Winter 2011-01-01

Performance Analysis of Batch Reactor Temperature Control Systems

Michael Healy
Technological University Dublin

John McGrory
Technological University Dublin, john.mcgrory@tudublin.ie

Recommended Citation

Healy, M.:Performance Analysis of Batch Reactor Temperature Control Systems. Masters Dissertation. Technological University Dublin, 2011.

Performance Analysis of Batch Reactor Temperature Control Systems

By

Michael Healy

This Report is submitted in partial fulfilment of the requirements of the Master of Engineering in Pharmaceutical Process Control and Automation of the Dublin Institute of Technology

January 10th 2011

Supervisor: Dr. John McGrory

School of Electrical Engineering Systems

Declaration

I certify that this thesis, which I submit in partial fulfilment of the requirements of the ME Pharmaceutical Process Control and Automation (Programme Ref: DT702/3) of the Dublin Institute of Technology, is entirely my own work and that any content that relates to the work of other individuals, published or otherwise, are acknowledged through appropriate referencing.

I also confirm that this work has not been submitted for assessment in whole or part for an award in any other Institute or University.

Signed: _________________________

Date: ________________________

Abstract

The aim of this project was to investigate the performance of a number of key control strategies in the temperature control of batch reactors. A bench scale model was built and a batch production system was then implemented on this model. As there was no a priori knowledge of the system a number of common system identification methods were investigated.

The system was controlled using a Mitsubishi FX(2)N Programmable Logic Controller which was interfaced with a PC running ICONICS, a Supervisory Control And Data Acquisition software package. The system identification methods produced two different models for the system and these models were examined against the actual system using Matlab/*SIMULINK*, a software package used for technical computing. Then a number of tuning rules were investigated and implemented on both models with the results compared and contrasted.

The standard Industry criteria were used to compare the performance of the servo response for each controller. The PI controller using Zeigler-Nichols tuning rules was set as the bench mark. The Cascaded control strategy offered no increase in performance in the servo response in either the actual process or the *SIMULINK* models. However the regulatory response of the Cascaded strategy would offer an improvement on the performance of the PI controller. The performance of the Smith Predictor was limited due to the minimal time delay relative to the time constant.

The Integrating method proved to offer an improvement on the two point method in terms of system performance and in the time required to identify the initial controller. Also the Smith Predictor offered a slight improvement in both the laboratory model and in the Matlab/*SIMULINK* simulations.

Table of Contents

Table of Figures

1. Introduction

1.1 Background

The project is based on the automation of a Bench Scale Batch Reactor and the problems associated with accurate temperature control. The project is an investigation into an existing problem which can be experienced during the initial start up of the reactor, the problem can occur when the temperature control is optimized around the nominal operating temperature range and thus the large difference between the ambient temperature and the desired setpoint can cause an overshoot in the temperature response. The problem can be solved by manually controlling the temperature until it is within an acceptable range around the setpoint and then switching in the controller. However this is not desirable as manual control depends entirely on the competency of the operator and variations between different operators can cause discrepancies between different batches.

There are a number of aims and ideas explored here, which can be broken up into different sections, these being:

1) A comparison of the commonly used empirical modelling methods to identify an accurate model for the process. The modelling methods used here produce a model of the system which is either in the form of a First Order Plus Dead Time (FOPDT) or a Second Order Plus Dead Time (SOPDT) model.

2) A comparison of the performance of a number of control strategies in the temperature control of a bench scale batch reactor. These strategies will include the most commonly used control strategies in the Pharmaceutical Industry where the performance of each control strategy will be compared using standard performance criteria. As each control algorithm will be analysed using the same process and with similar same conditions an accurate comparison is possible.

3) The process will also be modelled using simulation software; an example of such software is MATLAB from MathWorks®, to investigate the difference in the empirical modelling and the actual system.

In spite of innovations in predictive and advanced control techniques the majority of chemical industries still use Proportional Integral Derivative (PID) loops. In industrial process control applications more than 95% of the controllers are of PID type. The maintenance and operation of PID controllers are well known and are relatively easy to understand. The PID controller is also robust in nature when tuned correctly and thus is ideal for a wide range of applications. (1)

1.2 Aims/ Goals

There a number of goals proposed, including:

- Designing, Building and Implementing a Bench Scale Batch production system using the ISA S88 Standard.
- Developing an efficient and transparent SCADA system to control the system.
- A comparison of PI, Cascaded PI-P, and PI with a Smith Predictor control strategies.
- An analysis into the effects of a digital implementation of the control strategies.
- Investigating and analysing the effects of using a 1st order model to estimate an Nth order system.

1.3 Issues

There are a number of issues that can occur when undergoing a practical project such as the scalability issue. This is where the control strategy works for the bench scale but isn't as efficient when scaled up. This problem is down to the high ratio of surface area to the volume of the reactor, in a production size reactor the ratio of surface area to reactor volume is much less and hence the heating characteristic's cannot be scaled up. The results obtained here cannot be assumed to be directly comparable but they can be used to give an indication as to the expected performance of the production scale reactor.

Also the pharmaceutical industry is a very secretive industry and obtaining any information is very difficult, this is down to the highly competitive nature of the business. This practical information can be very useful in determining whether an idea or concept is viable.

There is an issue that exists when starting up some process; this issue is down to the method of control and the tuning of same. This is usually overcome using a manual start-up handled by an experienced Operator. This problem can occur in batch reactors even in the bench scale so the identification of a method that does not need a manual start up is clearly advantageous as it remove the manual intervention and the risk of operator error.

1.4 Summary of Chapters

The introduction to the project is given in **Chapter 1: Introduction**; this chapter gives a brief overview of the main concepts and ideas that are subsequently covered in greater depth in latter chapters. The goals and aims are also outlined with the reasoning and benefits behind them covered.

The next section is **Chapter 2: Literature Review**; in this chapter the literature that has been published in relation to the project is reviewed. This contains information about who, where and how the previous research has been carried out. The information is taken from research papers, journal publishing and catalogues. The aim of the chapter is to offer the reader a structured view of the current field of study.

In **Chapter 3: Methodology** the methods for data capture are covered along with the concepts of how the process operates. The process is described here including all the individual components and how they are interconnected. Also the method to translate the actual process into a simulation model is discussed along with the performance metrics that are used to compare different methods.

In **Chapter 4: Theoretical Background** the equations that define the operation of the heat transfer of the system are described at a basic level to give an estimation as to the performance of the system. The methods of implementation for the different control strategies are also discussed.

In **Chapter 5: Results** the findings of both the simulation models and the actual implementation are shown. The performance of each method is given and is then compared and contrasted. The results are mainly presented in graphical form as this provides a quick and clear indication as to the performance of each method.

In **Chapter 6: Conclusion and Discussion** the results and findings in this project are discussed and are brought together. The area where future work could potentially arise is dealt with in this section. And also included is a summary of the project.

2. Literature Review

2.1 Introduction

There is a large quantity of literature in the area of Control Systems; this is because control problems exist in many different industries. The use of control theory occurs in the Medical Industry along with the Manufacturing and Pharmaceutical Industries to name but a few. This means that there is huge interest in the development of new methods and in improving existing technology, which has led to a situation where there is a vast number of different methods but there is no hard fixed acceptance of when to use each method nor has there been an extensive direct comparison of the advantages and disadvantages of each method.

2.2 Feed Back Control

The process under inspection is a laboratory scale batch-reactor where the production of a batch is simulated. The aim is to replicate fully an industrial approach where the temperature needs to be maintained and controlled to within certain criteria. There are a number of different control methods available to achieve this. The control structure can be off two forms, Open loop and Closed Loop. The difference between the two arrangements is the existence of feedback in the closed loop system.

Feedback can be defined as a process through which a signal travels through a chain of causal relations to re-affect itself (2). An example of a negative feedback system that can be seen in everyday life is the domestic heating system seen in most homes, when the thermostat detects that the temperature of your home drops below the desired temperature the heating is turned on.

As open loop control systems do not contain any feedback they are rarely, if ever, used in industry but they can be seen in everyday life in the form of a toaster for example. The toaster operates on a timer and when the timer is reached the toast will ‘pop’ no matter how well cooked the toast is i.e. there is no sense of well done the toast is.

There are a number of different types of Closed Loop control systems available and these are examined in the next section, where each controller structure is critically examined and the final choice of controller is specified.

In the pharmaceutical sector, accurate temperature control is specified because any variation in temperature can adversely affect the yield, quality, purity, efficacy, potency and in general result in a compromised product.

2.3 System Identification

The first important step in any control system problem is to find out as much information about the process as possible. The information can then be used to determine the model for the system, with the most prevalent being the First Order Plus Dead Time (FOPDT) and the Second Order Plus Dead Time (SOPDT).

There are two different types of methods that can be used to identify a model as a representation of the actual system, these are defined here as:

1. **Theoretical model**

The use of a theoretical model requires, even for a simple system, a number of differential equations. The development of these equations is not always simple and thus can be largely time consuming. For complex processes if the model requires a large number of differential equations with a significant number of unknown parameters the equations may be insolvable.

The other problem with identifying a theoretical model is that the assumptions that are needed can cause large variations that can create unrealistic systems. The sensitivity of the model to the assumptions means that it is not always possible to achieve an accurate model.

2. **Empirical model**

The most simple and widely used method to identify any process is the step test; this is where a step is inputted into an open loop system with the output recorded to produce a Process Reaction Curve (PRC). An estimation of the system can made from the PRC and a FOPDT model can be obtained in the form $G_p(s) = \frac{Ke^{-\theta s}}{\tau s+1}$.

The process dead time, θ, is the time period following an upset during which the controlled variable is not yet responding. The time constant, τ, is a period between the time when a response is first detected and the time when the response has reached 63.2 % of its final (new steady-state) value. The gain, K, is to change in the controlled variable relative to the change in the controller output. (3) The PRC structure to estimate a FOPDT is given in Figure 2.1, where the two point method is used.

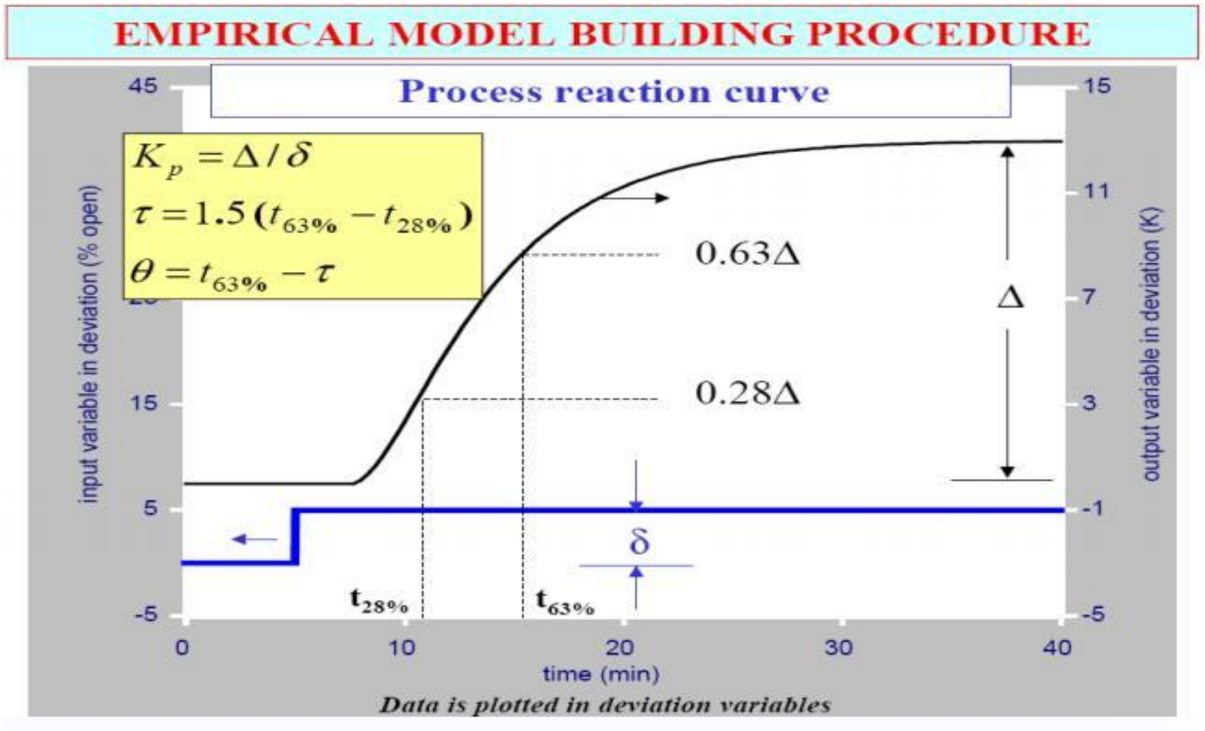

Figure 2.1 : Model Building using a Process Reaction Curve and the Two Point method (Courtesy of Aidan O'Dwyer, DIT)

The two point method has been shown to be an improvement on the tangent and point method. The accuracy of the model using the two point method is much higher than that of the tangent and point, and this is down to the discrepancies between the positioning of the tangent to the slope. The two point method is a mathematical method whereas the tangent and point requires a graphical reconstruction.

There is an alternative method as introduced by Yuwana and Seborg (4) which allows for FOPDT model to be estimated from the closed loop step response. The results from 2002 (5) show that the FOPDT model obtained using this method is a close representation of the equivalent process. The closed-loop methods are often more preferable in industrial applications in comparison to the open-loop methods this is because they can be executed in the knowledge that the process is under some manner of control. This is because closed loop method removes the possibility of process runaway that could potentially occur with the open-loop method and also the closed loop method can be completed with minimal disruption to the system.

The online method as described by O' Dwyer and Kealy (5) is illustrated in Figure 2.2 where the trend appears to be in the form of an under-damped SOPDT.

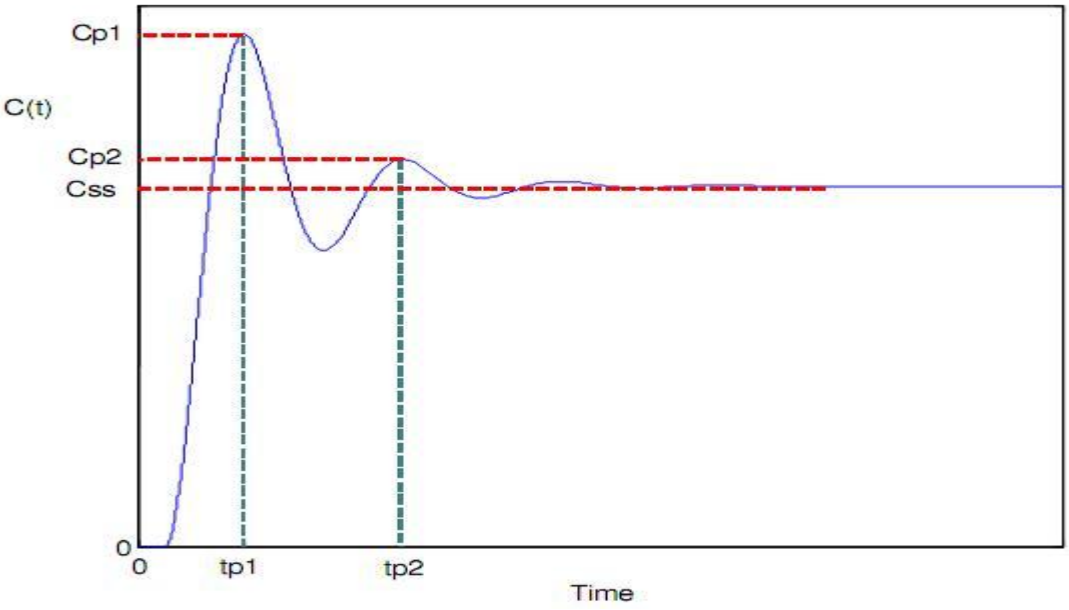

Figure 2.2 : Online tuning method

The method involves introducing a setpoint change to the existing system with the controller chosen either voluntary or involuntarily to exhibit an oscillatory response. This response is then approximated as an under-damped SOPDT as described be Mamat and Fleming (6).

