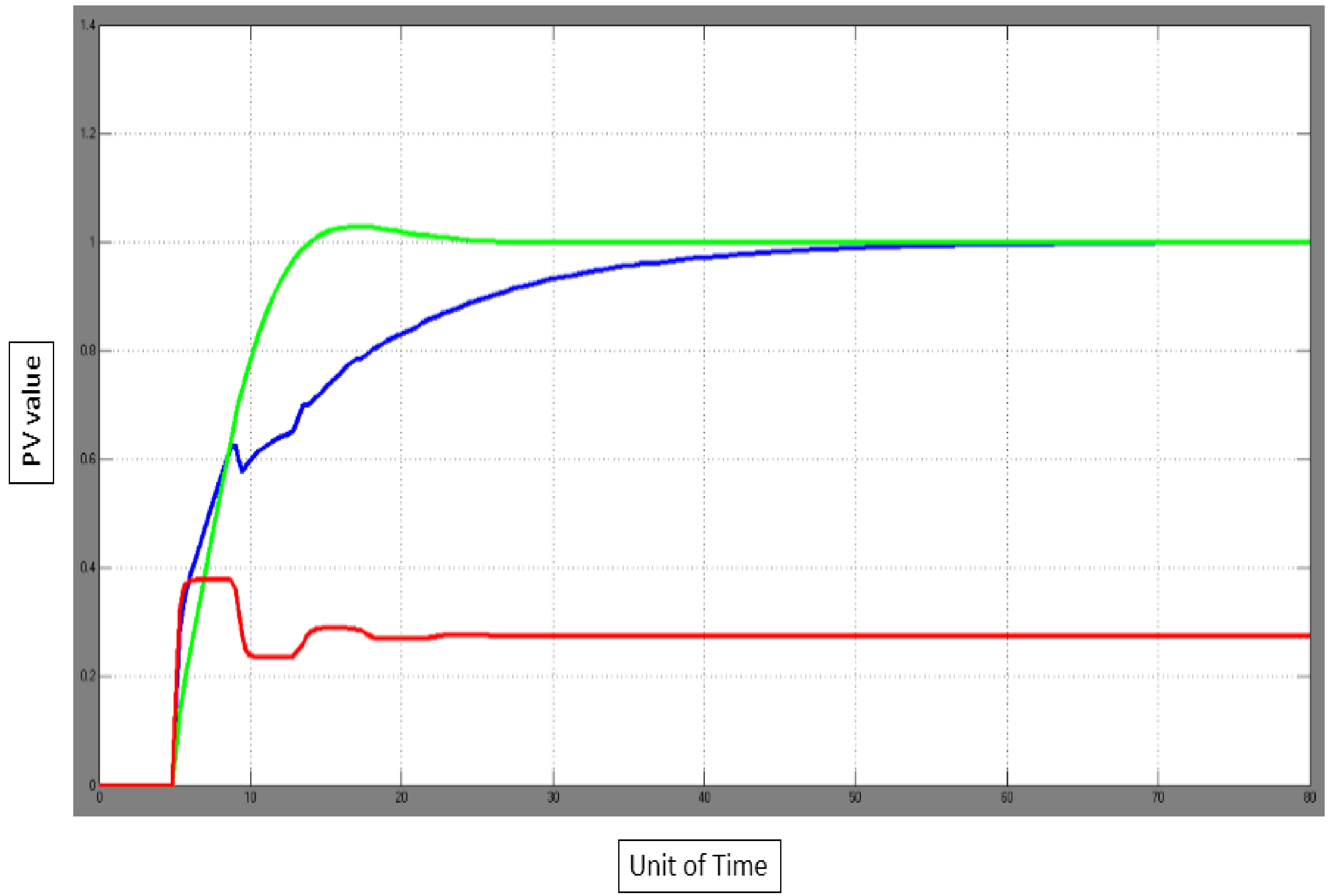
For flow process, Cohen Coon’s PID mode gives greater overshoot overall compared to Cohen Coon’s PI mode. It is however noted that the settling time of Cohen Coon’s PID is the slowest, at 80 time unit compared to PI and it compensates well the aggressive overshoot that happened early. Cohen Coon’s PI mode reaches steady state at 30 time unit. On the other hand, Cohen Coon’s P mode have a huge offset which set it far from qualified. Ziegler Nichols method gives unstable reaction for the process, and thus will not be considered. Thus, Cohen PI is concluded as the best controller for this system based on the observation and analysis that have been done.

Simply tabled and graphed;

Table 7: Summary for Controller’s Performance

	Cohen’s P **(Red line)**	Cohen’s PI **(Green line)**	Cohen’s PID **(Blue line)**
Offset	Large	-	-
Overshoot	Small	Small	-
Settling Time	-	30 time unit	80 time unit

Graph of Cohen Coon's Controllers Performance



4.3: Simulation of DT PID

Simulink's Discrete PID controller that comes from SimPower System is used to discretize CT PID. This block is done by Pierre Giroux and Gilbert Sybille of Hydro Quebc.

This block uses below correlation to emulate the continuous process as a discrete signal;

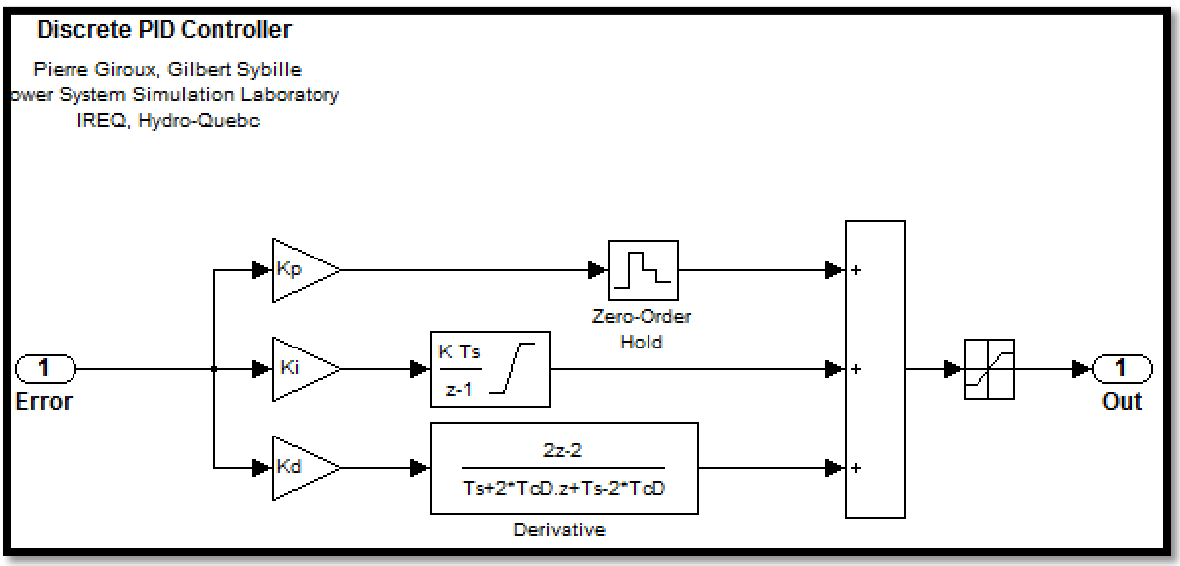
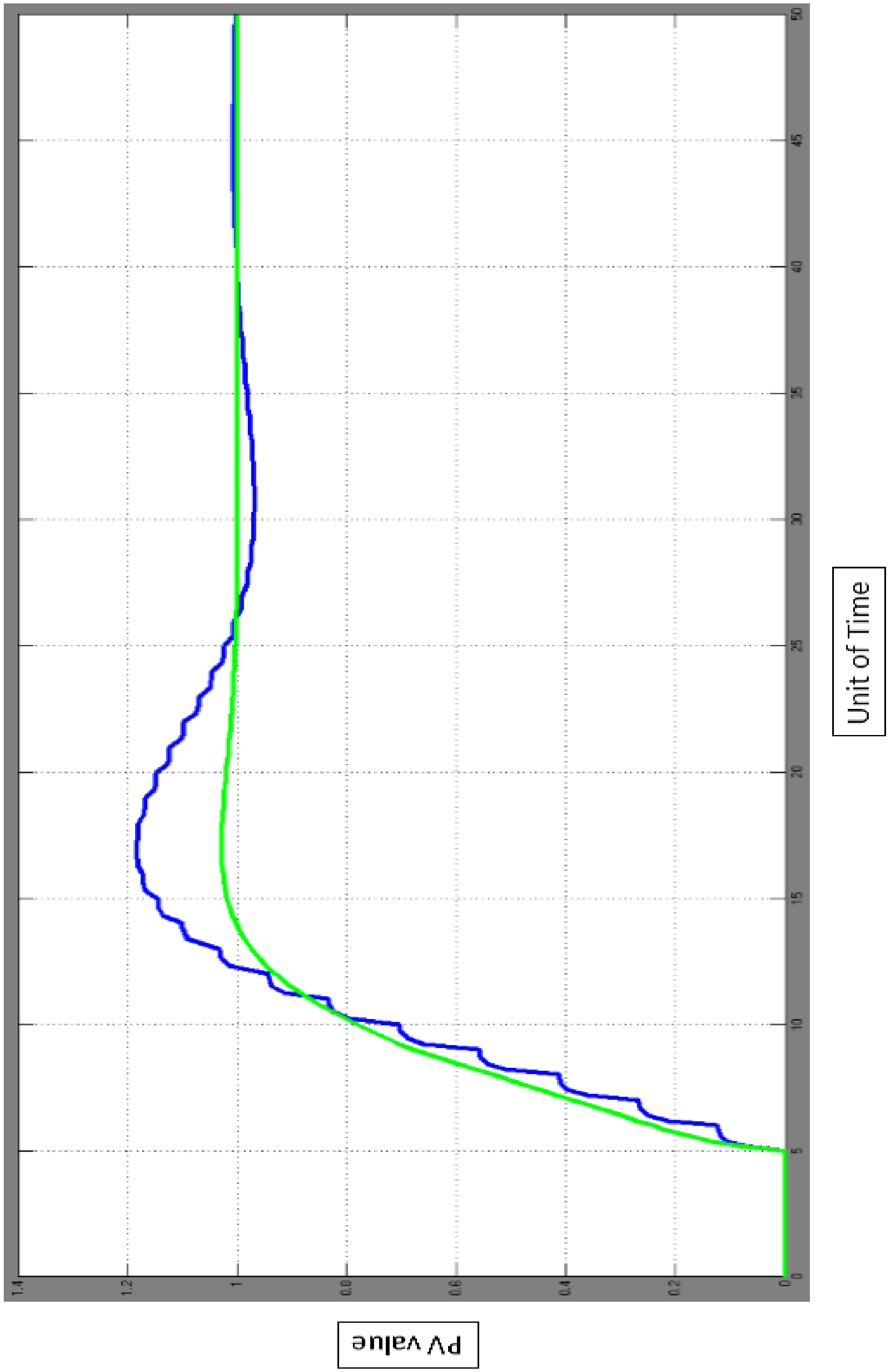


Figure 11: Discrete PID Controller

Graph of CT PID vs DT PID



4.4: DT PID Adjustment

Direct discretization of CT PID controller yield the degradation of transient performance as discussed in previous sections. Half of sampling time is added into the delay term as a mean to counter the issue and is termed DT PIDadj. This is proven when the adjusted DT PID is plotted against direct DT PID in the simulation.