This method can be used in the event of an oscillatory response; however for this method to be used there needs to be an existing controller in place. In this project there is no existing controller and thus this method cannot be used initially. The focus of this project is on the initial starting point for controlling new processes. Thus the system identification methods must come from the PRC. This method can be used in retrospect after the initial controller is implemented.

The above methods are used to identify a FOPDT model to estimate the system as quite often the FOPDT can accurately represent higher orders systems. However there are also a number of methods to ascertain a SOPDT in the event that the FOPDT model is unable to give an accurate representation of the actual process.

One such method to indentify a SOPDT (7) is given here:

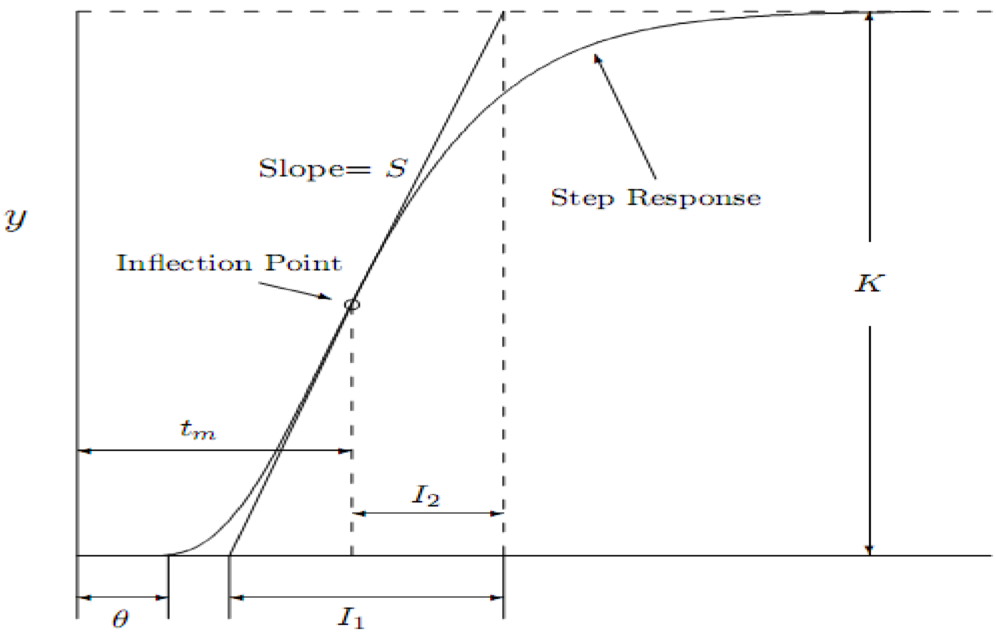

Figure 2.3 : SOPDT Model Identification method

The model that is obtained from this method is in the form of

$$G_p(s) = \frac{Ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{2.1}$$

Where:

K is the process gain

θ is the dead time

τ_2, τ_1 are the two time constants

The length of the segments I_1 and I_2 are measured and then used to find the time constants from the following equations and these are given here:

$$\frac{\tau_1}{I_1} = \left(\frac{\tau_2}{\tau_1}\right)^{\frac{\tau_2}{\tau_1 - \tau_2}} = (x)^{\frac{x}{1-x}} \tag{2.2}$$

And

$$\frac{\tau_1}{I_1}.x = \frac{\tau_2}{I_1}, \ where\ x = \frac{\tau_2}{\tau_1} \tag{2.3}$$

Also the point $\frac{\tau_1}{I_1}, \frac{\tau_2}{I_1}$ also satisfies the following equation

$$\frac{\tau_1}{I_1} + \frac{\tau_2}{I_2} = \frac{I_2}{I_1} \tag{2.4}$$

Thus the two time constants can be derived from the measured line segments giving a SOPDT model for the system. The advantage of this method is that there is no additional experimental work than the method to obtain the FOPDT model i.e. the same PRC can be used for the FOPDT and the SOPDT models. This allows a direct comparison of both methods. It is usually accepted that the FOPDT model can be used to approximate higher order processes with accuracy that is sufficient in most cases, however the potential advantages of having the accuracy of a higher model have not been fully exploited.

It is found that for open-loop processes that have dynamics of SOPDT, the efficiency of PID control is mostly around 65% only. But for FOPDT process, the efficiency is very close to 100%. Thus, compared with the optimal simple feedback system, the PID controller is not so efficient for SOPDT processes.(8)

There is also another method that is used to indentify controller parameters using the PRC; however this method does not need the full PRC. This method is referred to in this project as the Integrating method. The integrating model is used in process's such a liquid levels in a vessel where when a valve is open the level will continue to change in a linear fashion. (9)

This method works here by approximating the initial rise in the system's PRC as an Integrating system. Since this method only needs the initial rise it can be done in a much shorter time frame and also in a safer manner since the process is only active for such a short time.

The model produced by this method is seen in Equation (2.5).

$$G_p(s) = \frac{Ke^{-\theta s}}{s} \tag{2.5}$$

The gain, K, is found in the same manner as the gain for the FOPDT method where $K = \frac{change\ in\ output}{change\ in\ input}$ and then the time delay is easily identified as the time required for a change in the output. The condition for which this method is applicable is that for $t < \theta$ the output is zero. Also for $t > \theta$ the output is equal to $K * change\ in\ input * (t - \theta)$.

The Integrating model does not offer a full model for the entire system but rather an approximation of the initial rise. However there are methods to deduce controller parameters from the integrating model and thus a full model is not needed.

2.4 Controller Types

There are number of main controllers that are used in industry and these include the following:

- The Proportional Integral Derivative (PID) controller, including P only, PI and PD
- The Cascade control algorithm (usually containing some form of PID control)
- The Smith Predictor (or another form of Dead Time Compensator)

2.4.1 PID Control

The use of PID controller's is widespread in process industries which results in extensive solutions to the problem of tuning PID controller parameters in single input single output (SISO) systems. The de facto tuning rules are those developed by Ziegler and Nichols in 1942 (10). There are other important tuning rules such as are those introduced by Cohen and Coon in 1953 which are similar to the Ziegler-Nichols rules but can offer a slight improvement in certain situations. The PID control used in this project is mainly PI control with the derivative term set to zero. The initial tuning rules used here are the Zeigler-Nichols; however these are only the start point. The final part of tuning is a manual trial and error process. The efficiency and efficacy of this tuning is greatly increased with experience of tuning similar systems.

2.4.2 Cascade Control

The cascade control approach to temperature control is used in a number of different industries for a wide range of applications. The improvement in system performance based on cascade control approach over the conventional one from the point of improved time response and relative stability is verified through simulation results. Cascade control is used to improve the dynamic response of a feedback control loop to disturbances in the manipulated variable. (11)

The implementation of a cascade scheme changes the characteristic equation of process control system and as a consequence affects its stability. The effect of the cascade control strategy on the overall loop stability can be seen by performing a software based simulation. (12)

The use of cascade control can also be seen in the control of central air-conditioning system. (13)

Moreover, when disturbance to the control variable or non-linear final control element are included in the system, cascade control can be preferred in order to improve the closed-loop response.

There is information in the published literature on the tuning methods of cascade controllers (14) but it is rather limited. The standard method is to tune the inner loop first and then tune the outer loop. Lee and Park (1998) proposed a new method, finding the ideal controller that gives the desired closed-loop response and then finding the PID approximation of the ideal controller by Maclaurin series. The method, which can be applied to any open loop stable processes, enables us to tune the PID controller both for the inner loop and the outer loop simultaneously.

It is important to decide when to use cascade control, because it requires at least two measuring elements instead of one. According to Krishnaswamy, Jha and Deshpande (1990), the criteria to use cascade control are:

1. The inner loop is faster than or as fast as the outer loop
2. The disturbance affects the inner loop
3. There a two measured elements available and there is a causal relationship between the two

The most common cascade controller is chosen as PI-P because it has only three tuning parameters and gives a good performance. The inner loop needs to be as fast as possible and thus the proportional term alone helps to achieve this and also there is no need to have two integral terms as having a integral term in the outer loop will ensure that there is zero steady state error and thus the inner integral term is redundant.

2.4.3 Smith Predictor Control

The concept of the Smith Predictor has received much interest since the idea first came about in the 1957 and hence has been improved upon for many different applications. The discrete implementation of the Smith Predictor is covered well by Vodencarevic (15). The Smith Predictor is a dead time compensator and the key idea of the Smith Predictor is to isolate the time delay term from the linear component.

The problem with systems that have a long time delay is that the performance of the PID controller is very limited as to the specifications that it can meet. This means that the PID controller is not as effective as it could be and this is where the Smith Predictor is advantageous. The problem that the time delay introduces is that the gain of the controller has to be detuned as to prevent a massive overshoot and a ringing effect. This reduction in gain prevents this but introduces a sluggish response from the system.

The ideal form of the Smith Predictor is shown in Figure 2.4, where the estimate of the plant is fed back and compared to the setpoint to negate the delay.

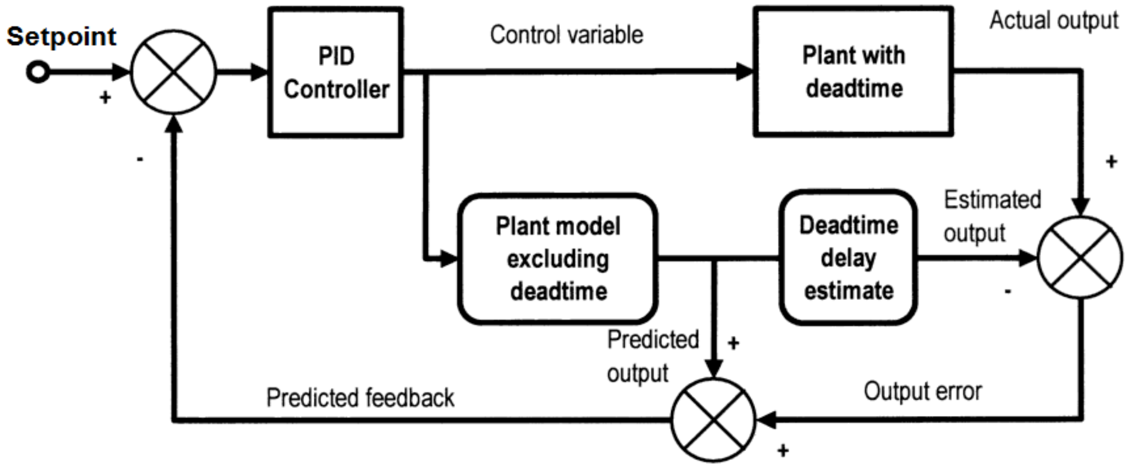

Figure 2.4 : Ideal Smith Predictor

2.5 System Implementation

The implementation of the controllers that are used in industrial applications are operating in discrete time while the process is usually a continuous time system, ideally both the controller and plant would be continuous time systems. The discrete time controllers are down to the use of computers or PLC's as the basis for implementation of the controllers.

The continuous controller can be seen in Figure 2.5 where the discrete approximation can be seen in Figure 2.6.

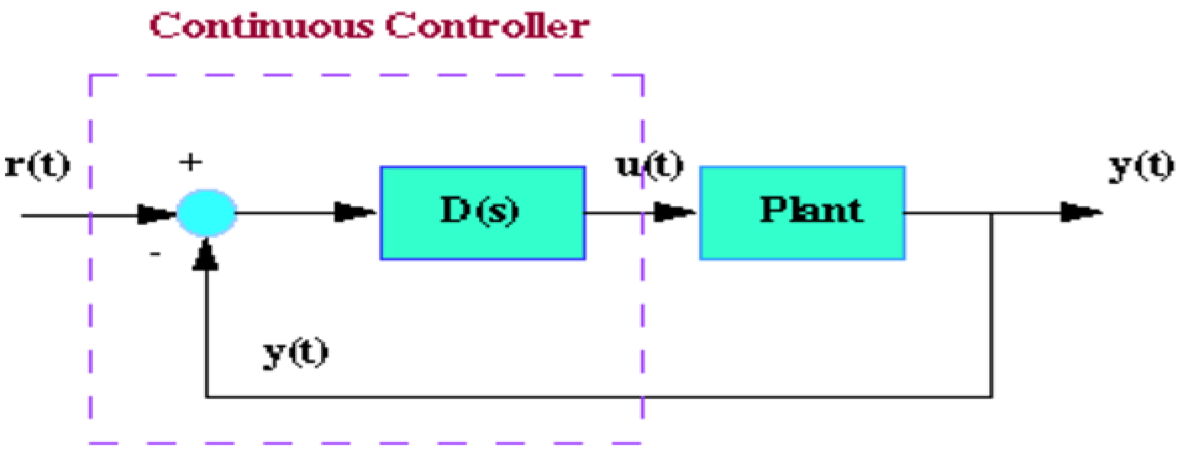

Figure 2.5 : Continuous Controller (Courtesy of Carnegie Mellon University)

The continuous controller can be approximated by a difference equation created from a zero order hold transformation. This is then implemented at a defined interval.

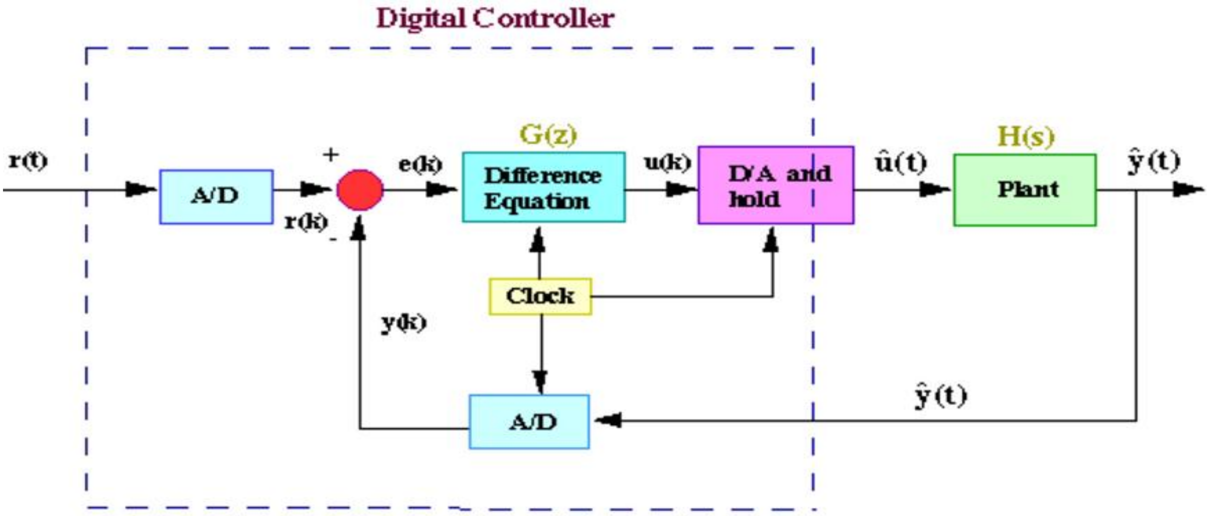

Figure 2.6 : Digital Controller (Courtesy of Carnegie Mellon University)

The majority of tuning rules assume that the controller is implemented in a continuous time mode and this introduces a problem that is caused by the sampling of the continuous process. The problem is that the discrete transformation introduces a delay term which is approximately equal to half the sampling time. This can be accounted for when designing discrete controllers to negate the effect of the sample and hold.

2.6 Summary

PI/PID control systems have been widely used in the industrial process control and the performance of these control systems are dependent on the dynamics of the open-loop process. Huang and Jeng (16) indicate that if the controller in a simple feedback loop has a general form, not constrained to the PI/PID controller, then the minimum attainable IAE criteria value for a unit step input is found to be 1.38θ. This is the best achievable value with an ideal controller and simple system. This IAE value can be used to identify when the system has reached its maximum performance.

In this chapter the literature describing the system identification methods have been introduced along with the different control methods. Also the methods which are used to implement these control strategy's have been expressed.