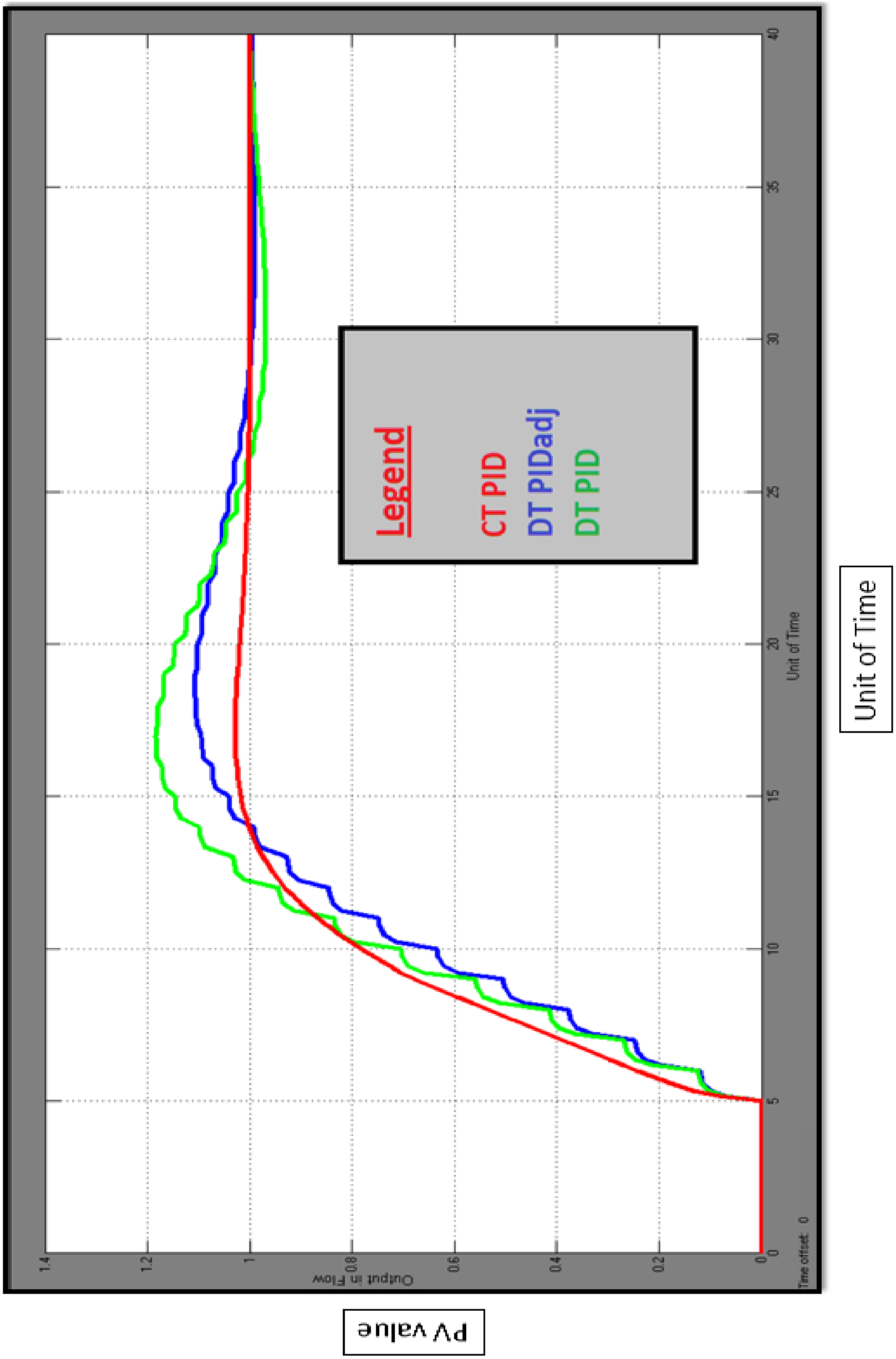
DT PID takes around 60 unit of time to settle while DT PIDadj reaches 1 at 45 unit of time. Although DT PIDadj settles faster, it does well for the overshoot percentage. It has considerably lower overshoot percentage compared to the DT PID. DT PIDadj has faster rise time as well.

Simply tabled;

Table 8: Performance of DT PID against DT PIDadj

	OS%	Ts	Tr
DT PID	High	60	Slow
DT PIDadj	Low	45	Fast

Graph of DT PID vs DT PIDadj



4.5: PIM's Design

As discussed previously in **2: Literature Review/Theory** section, discretization of plant can be achieved through SIM method which involves delta operation, while other controllers are discretized using Matched Pole Zero (MPZ). Using Matlab and Simulink, a model of discrete controller can be achieved as shown in figure below. The input of the system is converted to frequency domain from time domain and they are in state space form. Based on the relationship between z operator and ε operator, a conversion between the two can be used to produce an equivalent inverse Delta Block shown in figure 14. This block works in a similar fashion to unit delay block in Simulink.

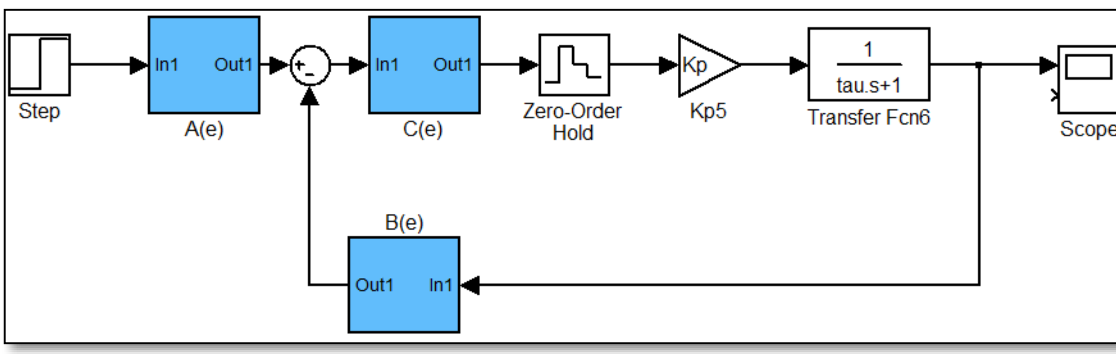


Figure 12: Simulation of PIM 3 Block Controller

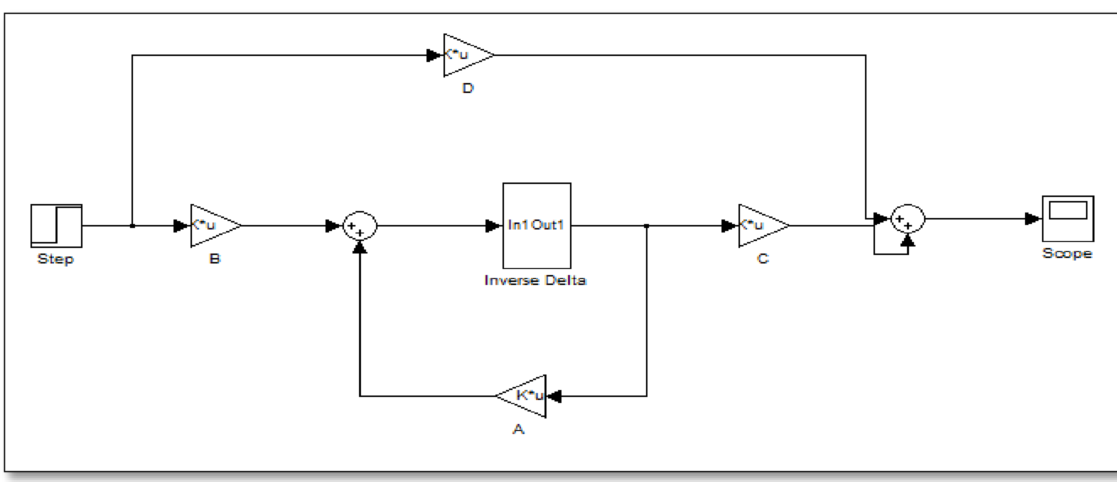


Figure 13: Controller's Configuration

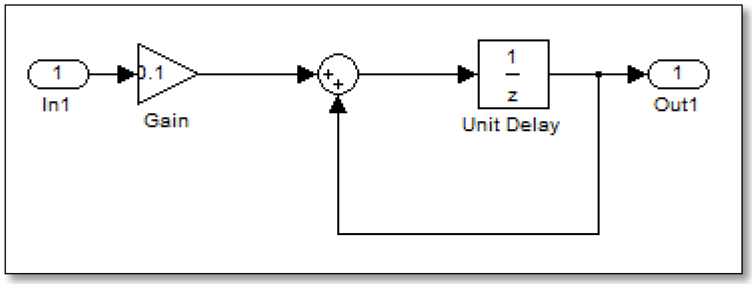


Figure 14: Epsilon Operator Block

4.6: Calculation of PIM's Parameters and its Variance

PIM controller's parameters are calculated with the aid of Delta Toolbox. The plant transfer function, G(s) is be discretized using SIM while PITF equation, M(s) is discretized using MPZ.