3. Methodology

3.1 Introduction

The test rig used for this project was designed and built to simulate an actual batch reactor operating environment. The test rig is build exclusively from stainless steel. The use of stainless steel in the production of reactor vessels in industrial application is very common due to its high resistance to corrosion and good heat transfer characteristic. The heating is supplied by a 3kW heating element in separate vessel. The capacity of the reaction vessel is approximately 7 litres while the jacket contains 2.5 litres. The heater has a capacity of approximately 6 litres given the overall total volume of the heating system at circa 10 litres. The Piping and Instrument Diagram (P&ID) for the system can be seen in Figure 3.1.

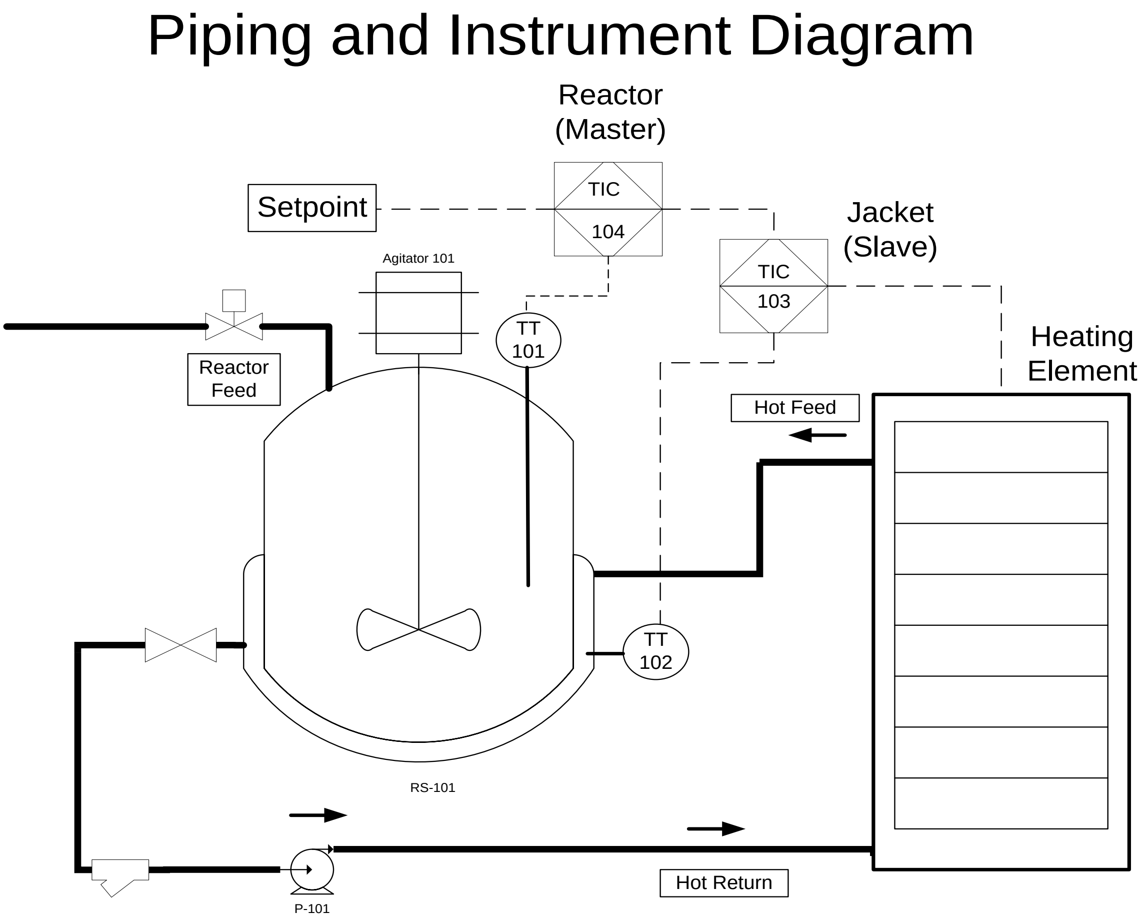

Figure 3.1 : P&ID for the system

The P&ID in Figure 3.1 shows the system with the cascaded control structure. This initial system is operating only with the control of the heating element (i.e. Hot Water), while the control of the cooling water is added to improve the performance of the system. The initial system can be seen in Figure 3.2.

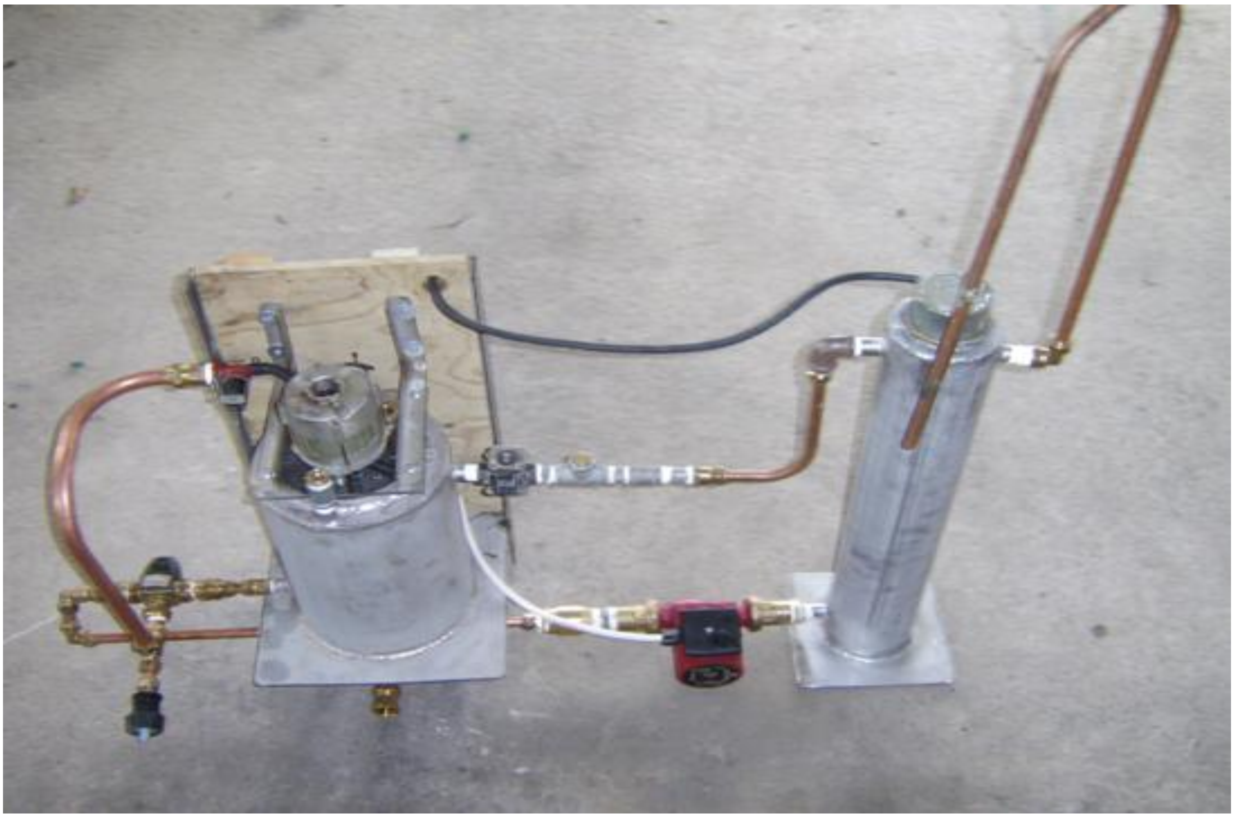

Figure 3.2 : The actual process

The heating component can be seen on the right of Figure 3.2, where the heating element is contained in a separate vessel to that of the actual reactor. This would allow for multiple reactors to be connected to the same heating source as would be the case in most industrial applications. However there are a number of small changes that would be needed to implement such a system the most important change would be to change the final element. The final control element is the heating element and this would need to be change to control of the flow with either a modulating valve or by using different pumping equipment.

The temperature probes are contained in both the jacket and in the product enclosure. The probes are fully immersed in both cases and give an accurate representation of the temperature. The jacket temperature probe is a Resistance Temperature Device (RTD) while the product probe is a Thermocouple. The accuracy of both devices is very similar over this temperature range as they are very accurately calibrated. They are both operating over the 4-20mA range and are converted to digital form using a 12 bit Analog to Digital Convertor (ADC).

The agitation in the product vessel is provided by a Permanent Split Capacitor (PSC) motor that is designed to run at 3650 RPM. This level of rotation causes the formation of vortexes in the product vessel. To counteract this, the motor would ideally be adapted to run at a lower speed however accurate speed control on a PSC motor needs a Variable Frequency Drive (VFD). But this is not available so the motor will instead be controlled by switching it on and off in a method similar to that of a Pulse Width Modulate (PWM) control. The motor will be switch on for 3 seconds and off for 5 seconds on a repeated loop. This provides consistent agitation and the reactor can be considered a continuously stirred batch reactor.

3.2 System identification

The first method used to find a suitable model for system is the Step test as described in Section 2.3, where the Process Reaction Curve obtained from the system is depicted in Figure 3.3.

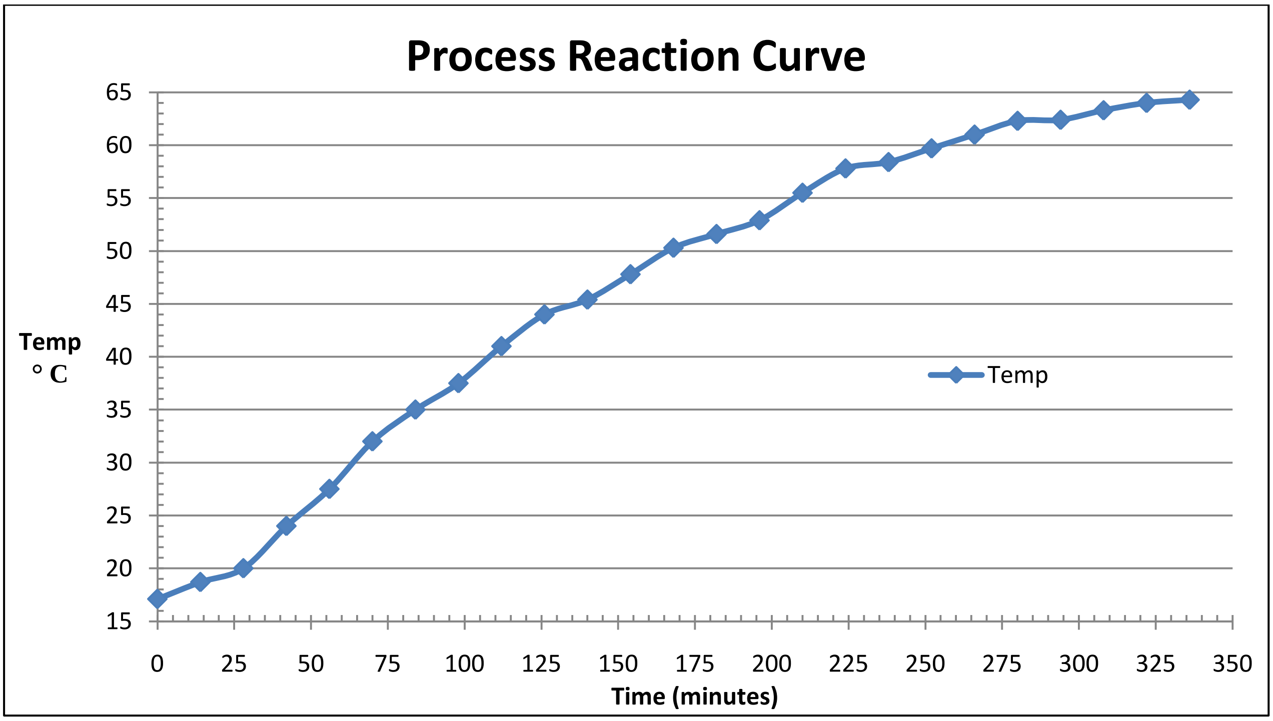

Figure 3.3 : The PRC for the system

The input to the process is 18.75% (or 375 counts/ decimal representation) of the maximum output from the controller. The standard two point method is then used to derive the FOPDT transfer function in the form of $G_p(S) = \frac{k_p}{\tau_p S+1} e^{-\theta S}$ where in this case we have:

$$k_p = \frac{64.3-17.1}{375} = 0.126 \frac{°C}{Count}$$

$$\tau_p = 1.5(t_{63\%} - t_{28\%}) = 127.5\ minutes$$

$\theta = t_{63\%} - \tau_p = 20\ minutes$

This gives the following FOPDT transfer function as a representation of the system (in Minutes):

$$G_p(S) = \frac{0.126}{127.5S + 1} e^{-20S} \qquad (3.1)$$

There are a number of other methods to find a model for the process such as that described by Oldenbourg and Sartorius (7) which results in a SOPDT transfer function in the form

$$G_p(s) = \frac{Ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \qquad (3.2)$$

To find the SOPDT transfer function the same PRC is used as in Figure 3-1. The Gain of the SOPDT is the same as that of the FOPDT, while the dead time is estimated graphically as depicted in Figure 3.4. The two time constants are inferred from the length of the line segments I$_1$ and I$_2$ which are measured from the PRC.

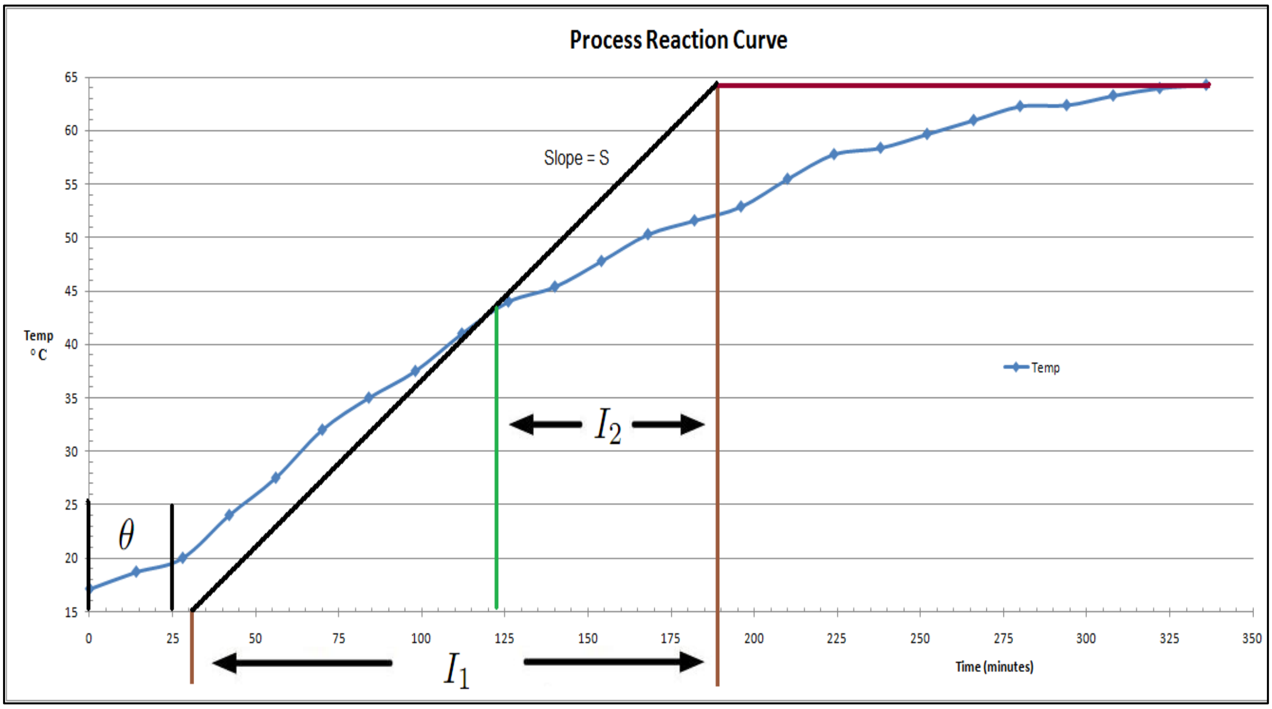

Figure 3.4 : The PRC for the system with the graphical construction to identify the SOPDT

The resulting SOPDT transfer function was found to be $G_p(s) = \frac{0.126e^{-25s}}{(2s+1)(97s+1)}$ which can be rewritten in the form $G_p(s) = \frac{0.126e^{-25s}}{194s^2+s+1}$ with the time constants and delay in terms of minutes.