4.6.1: PIM Controllers Parameters

In PIM technique, the same plant transfer function and PITF equation are used. By using SIM, G(ε) is obtained as below;

$$\frac{0.069}{(\varepsilon+0.9959)(\varepsilon+1)^4} \qquad \ldots\ldots\ldots\ldots\ldots(eqn9)$$

PITF is recognized as

$$\frac{1.793s^2+11.95s+11.52}{s^2+6.18s+0.7986} \qquad \ldots\ldots\ldots\ldots\ldots(eqn10)$$

And its discrete form, M(ε) is obtained using MPZ;

$$\frac{2.5939(\varepsilon+0.9959)(\varepsilon+1)^4(\varepsilon+0.6894)((\varepsilon+0.5)^3)}{(\varepsilon+0.9976)(\varepsilon+1)^4(\varepsilon+0.5)^3(\varepsilon+0.1237)} \qquad \ldots\ldots\ldots\ldots\ldots(eqn11)$$

4.6.2: PIMfp Controllers Parameters

In PIMfp technique, both plant transfer function's and PITF equation's delay term are substituted with Pade approximation term. Second order Pade approximation is used as it is observed that the second order equation reflects the actual delay the closest as opposed to the other orders when in use.

Second order Pade approximation is termed as below;

$$\frac{1.333s^2 - 2s + 1}{1.333s^2 + 2s + 1} \qquad \dots \dots \dots \dots \dots (eqn12)$$

Hence, the new plant transfer function, Gp(s) is determined as below;

$$\frac{0.0924s^2 - 0.1386s + 0.0693}{0.2426s^3 + 1.697s^2 + 2.182s + 1} \qquad \dots \dots \dots \dots \dots (eqn13)$$

And the Gp(ε) is;

$$\frac{-0.018915(\varepsilon - 0.7451)(\varepsilon + 1.79)}{(\varepsilon + 0.9959)(\varepsilon^2 + 1.142\varepsilon + 0.3656)} \qquad \dots \dots \dots \dots \dots (eqn14)$$

PITF equation with Pade approximation, Mp(ε) is as below;

$$\frac{2.8955(\varepsilon + 0.6894)(\varepsilon + 0.5)(\varepsilon + 0.6894)(\varepsilon + 0.9959)(\varepsilon^2 + 1.142\varepsilon + 0.3656)}{(\varepsilon + 0.5)(\varepsilon + 0.5362)(\varepsilon + 0.9983)(\varepsilon^2 + 0.5009\varepsilon + 0.09408)} \qquad \dots (eqn15)$$

4.6.3: PIMpp Controllers Parameters

In PIMpp technique, only PITF equation's delay term are substituted with Pade approximation term. First order Pade approximation is used as it is observed that the higher order equation gives higher overshoot as well as slow response.

First order Pade approximation is termed as below;

$$\frac{-2s+1}{2s+1} \qquad \ldots\ldots\ldots\ldots\ldots (eqn16)$$

Mp(ε) with Pade first order is as below;

$$\frac{3.47612.5939(\varepsilon+0.9959)(\varepsilon+1)^6(\varepsilon+0.6894)(\varepsilon+0.3935)}{(\varepsilon+0.9922)(\varepsilon+1)(\varepsilon+0.9983)(\varepsilon^2+0.4309\varepsilon+0.06558)(\varepsilon+1)^6} \qquad \ldots (eqn17)$$

4.7: Comparison of Controllers' Performance

Each variation of PIM method are compared with DT PIDajd to decide which controller has the best performance according the predetermined criteria. All of controllers are simulated to observe their transient performance.

PIM controller does not perform as anticipated when compared to DT PIDadj. Although it has 0% overshoot, it has a slow rise time and settling time. DT PIDadj compensates a high percentage overshoot, at 10.65% to give a boost in rise time as well as settling time. It can be said that PIM controller performs on par with DT PIDadj. The tradeoff between overshoot percentage and rise time gives the DT PIDadj a more favorable spot when a flow process is concern as anything less than 25% overshoot is considered acceptable.

PIMfp and PIMpp techniques on the other hand takes the advantage of numerical calculation involved through the use of Pade approximation. The difference between the modified techniques with the base approach lies in the formation of PITF equation. In PIM technique, a delay term is multiplied to the whole denominator equation as shown below;

$$\frac{Na(\varepsilon)\mathrm{Dg}(\varepsilon)}{[Nb(\varepsilon)\mathrm{Ng}(\varepsilon)+Dc(\varepsilon)\mathrm{Dg}(\varepsilon)](\mathrm{Delay\ term})} \qquad \ldots\ldots\ldots\ldots\ldots(eqn18)$$

The delay term is introduced in the denominator while only nominator of plant, Ng(ε) should be multiplied to the term. This may results in the passive behavior of the base technique as compared to the PIMfp and PIMpp approach. In the latter methods, an equivalent Pade equation is introduced to Ng(ε).