There are numerous tuning rules that can now be used to define a controller that will achieve a good level of performance from the process. The performance can be measured by a number of ways that allow a direct comparison of different controller types, and these are mainly concerned with the servo response of the system, although there are performance indicators for the regulatory response of the system.

The second set of performance criteria are related to the properties of the closed loop system in the frequency domain. These are the phase margin and the gain margin which can be found from the bode plot of the open loop system (i.e. $G_{ol}(S) = G_p(S)G_c(S)$ where subscript "ol" is open loop, "p" is process and "c" is the controller).

3.3 PLC and SCADA System

The Mitsubishi FX(2N) PLC is used to control the system; it contains a 12 bit Analogue to Digital Convertor along with a 12 bit Digital to Analogue Convertor. The Discrete control algorithm is implemented with a sample time of 1 second. This is fast enough to detect any changes in the temperature of the system. The program is organised using Sequential Function Charts and is written in a combination of instruction list and ladder logic.

The SCADA is designed using Iconic's Inc. Genesis GraphWorx package and is connected to the PLC using the Object Linking and Embedding for Process Control (OPC) from Kepware®. The OPC communicates with the PLC via a serial RS-232c connection. The OPC converts the machine code into decimal representation and this allows any computer program to interpret the data.

3.3.1 User Interface Design

The Visual Design of a user-computer interface affects both the user's initial impression of the interface and the systems longer-term usefulness. Visual design comprises all the graphic elements of an interface, including overall screen layout, menu and form design, use of colour, information coding, and placement of individual units of information with respect to one another. Good visual design strives for clarity, consistency, and attractive appearance.

If the meaning of an image is readily apparent to the viewer, we have visual clarity. An important way to achieve visual clarity is to use the visual organization of information to reinforce and emphasize the underlying logical organization. There are just a few basic visual-organization rules for accomplishing this end. (17)

The visual-organization rules concern similarity, proximity, closure, and good continuation. The rule of similarity states that two visual stimuli that have a common property are seen as belonging together. Likewise, the rule of proximity states that two visual stimuli that are close to each other are seen as belonging together. The rule of closure says that, if a set of stimuli almost encloses an area or could be interpreted as enclosing an area, the viewer sees the area. The good-continuation rule states that, given a juncture of lines, the viewer sees as continuous those lines that are smoothly connected.

The rules are applied to improve menu organization. The best menu uses proximity to form groups and similarity of indentation to show the two-level logical structure. Consistent application of visual-organization rules and coding, and consistent combination of visual elements into higher-level graphic objects and icons, constitute another important element of visual design. (17)

Visual elements can be thought of as letters in a graphic alphabet, to be combined into words whose meanings should be obvious to the viewer. Consistency must be maintained among as well as within single images; a consistent set of rules must be applied from one image to another. In coding, for example, it is unacceptable for the meaning of dashed lines to change from one part of an application to another. For placement consistency, keep the same information in the same relative position from one image or screen to the next, so that the user can locate information more quickly.

Visualization provides an ability to comprehend huge amounts of data; it also enables problems with the data itself to become immediately apparent. Visualization often reveals things not only about the data itself but also about the way the data is collected with an appropriate visualization, errors and patterns come to the fore. Visualization allows understanding of both the bigger picture and the finer details. (17)

3.3.2 SCADA Components

The SCADA was designed using the rules as defined in the previous section. The automated valves used in the system are depicted as seen in Figure 3.5 and Figure 3.6 whereas the pump used is depicted in Figure 3.7 and in Figure 3.8. Also the colour code to represent the state of operation used here is indicated in the following images.

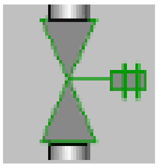

Figure 3.5 : Valve Off

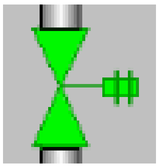

Figure 3.6 : Valve On

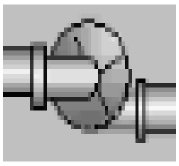

Figure 3.7 : Pump Off

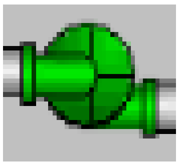

Figure 3.8 : Pump On

The colour code follows an intuitive and easy to understand pattern. The colour green is associated with ”GO” (as used in the traffic light system) and here it is used to represent a “running” or “open” element. This same colour scheme applied to all the elements contained in the process.

When the element is not "ON" the colour changes to a "Greyed out" appearance which is a clear contrast to the vibrant green. The Grey colour symbolises that the element is not in use and is not running.

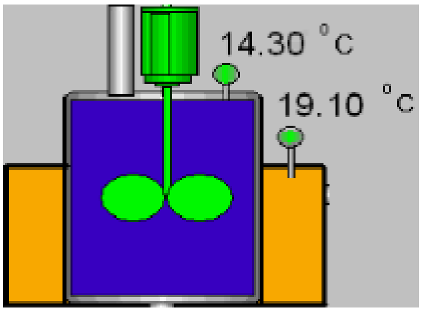

Figure 3.9 : Cold Product

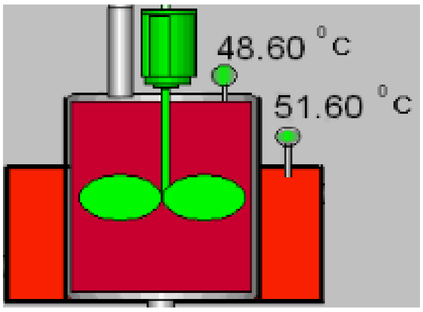

Figure 3.10 : Hot Product

The temperature of the jacket and product are important parameters and thus are animated to give a clear indication as to their status. This is shown by using a colour scheme; the colours chosen are the simplest representation of temperature and also the most common. The colours are blue for cold and red for hot. This is shown in Figure 3.9Figure **3.9** Figure 3.10 where the temperature of the product can be seen with the need to review the temperature reading. This animation provides a very simple but important status update on the product temperature.

The overall SCADA homepage can be seen Figure 3.11, where the SCADA is designed to represent the actual process as closely as possible.

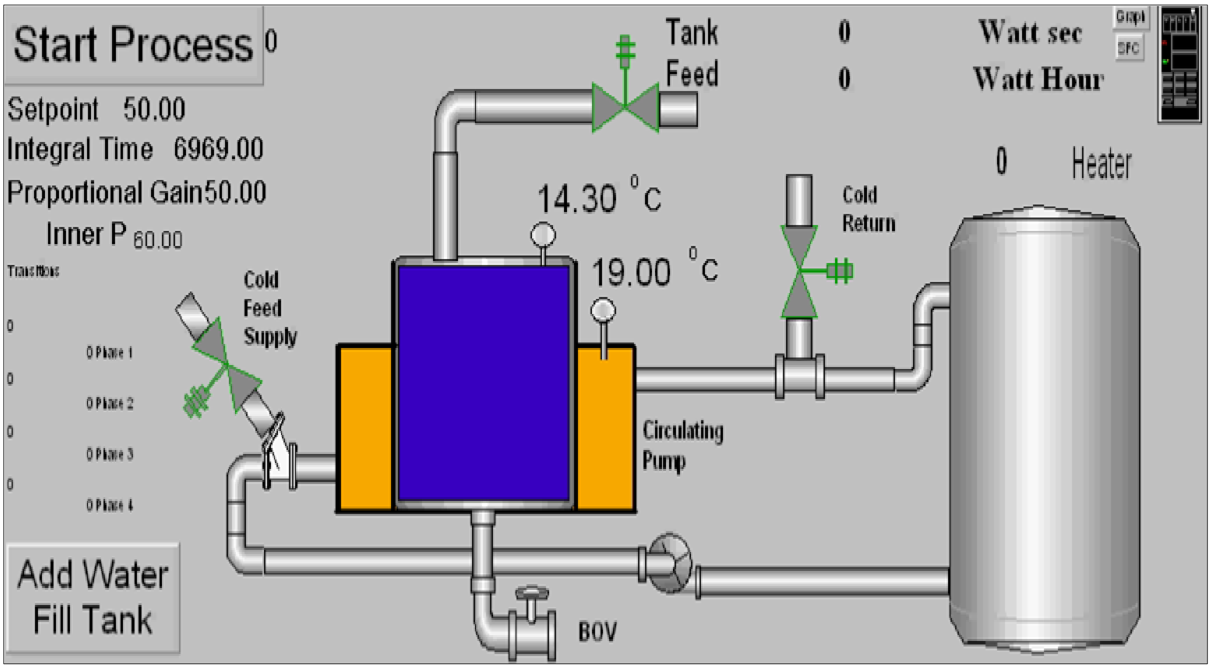

Figure 3.11 : SCADA image

3.4 Simulation Software (Matlab)

To simulate the continuous time domain transfer function in the discrete time domain we need to make some approximations, the first of these is the estimation of the time delay (also known as dead time). This dead time exists because no measurement or response to a process can be truly instantaneous. Although all systems have some degree of dead time, too much dead time can lead to problems with system response. If the dead time is not appropriately accounted for, the lag in data readings can have detrimental effects on the implementation of control (18).

The transfer function for the time delay is given here as:

$$H(s) = e^{-\theta s} \tag{3.3}$$

Among the many methods to estimate time delay, Pade approximations are the most frequently used methods to estimate dead-time by a rational function (19). The 2nd order Pade approximation in particular is the most widely used, although hider orders can be used (20). It can be seen that the Pade-approximations give a somewhat accurate expression for the time-delay, but in this project it will only be used where the exact transfer function $e^{-\theta s}$ cannot be used directly.

The majority of ordinary differential equation numerical integrators, such as Matlab, require pure differential equations that have no time delays. As the system consists of differential equations with time delays, the Pade approximation can be used to convert them to delay-free differential equations, which can then be numerically integrated.

The time delay term can be written as (21):

$$e^{-\theta s} \approx 1 - \theta s \tag{3.4}$$

Where the first order Pade Approximation can be written in the form:

$$e^{-\theta s} \approx \frac{1 - \frac{1}{2}\theta s}{1 + \frac{1}{2}\theta s} \tag{3.5}$$

While the more accurate 2nd order approximation can be seen below:

$$e^{-\theta s} \approx \frac{1 - \frac{1}{2}\theta s + \frac{\theta^2 s^2}{12}}{1 + \frac{1}{2}\theta s + \frac{\theta^2 s^2}{12}} \quad (3.6)$$

Where θ = Dead Time

The comparison of the 1st and 2nd order Pade approximations can be seen in Figure 3.12 where they are plotted against a pure time delay for a simple step test. As can be seen the difference between the two approximations are very minimal and should not have any major impact in any of the further simulations.

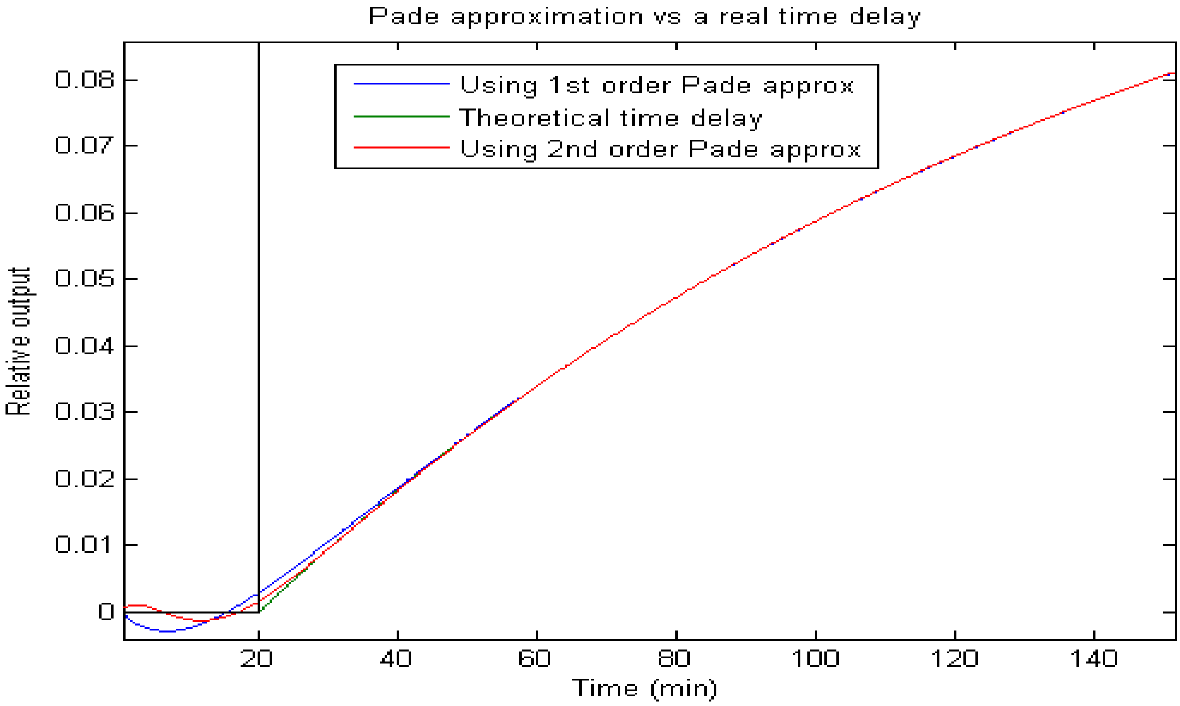

Figure 3.12 : Pade Approximation vs. pure time delay

Also in Figure 3.12 it can be seen that the first-order approximation has an inverse response initially, while the second-order approximation has a double inverse response. This is due the single positive zero for the first order approximation, and there are two positive, complex-conjugate zeros in the numerator of the second order transfer function.

The time delay can also be easily handled in *SIMULINK* with the added advantage that no approximation is required. The first phase of the analysis is to produce a step test to compare the two process defining transfer function namely the FOPDT and the SOPDT. The *SIMULINK* model can be seen in Figure 3.13, where the delay term is implemented as a transport delay.

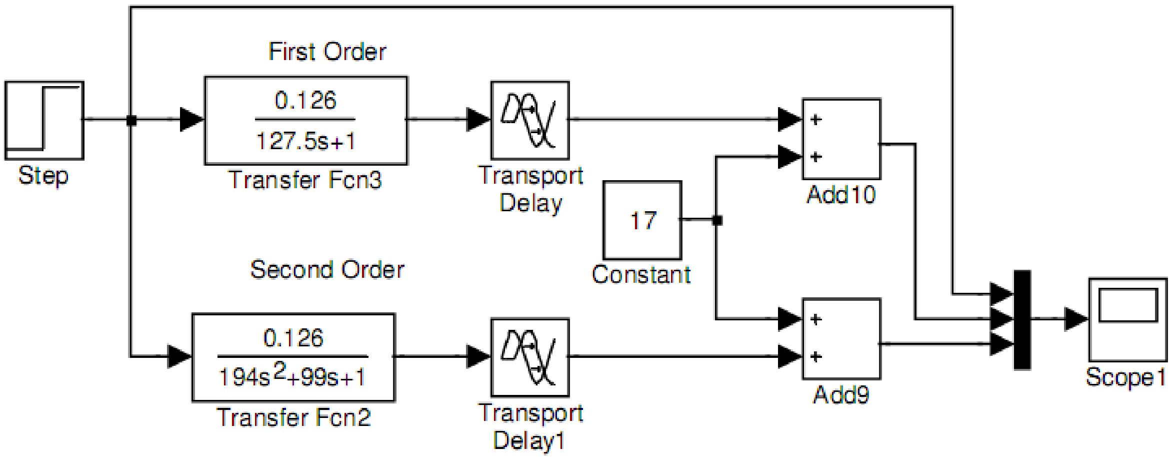

Figure 3.13 : *SIMULINK* model for the open loop step test comparing the FOPDT and the SOPDT

The simulation is basically a PRC implemented using *SIMULINK* where the initial temperature is at 17°C as this is a typical temperature found in the laboratory. The result of the simulation can be seen in Figure 3.14.