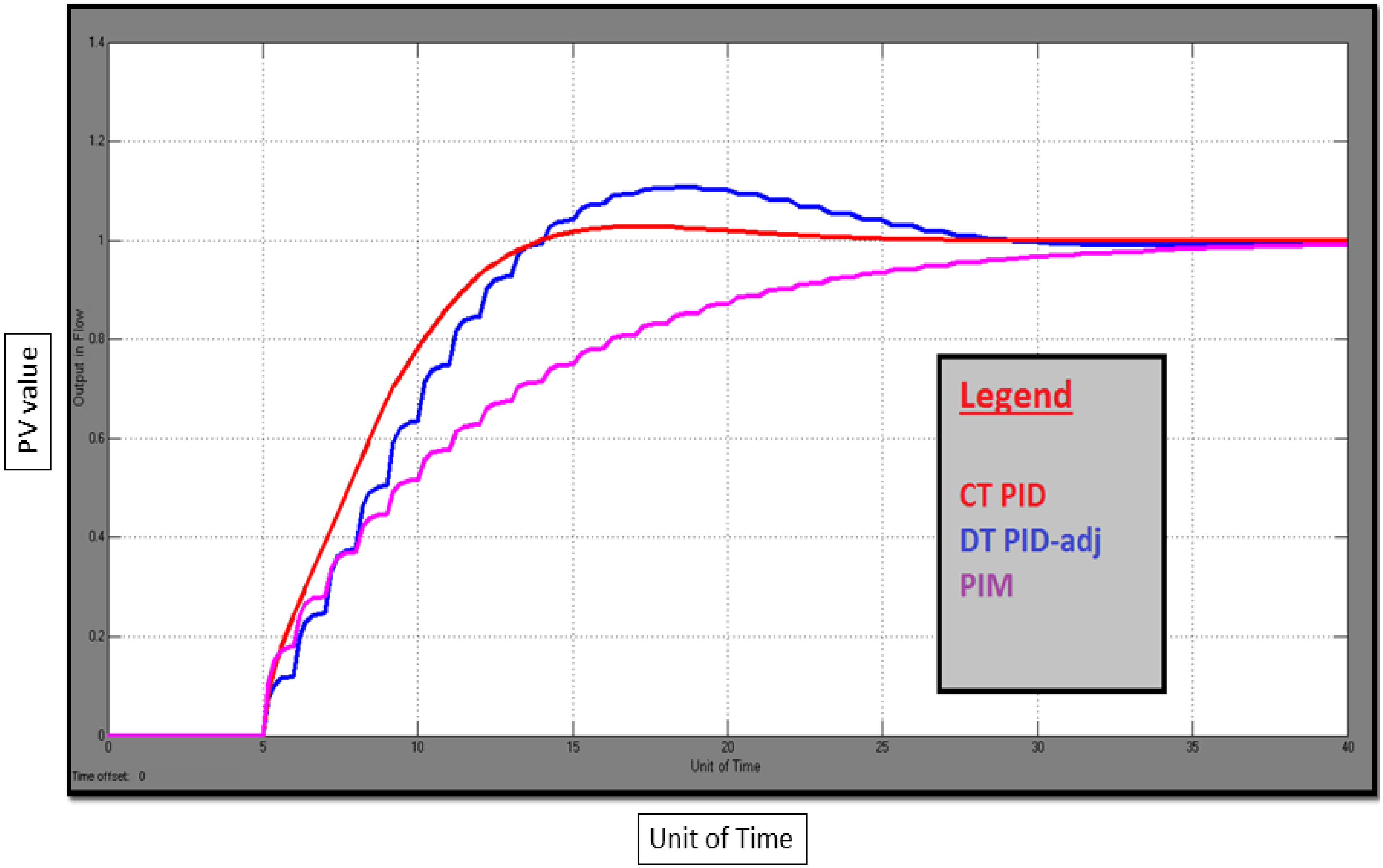
Both methods excel in all criteria and are considered superior to DT PIDadj. PIMfp has the fastest rise time among all controllers, at 38 unit of time. Surprisingly, it has the lowest percentage overshoot even compared to the CT PID. PIMfp also has a considerably fast rise time compared to the other controllers.

Table below conclude the comparison among controllers;

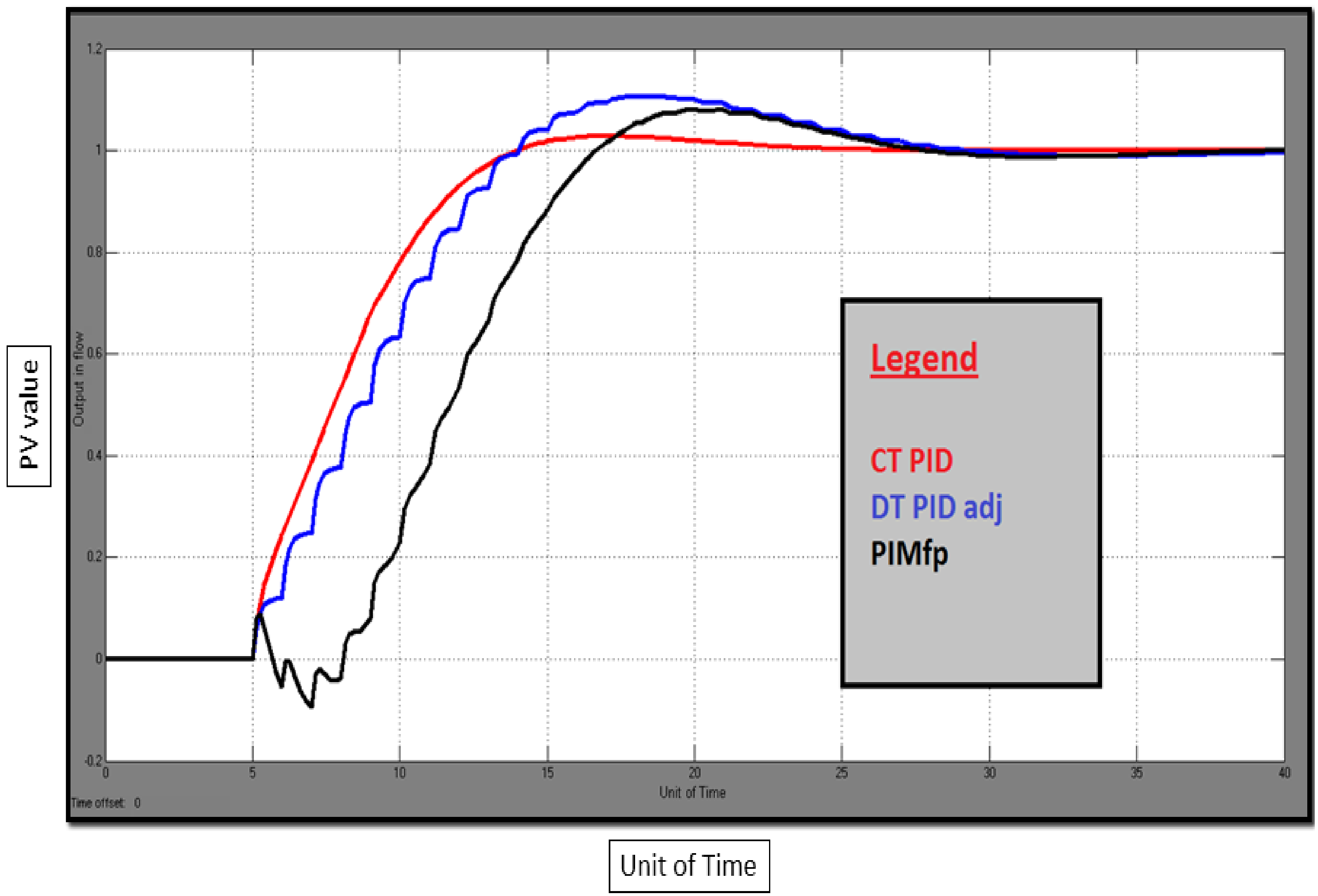
Table 9: Comparison of Controllers' Performance