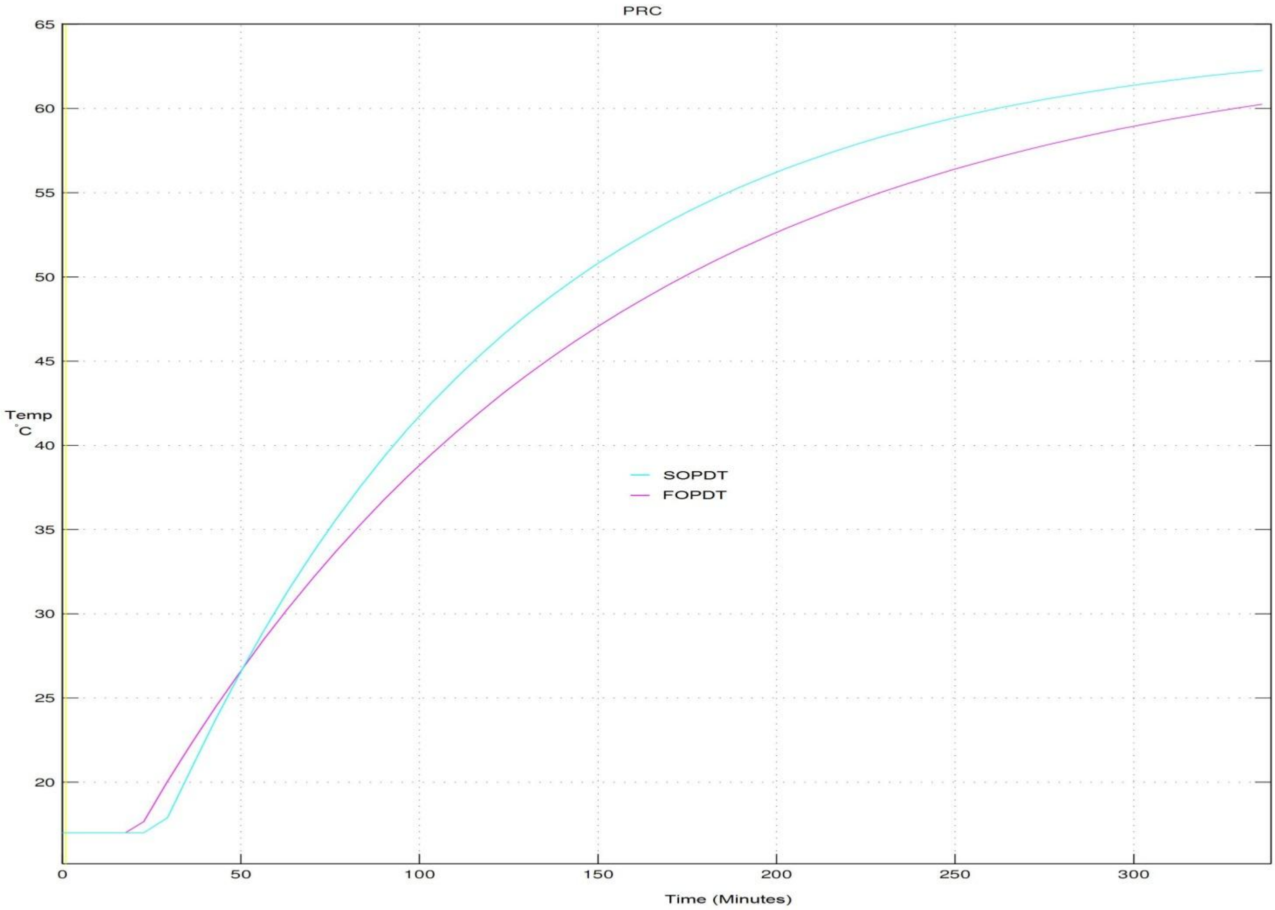

Figure 3.14 : FOPDT vs. SOPDT step test

The next step is to compare the Step Test (or PRC) from *SIMULINK* with that of the actual process to verify the accuracy of the models. To do this the above image is transposed onto the PRC in Figure 3.3 to give a direct comparison, this can be seen in Figure 3.15.

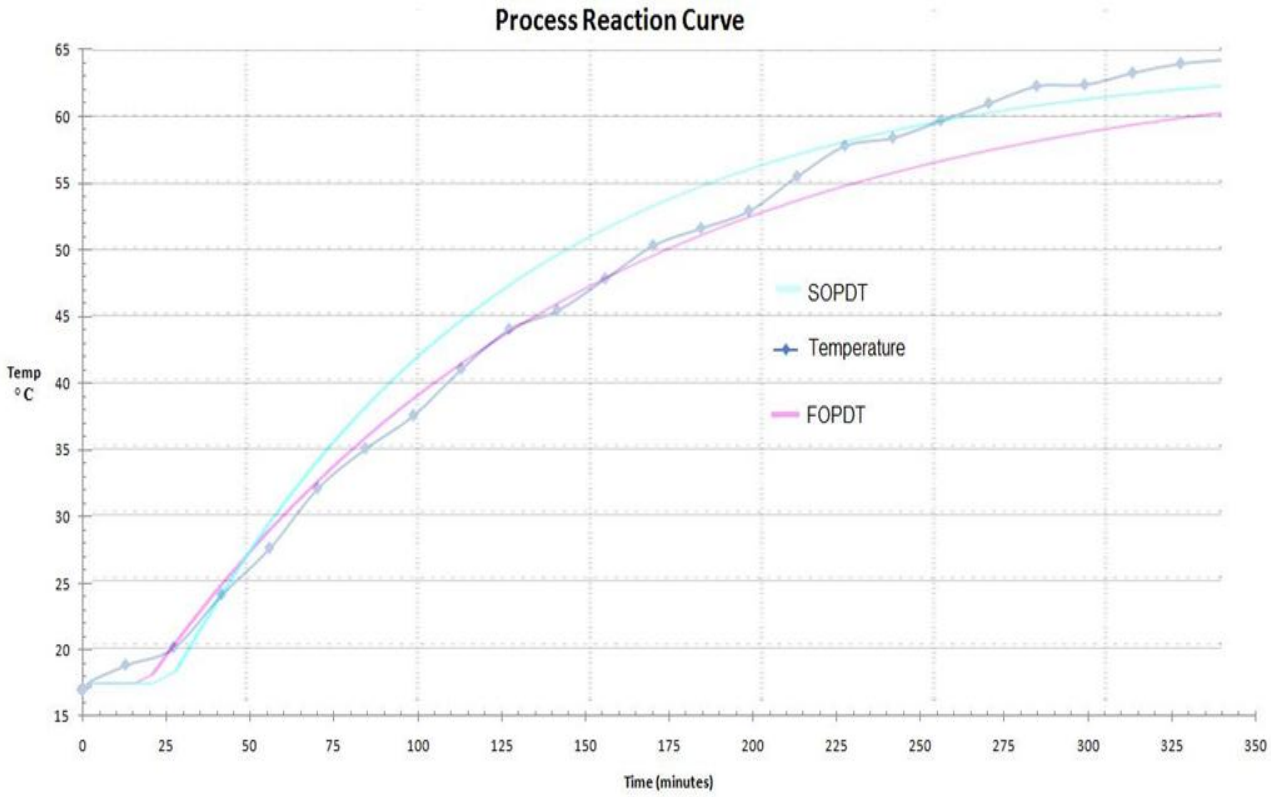

Figure 3.15 : Comparison of the FOPDT and SOPDT models with the PRC for the actual system

The FOPDT approximation emerges to be more accurate than that of the SOPDT as depicted in Figure 3.15. The difference between the SOPDT model and the actual PRC is very large and it appears that the two time constants are not large enough, while the FOPDT appears to be moderately accurate although there are still some discrepancies that could be attributed to both noise in the measured output and also to inaccuracies in the model.

The response of the integrating model that is also used to identify controller parameters can be seen in Figure 3.16.

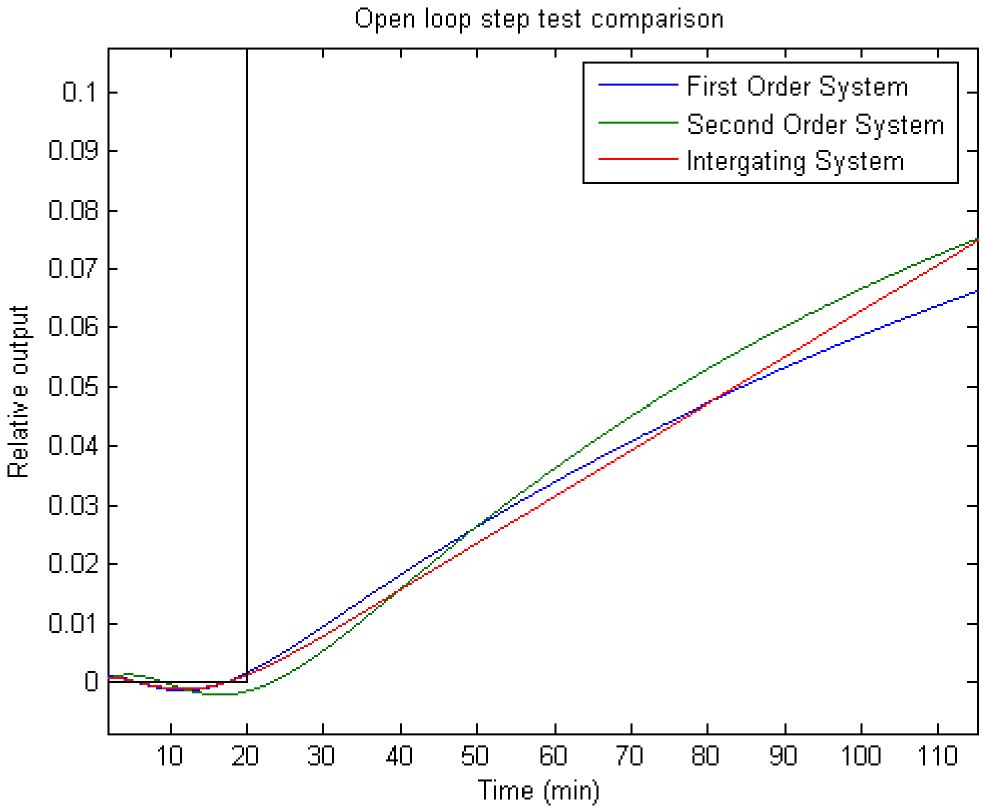

Figure 3.16 : Open Loop step test comparison

The Integrating model is a very accurate approximation for the initial rise of the system and there is similarity between all three of the models.

3.5 Performance Criteria

There a number of performance indicators that can be used to determine the best controller, the criteria used here are:

1. Percentage Overshoot (%OS)
2. Settling Time ($t_s 2\%$)
3. Rise Time (10-90%)
4. Steady State Error (SSE)
5. Gain Margin (GM)
6. Phase Margin (PM)

The criteria for the temperature control regulation of this system is to have a maximum overshoot of 4% with zero steady state error, the 10-90% Rise time should be less than 15 minutes and the 2% settling time of under 50 minutes.

Also the Gain Margin should be between 1.7 and 4 with the Phase Margin required to be between 45° and 65°. These values are requested as they represent a system which is stable and within a safe region of stability. The aim is to ensure that the system is tuned effectively but also to be stable.

The first and most important requirement is to have a maximum overshoot of 4%; this is defined in Figure 3.17 where the largest deviation is used to calculate the percentage overshoot.

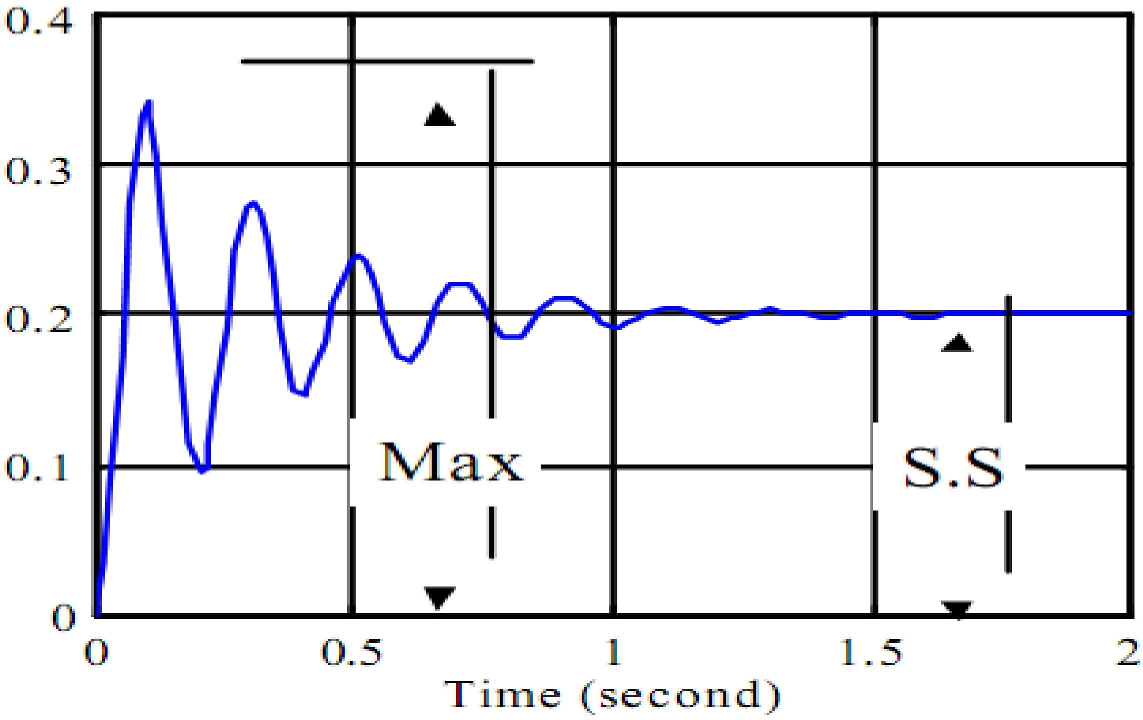

Figure 3.17 : Percentage Overshoot

There is a simple formula that is used to convert the overshoot to a percentage and it is given here as:

$$\%OS = \left(\frac{Max - S.S}{S.S}\right) \times 100\% \tag{3.7}$$

This enables different input step sizes to be directly compared to each other and can give a good indication to the overall performance of the controller.

The next indicator of the performance of the system used here is the settling time which is defined in Figure 3.18.

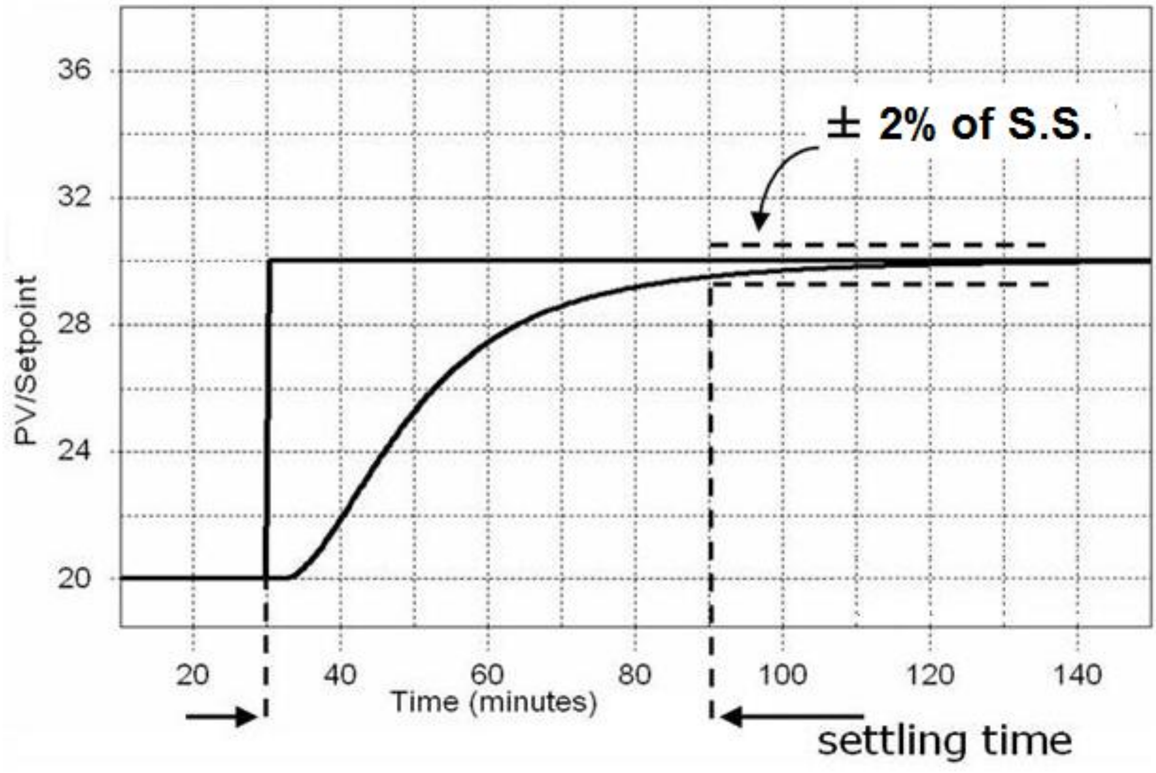

Figure 3.18 : 2% Settling Time

The Settling time gives a good indication as to how quickly the system meets the required value. The 2% envelope is a typical value that is used and this allows for slight variation around the setpoint.

Also is it important that there is zero steady state (S-S) error, this is where the output tracks the input exactly as required. This specification can be met by using a controller that has an integral term. The $\frac{1}{s}$ term will eventually bring all systems to the required steady state value as long as the system is stable.

The next specification is the Rise Time where again typical values are used i.e. 10-90% rise time, this is depicted in Figure 3.19.

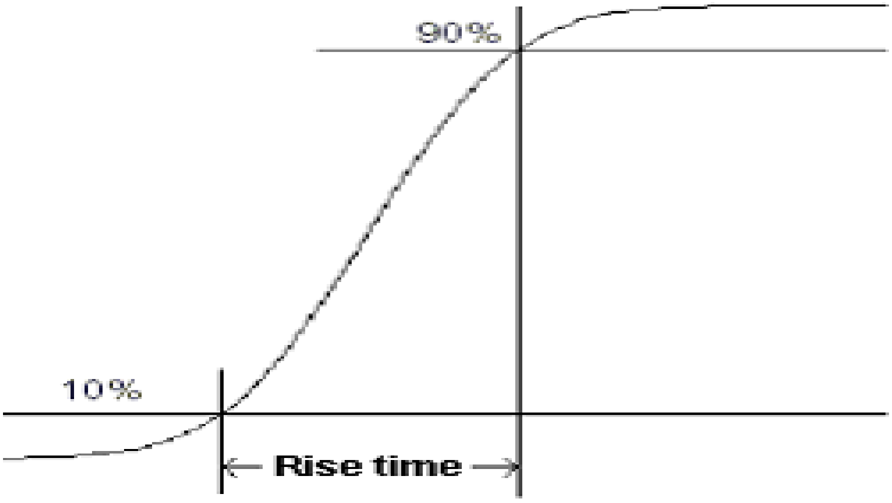

Figure 3.19 : Rise Time

The rise time gives a good indication as to how quickly the system can respond to a setpoint change.

The Phase Margin and Gain Margin are determined as defined in Figure 3.20.

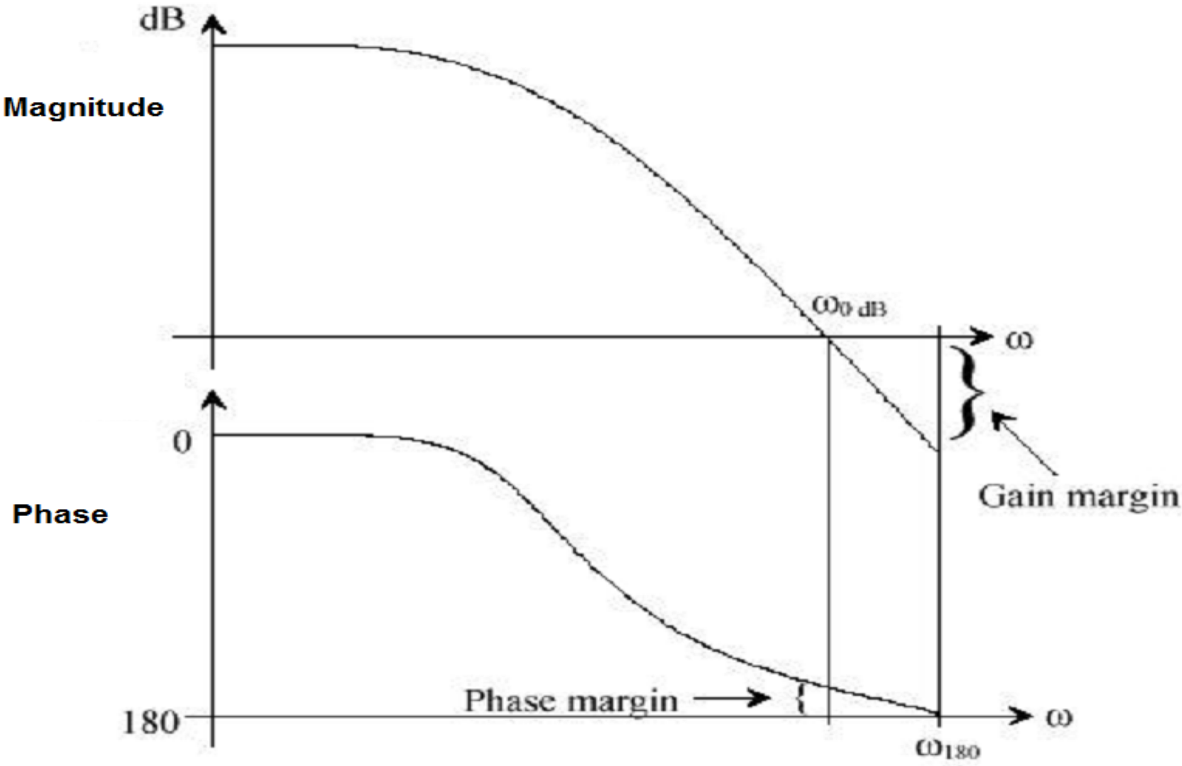

Figure 3.20 : Bode Plot to determine PM and GM

The Bode plot shows the frequency response of the system, where there is a both a phase and a gain component. The Gain and Phase margin give an indication as to the stability of the system, where the Gain Margin is the Gain at a phase of -180° while the Phase margin is the Phase angle at the point where the Gain crosses the 0dB point.

3.6 Summary

In this chapter the methodology of both the experimental and simulated work is defined with an explanation of the crucial areas. The different methods used to define a model of the system are described and are implemented giving two different transfer functions. There is a FOPDT and a SOPDT function that both describe the actual process. These are then used to obtain tuning values for the controller using a number of different methods as described here.

The method in which these controllers will be implemented on the PLC and how they will be displayed on the SCADA system are illustrated. The approach to building a SCADA system is expressed in a generic way that can be implemented on any process for any purpose. These are followed to ensure that the SCADA is as streamlined as possible and remains fully in control of the process.

The simulation software is also examined in this chapter to ensure that simulations undertaken give repeatable results and also to ensure that any problems or issues are dealt with. The method in which Matlab deals with time delays can cause problems but the approximation is accurate enough in this instance to prevent any errors.

As the performance of the controllers is under scrutiny there is a need to define the metrics in which the different systems and controllers can be compared. These are detailed in this section and the desired specifications are asserted.

4. Theoretical Background

4.1 Introduction

This section covers the necessary mathematical equations and derivations that are used in this project. The heat transfer calculations are included with the aim to identify the correlation between the model for the system and the actual system. The heat transfer equations derived can be used to give an indication as to the expected time to heat the product.

The implementation of the control strategy's on the PLC means that they need to be converted to their discrete counterpart. The difference equations to describe the controllers can then be converted to ladder logic or statement list which can be implemented on the PLC.

4.2 Heat Transfer

Heat transfer is one of the most important industrial processes. Throughout any industrial facility, heat must be added, removed, or moved from one process stream to another. There are three basic types of heat transfer which are conduction, convection, and radiation. The two most common forms encountered in the chemical processing industry are conduction and convection.

Any overall energy balance starts with the following equation:

$$Q = MC_p\Delta T\left(Kg.\frac{Joule}{g.{}^\circ C}.{}^\circ C\right) \tag{4.1}$$

Where:

Q = heat transferred in thermal unit per time (Btu/h or kW)

M = mass flow rate

T = temperature

Cp = heat capacity or specific heat of fluid

When we fill in the values we find that:

$$Q = (5 \times 4.186 \times 30) \times 1000$$

$$Q = 627.9\ kJ$$

In theory, the heat given up by the hot fluid is never exactly equal to the heat gained by the cold fluid due to environmental heat losses. In practice, however, they are generally assumed to be equal to simplify the calculations involved. Any environmental losses are generally minimized with insulation of equipment and piping. (22)

The time required to heat the vessel to the required temperature can be derived using first principles. (23) These can be used to verify that the system is performing as expected.

The first step is to translate the fact that the heat lost by the utility liquid in the jacket is equally to the heat gain by the content of the vessel. This can be seen in Equation (4.2) where W is the flow rate of the liquid in the jacket, U is the Overall Heat transfer coefficient, A is the area, T_1 is the Jacket inlet temperature and T_2 is the jacket outlet temperature.

$$Q = WC_p(T_2 - T_1) = UA\left[\frac{T_2 - T_1}{\ln\left(\frac{T(t) - T_2}{T(t) - T_1}\right)}\right] \tag{4.2}$$

The next step is to solve for the unknown jacket outlet temperature T_2, which can be seen in Equation (4.3).

$$T_2 = T(t) + exp\left(\frac{-UA}{WC_p}\right)(T_1 - T(t)) \tag{4.3}$$

The next stage involves the rate of temperature change of the contents of the vessel and is given here as seen in Equation (4.4).

$$mC_p\frac{dT}{dt} = Q \tag{4.4}$$

The process temperature can be written as a function of time as seen in Equation (4.5) and this is found by substitution in Equation (4.2) and (4.3).

$$T(t) = (T_0 - T_1)exp\left\{\frac{WC\left(exp\left[\frac{-UA}{WC_p}\right] - 1\right)t}{mC_p}\right\} + T_1 \tag{4.5}$$

The final step is to re-arrange Equation (4.5) to give the time (t).

$$t = \frac{mC_p \ln\left(\frac{T_f - T_2}{T_0 - T_1}\right)}{WC_p\left(exp\left(\frac{-UA}{WC_p}\right) - 1\right)} \tag{4.6}$$

Where T_f is the final temperature and T_0 is the initial temperature.

When the corresponding values are identified and filled into the expression in Equation (4.6), the expected time for the contents to reach the required temperature is circa 30 minutes. However this is an estimation and is only used to give an indication as to the expected performance of the process. As these expressions are based on a number of approximations the theoretical analysis of this system will not be investigated further as this is not the focus of this project.

4.3 Discrete FOPDT Transfer Function

The transfer function for a FOPDT in given here as:

$$G_p(S) = \frac{k_p}{T_pS + 1}e^{-ST_d} \tag{4.7}$$

The FOPDT transfer function depicted in Equation (4.7) is the standard continuous-time transfer function where the parameters are the same as those previously defined.

A discrete equivalent of the continuous plant is needed to simulate the PLC based sampled-data system. In the PLC system the controller is discrete while the plant is continuous. The discrete-time equivalent of the system needs to be obtained to simulate the entire system as a discrete-time system.

The trapezoid rule can be used to approximate the continuous time term ‘s’ with a discrete time term ‘z’. Under trapezoid rule, the discrete-time system is stable if and only if the continuous-time system is stable. The use of this approximation introduces an additional time delay which is approximately equal to half the sample time (24). In this instance the effect will be negligible as the sampling time is much smaller than the time constant of the system so the additional time delay will be absorbed without any noticeable effect on the overall performance.

$$S \rightarrow \frac{2}{T_s}\frac{1 - Z^{-1}}{1 + Z^{-1}} \tag{4.8}$$

The Tustin Rule (also known as Trapezoidal Rule) is given in Equation (4.8), the first step is then to substitute for 'S' in the continuous-time transfer function with the transform in Equation (4.8). This can be seen in Equation (4.10) where the transfer function is expressed as the Output Y (Z) divided by the Input U (Z).

$$G_p(Z) = (1 - Z^{-1})\mathbb{Z}\left\{\frac{G_p(S)}{S}\right\} = (1 - Z^{-1})\mathbb{Z}\left\{\frac{k_p}{T_p S^2 + S}\right\} \tag{4.9}$$

The expression in Equation (4.9) is the Zero Order hold method to convert the continuous time expression into the discrete time equivalent.

$$G_p(Z) = \frac{Y(Z)}{U(Z)} = \frac{k_p - k_p Z^{-1}}{\left(\frac{2T_p - 2T_p Z^{-1}}{T_s + T_s Z^{-1}}\right)^2 + \frac{2T_p - 2T_p Z^{-1}}{T_s + T_s Z^{-1}}} \tag{4.10}$$

Then the equation can be rearranged by cross multiplication to give Equation (4.11) :

$$Y(Z)\left[\left(\frac{2T_p - 2T_p Z^{-1}}{T_s + T_s Z^{-1}}\right)^2 + \frac{2T_p - 2T_p Z^{-1}}{T_s + T_s Z^{-1}}\right] = U(Z)\left[k_p - k_p Z^{-1}\right] \tag{4.11}$$

$$Y(Z)\left[\frac{4T_p - 8T_p Z^{-1} + 4T_p Z^{-2}}{{T_s}^2 + 2{T_s}^2 Z^{-1} + {T_s}^2 Z^{-2}} + \frac{2T_p - 2T_p Z^{-1}}{T_s + T_s Z^{-1}}\right] = U(Z)\left[k_p - k_p Z^{-1}\right] \tag{4.12}$$

$$Y(Z)\left[\frac{4T_p - 8T_p Z^{-1} + 4T_p Z^{-2} + \left(2T_p - 2T_p Z^{-1}\right)(T_s + T_s Z^{-1})}{{T_s}^2 + 2{T_s}^2 Z^{-1} + {T_s}^2 Z^{-2}}\right] = U(Z)\left[k_p - k_p Z^{-1}\right] \tag{4.13}$$

$$Y(Z)\left[\frac{4T_p - 8T_pZ^{-1} + 4T_pZ^{-2} + 2T_pT_s - 2T_pT_sZ^{-2}}{{T_s}^2 + 2{T_s}^2Z^{-1} + {T_s}^2Z^{-2}}\right] = U(Z)\left[k_p - k_pZ^{-1}\right] \tag{4.14}$$

$$\begin{aligned} &Y(Z)\left[4T_p - 8T_pZ^{-1} + 4T_pZ^{-2} + 2T_pT_s - 2T_pT_sZ^{-2}\right] \\ &\qquad = U(Z)\left[\left(k_p - k_pZ^{-1}\right)({T_s}^2 + 2{T_s}^2Z^{-1} + {T_s}^2Z^{-2})\right] \end{aligned} \tag{4.15}$$

$$\begin{aligned} &Y(Z)\left[4T_p - 8T_pZ^{-1} + 4T_pZ^{-2} + 2T_pT_s - 2T_pT_sZ^{-2}\right] \\ &\qquad = U(Z)\left[k_p{T_s}^2 + 2k_p{T_s}^2Z^{-1} + k_p{T_s}^2Z^{-2} - k_p{T_s}^2Z^{-1} \right.\\ &\qquad \left. - 2k_p{T_s}^2Z^{-2} - k_p{T_s}^2Z^{-3}\right] \end{aligned} \tag{4.16}$$