Method	Input to SIM	Input to PITF	Ts	OS %	Tr
CT PID	-	-	28	2.81%	26.75
DT PIDadj	-	-	45	10.65%	36.24
PIM	G(s)	G(s) & C(s)	50	0%	61.71
PIMfp	Gp(s)	Gp(s) & C(s)	45	8.28%	26.53
PIMpp	G(s)	Gp(s) & C(s)	38	1.99%	32.07

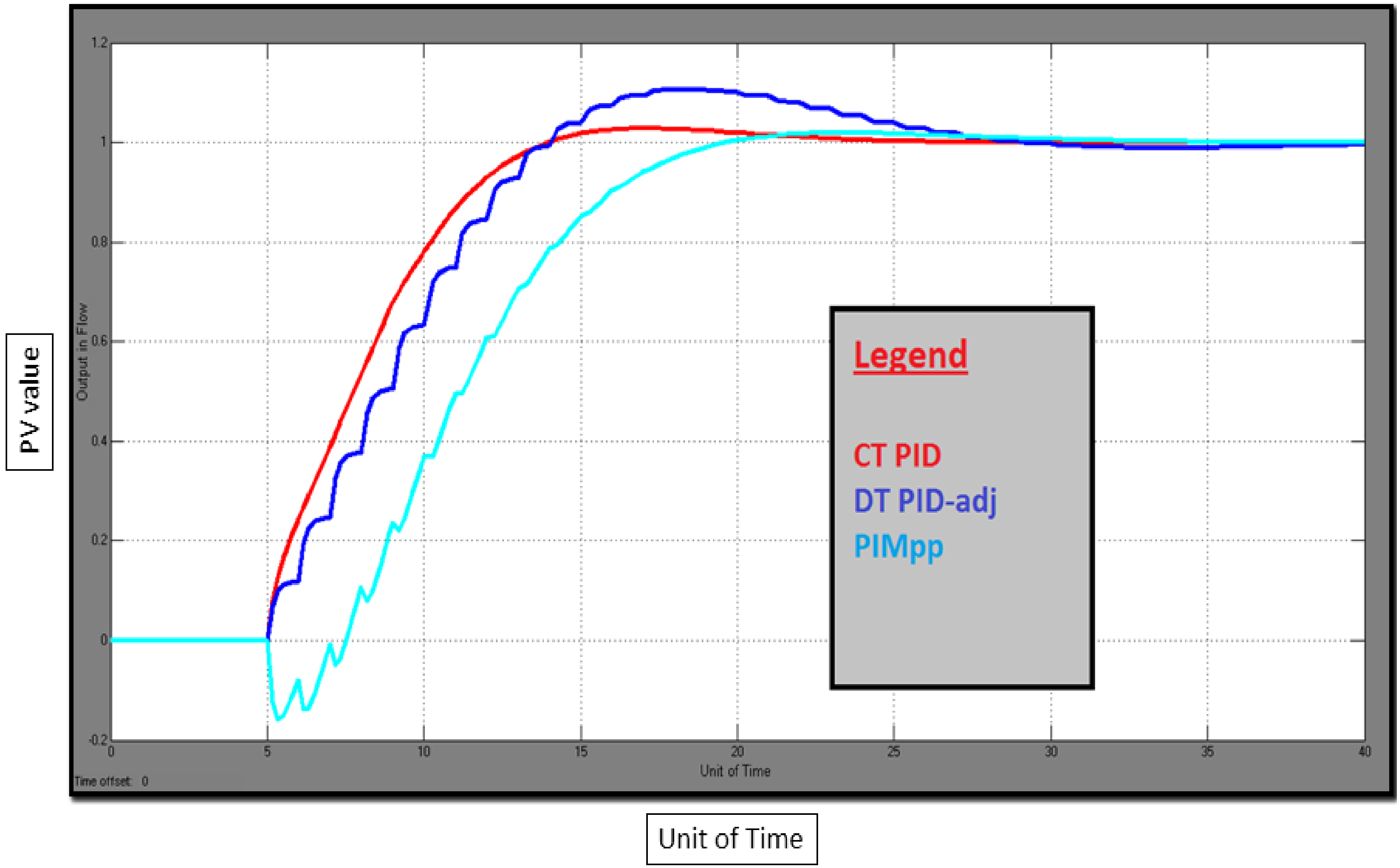
Graph of PIM vs DT PIDadj



Graph of PIMfp vs DT PIDadj



Graph of PIMpp vs DT PIDadj



5: Conclusion and Recommendation

Discretization of a controller is achieved through various methods. Degradation of performance is a norm after a controller is discretized compared to its analog counterpart. This project will deliver a simulation of PIM controller based on MATLAB code and a plant simulation in Simulink.

In this project, PIM based technique is compared to the normal approach of discretizing CT PID by adding half of interval time. Experiment is done using temperature flow controller FIC 631. With enough data, their respective transfer function is obtained through statistical modelling method. CT PID values is then obtained by using both Cohen Coon and Ziegler Nichols open loop tuning. The simulation on the other hand is obtained using model designed in Simulink and Matlab. It is discovered that different tuning method is preferred for each processes. Cohen Coon technique suits flow process the best while Ziegler Nichols performs better at temperature process. Afterward, PIM based method model is designed and its performance will be fared against the selected PID modes for each process.

It is observed that PIM design could fare the performance of DT PID in the respective criteria; settling time, overshoot, offset. Due to some limitation in PIM design, a delayed system however could not be closely emulated using PIM technique. This could pose a problem if discretization of a slow process is done. A further improvisation in derivation of PITF equation are done to achieve significant result. Pade approximation of delay coefficient is integrated into the conventional technique in multiple. These modified approaches drastically improves the transient performance and significantly a better option compared to DT PID.

As observed, it is best to vary the experiment in multiple type of plant. Different reaction may yield different outcome as they differ in reaction’s speed. While slow reaction may be prone to more error throughout the duration since the sampling time affect the overall Diophantine’s output, fast reaction may benefit from a PID controller as much as from PIM controller. More data should be collected to test the versatility of PIM based approach.

6: References

[1] G.F Franklin, J.D. Powel, and M.Workman, "*Digital Control of Dynamic Systems*", Addison Wesley, 1997.

[2] R.H Middleton and G.C Goodwin, "*Digital control and Estimation: A unified Approach*", Prentice-Hall, Eaglewood CLiffs, N.J., 1990.

[3] N. Hori, T.Mori, and K. Kanai, "*On Matched Pole-Zero Discrete-Time Models*," IEE Proc. Part-D, vol. 139, 1992, 273-278.

[4] H.H. Ray, B.G. William, and L.J Christopher, "*MRI : The Basic*," Lippincott Williams & Wilkins, 2010.