$$\begin{aligned} &Y(Z)\left[4T_p + 2T_pT_s - 8T_pZ^{-1} + (4T_p - 2T_pT_s)Z^{-2}\right] \\ &\qquad = U(Z)\left[k_p{T_s}^2 + \left(k_p{T_s}^2 + 2k_p{T_s}^2\right)Z^{-1} + (k_p{T_s}^2 \right.\\ &\qquad \left. - 2k_p{T_s}^2)Z^{-2} - k_p{T_s}^2Z^{-3}\right] \end{aligned} \tag{4.17}$$

$$\begin{aligned} &\left[\left(4T_p + 2T_pT_s\right)Y(i) - 8T_pY(i-1) + (4T_p - 2T_pT_s)Y(i-2)\right] \\ &\qquad = \left[k_p{T_s}^2U(i) + \left(k_p{T_s}^2 + 2k_p{T_s}^2\right)U(i-1) + (k_p{T_s}^2 \right.\\ &\qquad \left. - 2k_p{T_s}^2)U(i-2) - k_p{T_s}^2U(i-3)\right] \end{aligned} \tag{4.18}$$

$$\begin{aligned} &\left[\left(4T_p + 2T_pT_s\right)Y(i)\right] \\ &\qquad = 8T_pY(i-1) - \left(4T_p - 2T_pT_s\right)Y(i-2) + k_p{T_s}^2U(i) \\ &\qquad + \left(k_p{T_s}^2 + 2k_p{T_s}^2\right)U(i-1) + (k_p{T_s}^2 - 2k_p{T_s}^2)U(i-2) \\ &\qquad - k_p{T_s}^2U(i-3) \end{aligned} \tag{4.19}$$

$$
\begin{aligned}
Y(i) = \frac{1}{\left(4T_p + 2T_pT_s\right)} \Big[& 8T_pY(i-1) - \left(4T_p - 2T_pT_s\right)Y(i-2) + k_pT_s{}^2U(i) \\
& + \left(k_pT_s{}^2 + 2k_pT_s{}^2\right)U(i-1) + (k_pT_s{}^2 - 2k_pT_s{}^2)U(i-2) \\
& - k_pT_s{}^2U(i-3) \Big]
\end{aligned}
\tag{4.20}
$$

$$
\begin{aligned}
Y(i) = \frac{1}{\left(4T_p + 2T_pT_s\right)} \Big[& 8T_pY(i-1) - \left(4T_p - 2T_pT_s\right)Y(i-2) + k_pT_s{}^2U(i) \\
& + \left(3k_pT_s{}^2\right)U(i-1) + (k_pT_s{}^2)U(i-2) - k_pT_s{}^2U(i-3) \Big]
\end{aligned}
\tag{4.21}
$$

So

$$
\begin{aligned}
Y(i) \ = a\big\{ & [Y(i-1)](b) - [Y(i-2)](c) \\
& - (d)\big(U(i) + 3U(i-1) + U(i-2) - U(i-3)\big)\big\}
\end{aligned}
\tag{4.22}
$$

Where:

$$
\begin{aligned}
a &= \frac{1}{\left(4T_p \ + 2T_pT_s\right)} \\
b &= 8T_p \\
c &= 4T_p - 2T_pT_s \\
d &= k_pT_s{}^2
\end{aligned}
$$

The expression shown in Equation (4.22) is the output of the discrete First Order System only. The delay term is given in Equation (4.23). As can be seen the delay component is dealt with in the same manner as the First Order system. The difference equation seen in Equation (4.25) is used to implement the delay term in the discrete implementation using the PLC.

$$
G_p(S) = \ e^{-ST_d} \tag{4.23}
$$

$$
G_p(Z) = (1 - Z^{-1})\mathbb{Z}\left\{\frac{G_p(S)}{S}\right\} = (1 - Z^{-1})\mathbb{Z}\left\{\frac{e^{-sT_d}}{S}\right\} \tag{4.24}
$$

$$Y(i) = U\left(i - \frac{T_d}{T_s}\right) - U\left(i - 1 - \frac{T_d}{T_s}\right) \tag{4.25}$$

The expressions given in Equation (4.22) and in Equation (4.25) are used to implement the Discrete Smith Predictor.

4.4 Discrete PI Controller

The Proportional Integral controller that is implemented in the PLC can be seen in the block diagram in Figure 4.1.

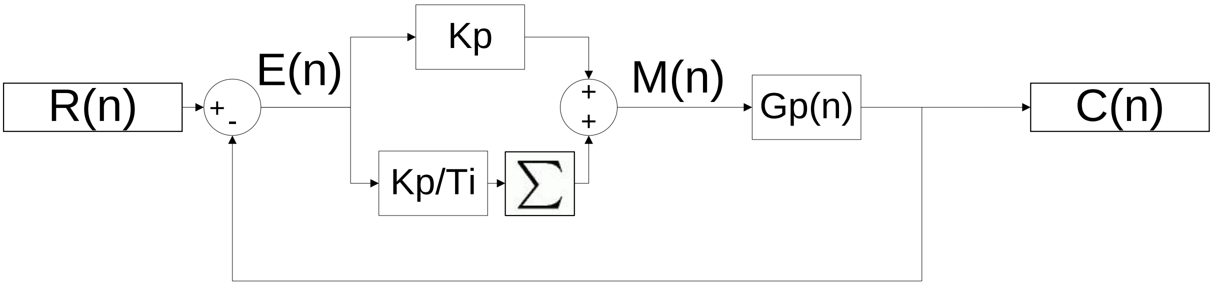

Figure 4.1 : Discrete PI controller block diagram

The Continuous time domain PI controller can be approximated by a discrete time equivalent algorithm which can be easily implemented in a PLC as it only contains basic mathematic operations.

The output from the PI controller (M (n)) is a combination of the Proportional segment and the Integral term as illustrated below:

$$M(n) = P(n) + I(n) \tag{4.26}$$

The error signal is generated by subtracting the process feedback from the Setpoint as seen in Equation (4.27).

$$E(n) = R(n) - C(n) \tag{4.27}$$

The proportional component is calculated by multiplying the error by the proportional gain as depicted below:

$$P(n) = k_p * E(n) \tag{4.28}$$

The integral component of the controller is the addition of the previous integral term and the term containing the integral time, the sampling time, the proportional gain and the error term as illustrated below in Equation (4.29):

$$I(n) = I(n-1) + \frac{E(n) * k_p * \Delta t}{\tau_i} \tag{4.29}$$

Then the total controller output can be rewritten in terms of its individual components as can be seen in Equation (4.30) :

$$M(n) = k_p * E(n) + \left[I(n-1) + \frac{I(n) * k_p * \Delta t}{\tau_i} \right] \tag{4.30}$$

This is calculated at each 1 second interval, which gives an adequate resolution as the time constant for the process is in the order of minutes. The overall effect of using a time sampled approach should not have any negative impact on the system as long as the time interval is small enough to give good resolution.

The PID controller that is implemented on the PLC also has an anti-windup constituent to overcome the problem of integrator windup. The problem occurs when for example a valve is 100% open and then the controller tries to open more but this is not physically possible. So the integration value continues to rise even thought the valve is already fully open. The problem is that when the controller value falls to say 50% but the integration value is above where it should be and thus has to integrate down past the value where it should be at when the valve was at 100%. This causes a sluggish response to system as the integration value is too high. (25)

So to overcome this problem an anti-windup component is included in the control strategy, this is where the integration is stopped when the controller output reaches either 100% or 0% and thus the integral term cannot windup to an unrealistic value.

4.5 International Society of Automation (ISA) standard

The introduction of the S88 batch standards came about as a way to improve the automation industry and to create a single template that each automation project could be designed around.

The standard helps to modularize processes and provide frameworks for recipes; how much is automated may depend on the type of process you are trying to control and does not have any effect on the s88 approach.

“S88 is a generic term for an international standard relating to batch systems, IEC61512. It was begun as part of the ISA’s standardization activity started in 1988”

The introduction of S88 came about as a solution to a number of problems such as:

1. There was no accepted form of batch control
2. Customers found it difficult to explain to vendors what they would like to do with batch processes
3. Batch control systems required considerable specialised labour
4. Combining control systems from several manufacturers into one system was very difficult

This lead to numerous benefits including a cost reduction due to standardisation and recyclable code, easier technology and integration of different suppliers was made simpler.

“One of the goals of the S88 batch system design standardization is for anyone to able to design a batch system within the same framework by adopting this modelling concept” (26)

4.6 Summary

In this chapter the mathematical equations that define the operating principles have been examined, leading to an approximate description of the process. These expressions are used to give an indication as to the operating principles which are used to help develop efficient and effective control strategies.

The discrete expressions for the PI controller are used to implement the controller on the PLC. There are also a number of components to the control strategy that are not included here but are accounted for in the PLC such as the Anti-Integral windup.

The discrete representation of the FOPDT model is used in the implementation of the Smith Predictor on the PLC and is broken into two parts. The first is the description with the delay and the second is the delay. This is then implemented as described in Figure 2.4 where the ideal smith predictor is shown. The FOPDT model is used as it is the simplest and offers a good representation of the actual process.

5. Results

5.1 Introduction

This section covers the results found from the laboratory testing of the system and from the Matlab simulations. As previously stated there are 2 models for the system and the initial aim here is to identify which one is more accurate and then decide if its accuracy is close enough to allow the simulation software be used to experiment to determine the best possible controller algorithm and then the parameters that can be used to achieve the best performance.

The initial heating stage is used to obtain the response of the system with the relevant controller; this stage involves a step input to the setpoint. For the purpose of a consistent analysis a setpoint of 50 °C is used throughout the experimental stage. The performance of the different controllers at this setpoint gives a valid indication as to the overall performance of the controller.

The performance of each controller at maintaining the required setpoint is relatively consistent between each controller and in this instance does not require further investigation. There are slight differences but the majority of controllers used here have good steady state tracking.

The disturbance rejection of the system gives an indication as to the response to the system in the event of an unexpected change in the operating conditions and is also an important aspect in the analysis of the performance of the controller. This can be implemented in the simulations by using the step response of first order system as an additional input into the process or plant.

However in the actual bench scale process the effect of any disturbance is minimised by the high surface area to volume ratio and thus any small disturbances are easily absorbed and have no adverse effect on the performance of the controller. The only disturbance is an unexpected change in ambient temperature and this has a slow response. The main difference between each controller at this scale in this model is the initial heating phase and hence receives the most attention.

5.2 System Performance

The first step in the testing stage was to determine a FOPDT model (namely **model 1** which is $G_p(s) = \frac{0.126e^{-20s}}{127.5s+1}$) from the Process Reaction Curve. The initial controller used here is a Proportional Integral (PI) controller with the parameters chosen using the Zeigler-Nichols tuning rules. The controller is in the form as seen in Equation (5.1).

$$G_c(s) = k_p\left[1 + \frac{1}{\tau_i s}\right] \qquad (5.1)$$

Where k_p is the Proportional Gain and τ_i is the Integral Time

There are two models that can be used to describe the physical process; the first model is a FOPDT of the form $G_p(s) = \frac{0.126e^{-20s}}{127.5s+1}$ which will be referred to as **Model 1.** While the second model is of the SOPDT type namely $G_p(s) = \frac{0.126e^{-25s}}{194s^2+99s+1}$ and will be referred to as **Model 2.**

The Zeigler-Nichols rules suggests a gain value of $k_p = 45.5$ and an integral time of $\tau_i = 66.6$ minutes, so to implement these on the PLC the integral time has to be converted to seconds which gives a value of τ_i=3996 seconds. The integral time can also be referred to as the integral gain $k_i = \frac{1}{\tau_i}$. The initial temperature of the reactor is room temperature which is not under control so there will be very slight variances between runs as the room heats up or cools down.

5.2.1 PI Control

The *SIMULINK* model that is used to simulate the PI controller can be seen in Figure 5.1.

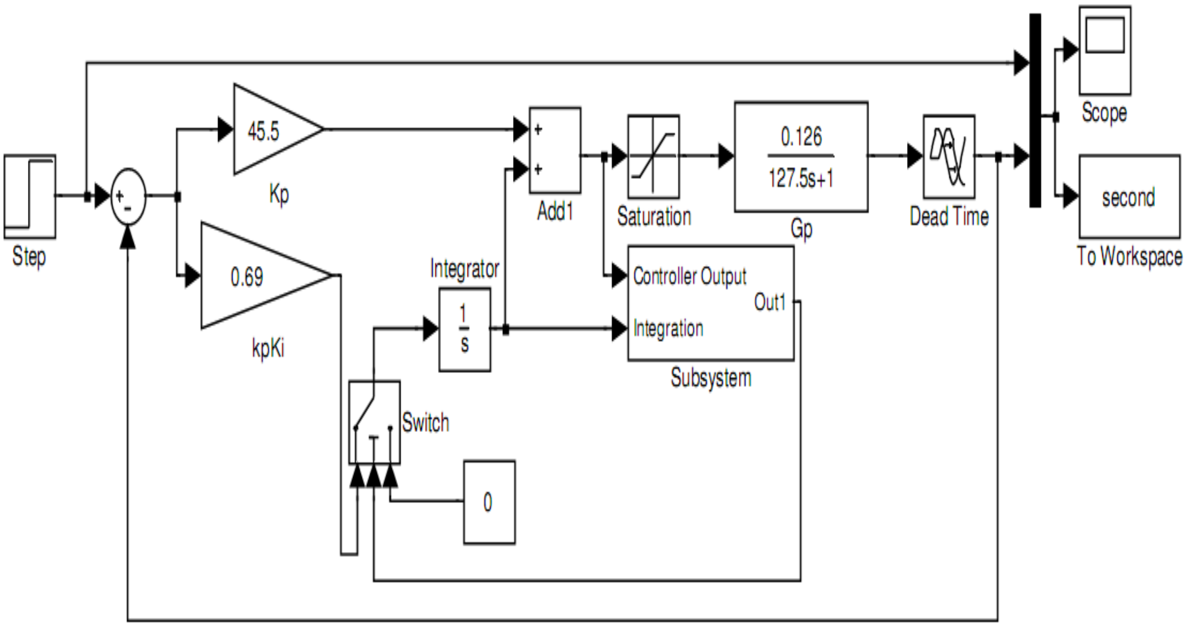

Figure 5.1 : *SIMULINK* model of PI controller

This controller is in the Ideal PID format but with the D term at zero. Also incorporated in the controller is the Integral Anti-Windup which prevents the integral term from winding up when the controlling element saturates. The model is designed to match the actual system as closely as possible and so the saturation block matches the maximum output in the actual system.

The Anti-Windup implementation as seen in Figure 5.2 is identical to the method used in the PLC code. The Integral term is only allowed to accumulate when the controller is below the maximum output. When the controller output is greater than the maximum allowed the integral term does not increment and stays at its previous value.