[5] M. Gopal, "*Control Systems: Principles and Design*," Tata McGraw-Hill Education, 2008.

[6] Astrom, Karl J. "*Computer Controlled Systems, Theory and Design,"* Second Edition. ISBN 0131686003. Prentice-Hall, 1990.

[7] A. H. D. Markazi and N.Hori, "*A New Method with Guaranteed Stability for Discretization of Continuous Time Control System*," Proc. American Control Conference, vol 2, 1992. 1397 - 1402

[8] Ziegler, J.G and Nichols, N. B. (1942). "*Optimum settings for automatic controllers."* Transactions of the ASME 64. pp. 759–768.

7: Appendices

7.2.1 – Delta Transform Table

18 第2章 デルタ変換法

表 2.2 デルタ変換表

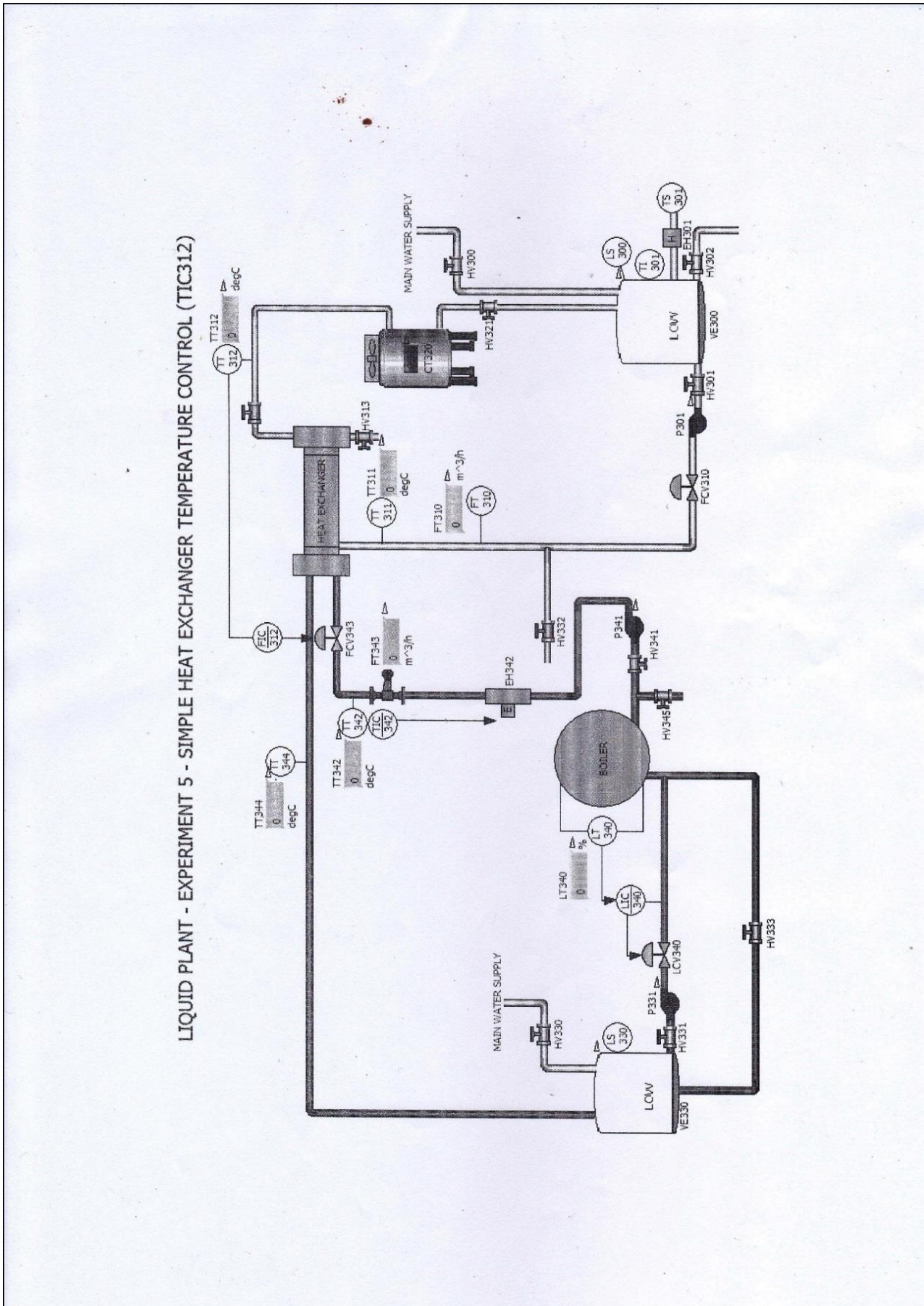
	$F(s)$	$f(t)$	$f(k)$	$F(z)$	$F(\varepsilon)$	$F(\varepsilon)\|_{T=0}$
1.	1	$l(t)$	$l(k)$	1	1	1
2.	e^{-isT}	$l(t-iT)$	$l(k-i)$	z^{-i}	$(T\varepsilon+1)^{-i}$	1
3.	$\frac{1}{s}$	$I(t)$	$I(k)$	$\frac{Tz}{z-1}$	$\frac{T\varepsilon+1}{\varepsilon}$	$\frac{1}{\varepsilon}$
4.	$\frac{e^{-sT}}{s}$	$I(t-T)$	$I(k-1)$	$\frac{T}{z-1}$	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$
5.	$\frac{1}{s+a}$	e^{-at}	e^{-akT}	$\frac{Tz}{z-e^{-aT}}$	$\frac{T\varepsilon+1}{\varepsilon+\frac{1-e^{-aT}}{T}}$	$\frac{1}{\varepsilon+a}$
6.			$(1-aT)^{k-i}$	$\frac{T}{z-(1-aT)}$	$\frac{1}{\varepsilon+a}$	$\frac{1}{\varepsilon+a}$
7.	$\frac{1}{s^2}$	t	kT	$\frac{T^2z}{(z-1)^2}$	$\frac{T\varepsilon+1}{\varepsilon^2}$	$\frac{1}{\varepsilon^2}$
8.	$\frac{e^{-sT}}{s^2}$	$t-T$	$(k-1)T$	$\left(\frac{T}{z-1}\right)^2$	$\frac{1}{\varepsilon^2}$	$\frac{1}{\varepsilon^2}$
9.	$\frac{2}{s^3}$	t^2	$(kT)^2$	$\frac{T^3z(z+1)}{(z-1)^3}$	$\frac{(T\varepsilon+1)(T\varepsilon+2)}{\varepsilon^3}$	$\frac{2}{\varepsilon^3}$
10.	$\frac{a}{s(s+a)}$	$1-e^{-at}$	$1-e^{-akT}$	$\frac{T(1-e^{-aT})z}{(z-1)(z-e^{-aT})}$	$\frac{\alpha(T\varepsilon+1)}{\varepsilon(\varepsilon+\alpha)}$	$\frac{a}{\varepsilon(\varepsilon+a)}$
11.	$\frac{1}{(s+a)^2}$	te^{-at}	kTe^{-akT}	$\frac{T^2e^{-aT}z}{(z-e^{-aT})^2}$	$\frac{(T\varepsilon+1)e^{-aT}}{(\varepsilon+\alpha)^2}$	$\frac{1}{(\varepsilon+a)^2}$
12.	$\frac{a}{s^2+a^2}$	$\sin at$	$\sin akT$	$\frac{T\sin(aT)z}{z^2-2\cos(aT)z+1}$	$\frac{(T\varepsilon+1)\left(\frac{\sin(aT)}{T}\right)}{\varepsilon^2+T\beta^2\varepsilon+\beta^2}$	$\frac{a}{\varepsilon^2+a^2}$
13.	$\frac{s}{s^2+a^2}$	$\cos at$	$\cos akT$	$\frac{Tz(z-\cos(aT))}{z^2-2\cos(aT)z+1}$	$\frac{(T\varepsilon+1)\left(\varepsilon+\frac{T}{2}\beta^2\right)}{\varepsilon^2+T\beta^2\varepsilon+\beta^2}$	$\frac{\varepsilon}{\varepsilon^2+a^2}$

$$\alpha=\frac{1-e^{-aT}}{T}\to a\ (\text{as } T\to 0),\quad \beta=\frac{\sin\frac{aT}{2}}{\frac{T}{2}}\to a\ (\text{as } T\to 0)$$

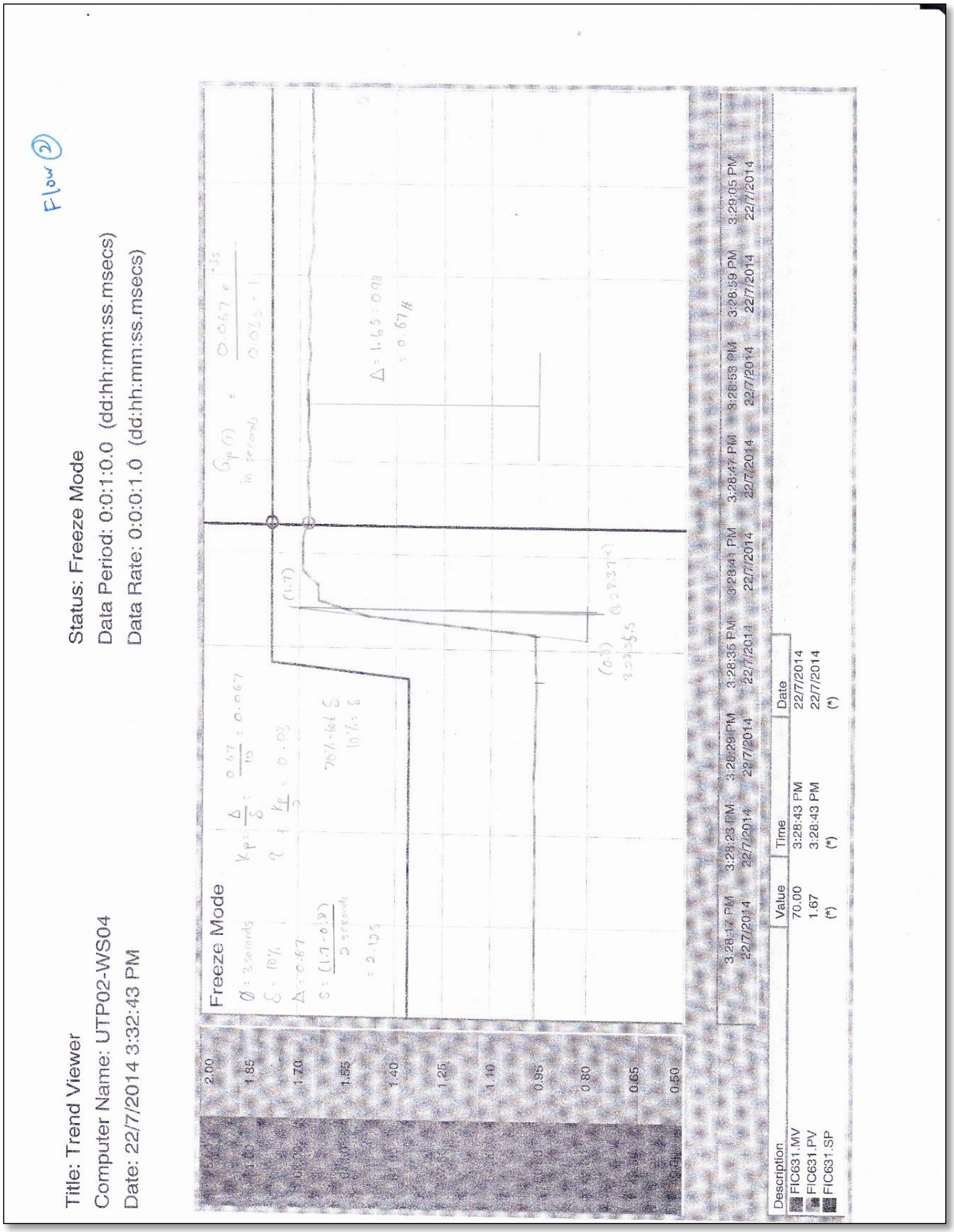
$a = \frac{1}{T}\ln(\alpha T - 1)$

$\to \alpha$ as $T \to 0$

7.3.1 – Plant's Diagram



7.4.1.1 – Flow's PRC



7.4.2 – PID Formula

Cohen-Coon Tuning Rules

	K_c	τ_{Int}	τ_{Der}
P	$\frac{1}{rK}\left(1+\frac{r}{3}\right)$		
PI	$\frac{1}{rK}\left(0.9+\frac{r}{12}\right)$	$\tau_{del}\frac{30+3r}{9+20r}$	
PID	$\frac{1}{rK}\left(\frac{4}{3}+\frac{r}{4}\right)$	$\tau_{del}\frac{32+6r}{13+8r}$	$\tau_{del}\frac{4}{11+2r}$

$$G_m(s)=\frac{k_m e^{-ds}}{\tau_m s+1}$$

Controller	k_c	τ_I	τ_D
P	$\frac{1}{K_m}\cdot\frac{\tau_m}{d}$	-	-
PI	$\frac{0.9}{Km}\cdot\frac{\tau_m}{d}$	$\frac{d}{0.3}$	-
PID	$\frac{1.2}{Km}\cdot\frac{\tau_{pn}}{d}$	2d	0.5d

The Ziegler Nichols open loop correlations