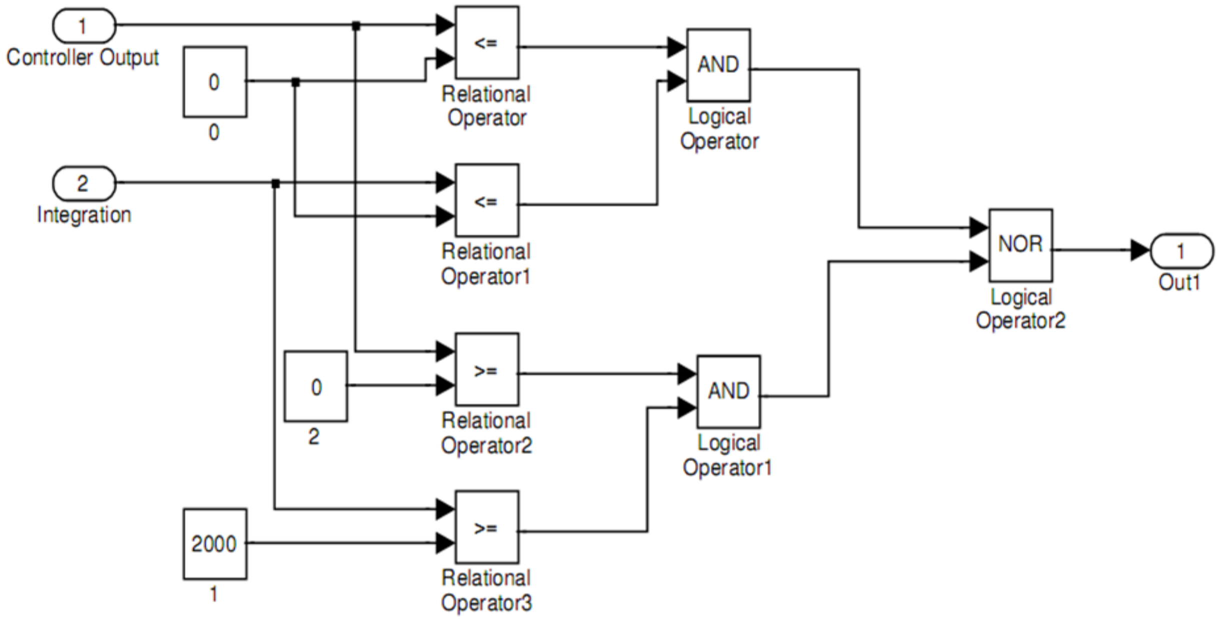

Figure 5.2 : Integral Anti-Windup element

There a number of different methods to prevent Integral Windup such as the parallel or series form. However this method was chosen as this matches the method used in the actual process. Also this method requires no additional tuning while other methods do.

The result from the heating stage of the process can be seen in Figure 5.3 where the Setpoint is 50°C and the controller is implemented using the Zeigler-Nichols rules (applied to **Model 1**).

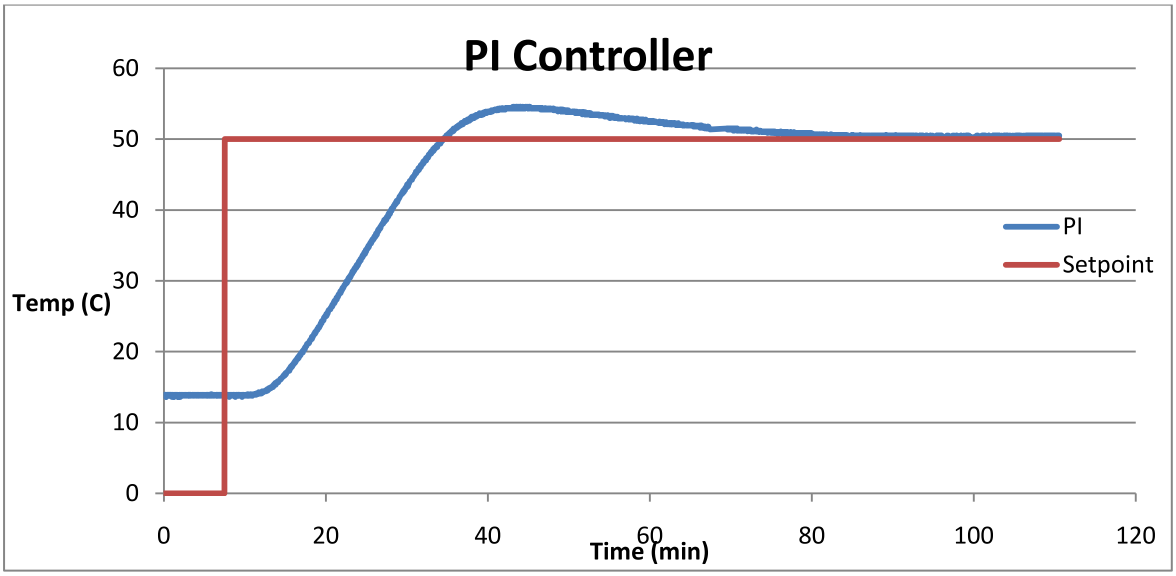

Figure 5.3 : Step Test on the Actual System

As expected there is a delay between the change in the setpoint and a change in the temperature of the product. The delay is an inherent property of the system and thus even the perfect controller cannot remove this delay. The overshoot in the response is quite large however the time to return to the setpoint is relatively quick and there is no ringing response which is desirable.

The simulation of **Model 1** was completed using *SIMULINK* with the same controller and the response is shown in Figure 5.4.

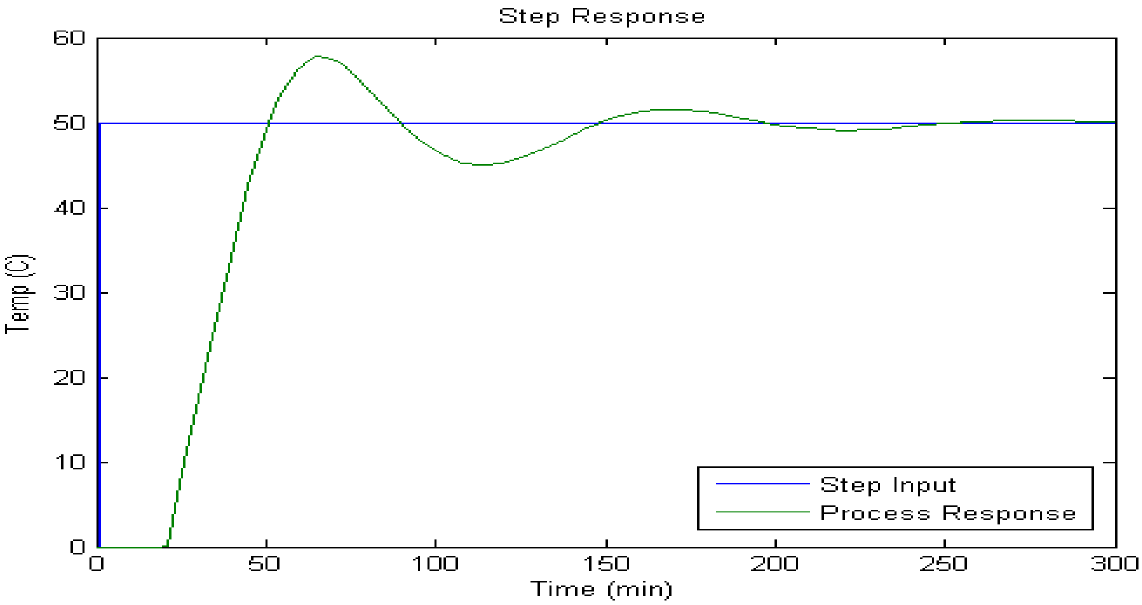

Figure 5.4 : Step Response obtained from *SIMULINK* simulation

As can be seen in Figure 5.4 the response has an overshoot and ringing response. The Integral of the Absolute Error (IAE) is 2145 and the system is stable. This is quite different to the response shown by the actual system. The settling time is extensive due to the oscillating response, while the initial overshoot is quite large.

The next step is to simulate **Model 2** using the same controller as above and the result is shown in Figure 5.5. As can be seen the systems response contains a substantial overshoot and is very oscillatory. The system is not stable and this is further depicted in the bode plot of the system.

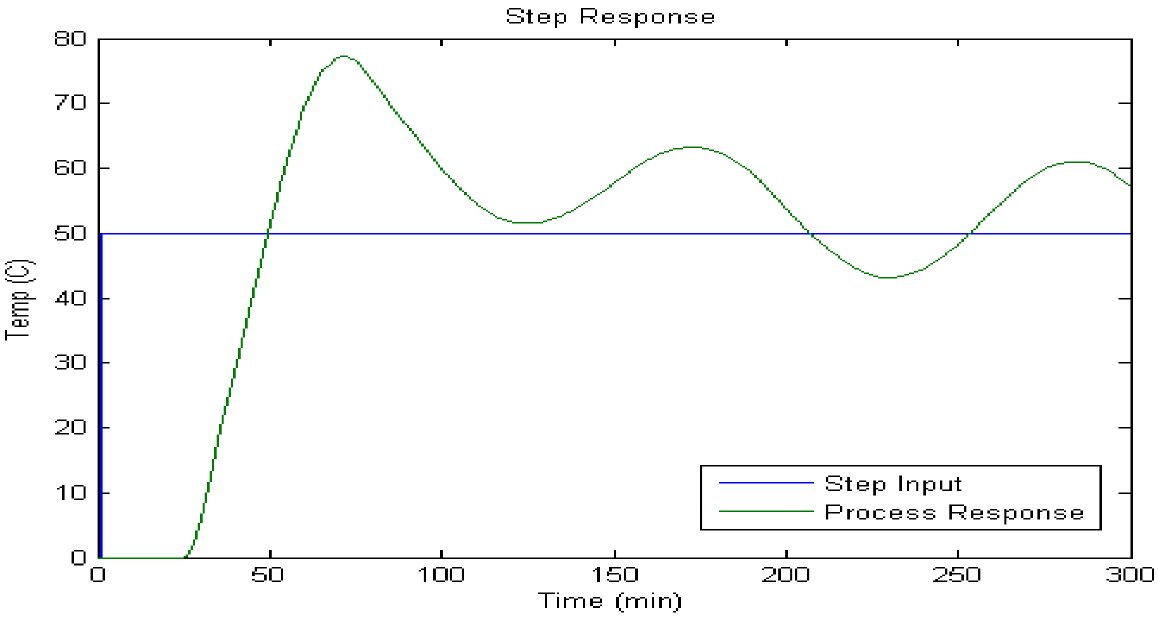

Figure 5.5 : Step Response obtained from Matlab simulation

Thus there are no criteria met using this controller with regards to **Model 2.**

The Bode plot in Figure 5.6 shows the frequency response of the system where **Model 1** is depicted as Go1st and **Model 2** as Go2nd.

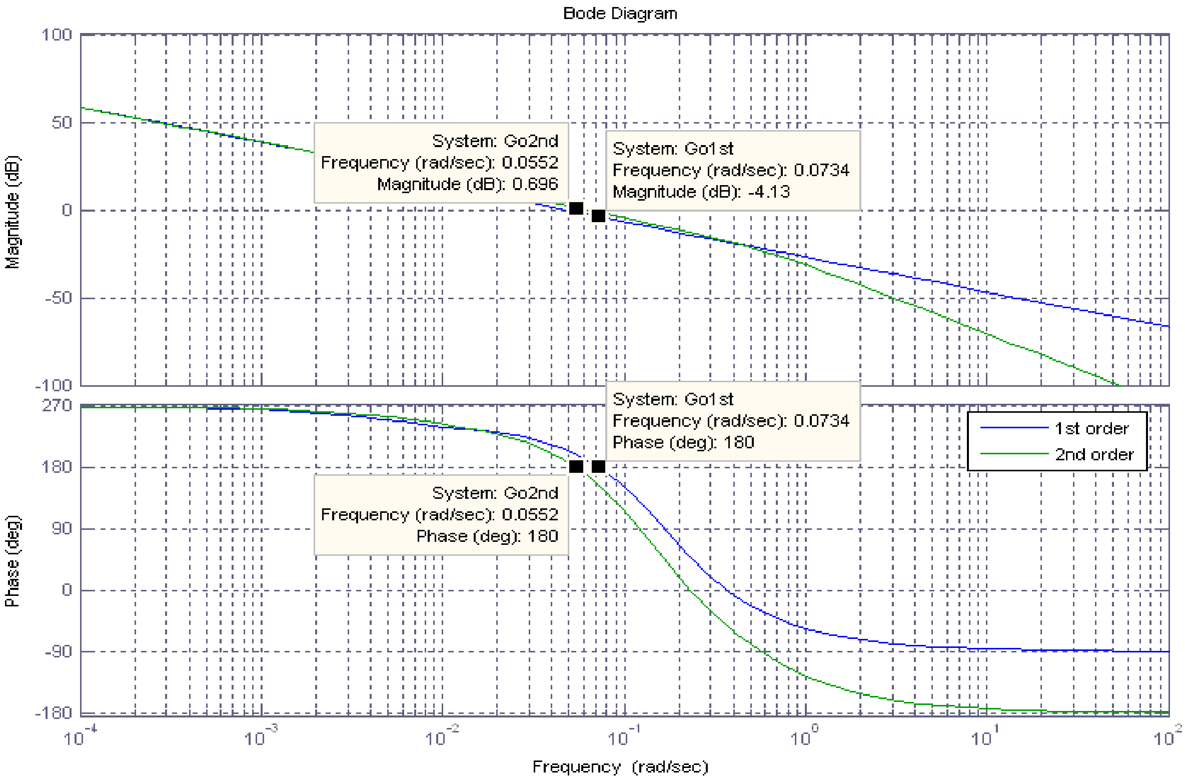

Figure 5.6 : Bode Plot

The gain margin for the two models is also depicted and the difference between the two models does not look to be that significant however the effect of this difference has a massive effect on the performance of the controller.

So from the viewpoint of the accuracy of the two models compared to the actual process it is clear that **Model 1** is more accurate than **Model 2.** This will be further investigated to determine the true accuracy of the process models.

Integrating Model tuning rules

The next step is to implement the tuning rules that can be inferred from the integrating model of the process. The PI controller that is suggested by the Integrating model has a gain $k_p = 57.2$ and an integral time of $\tau_i = 66.6\, minutes$. As this is calculated on a second basis by the PLC the integral time used is $\tau_i = 3996\, seconds$. The result from the experiment can be seen in Figure 5.7.

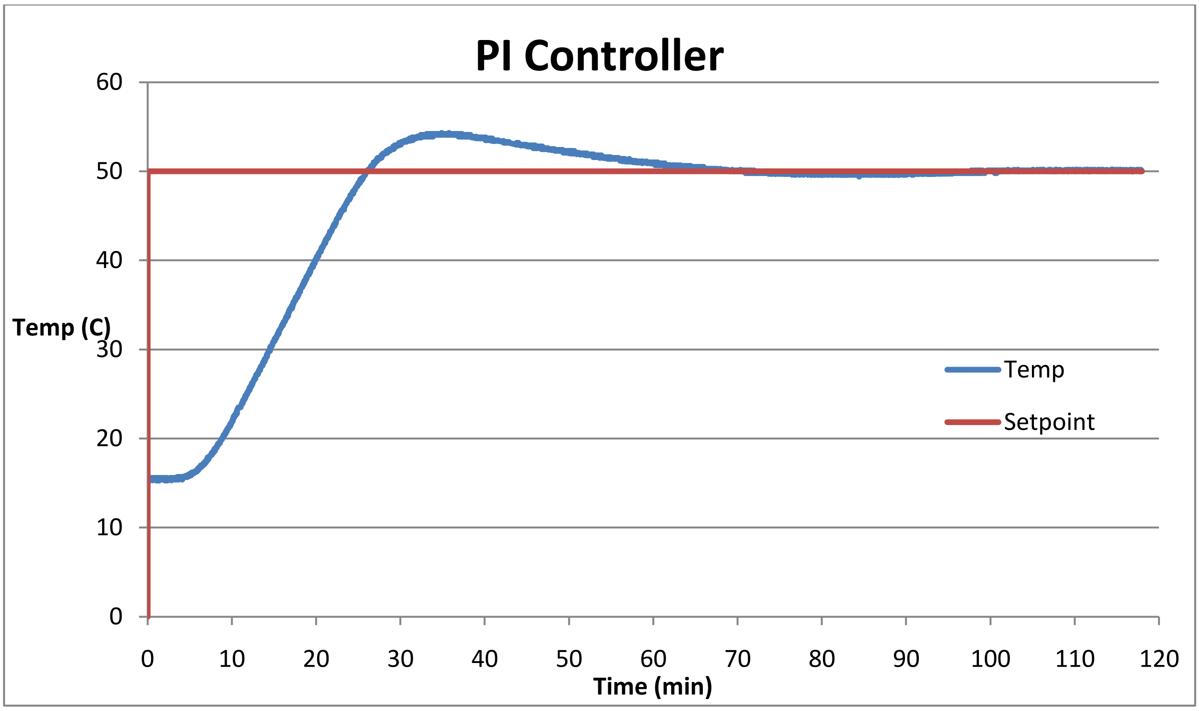

Figure 5.7 : Actual System response to a Setpoint Change

The response has an overshoot of 8.6% which is double the maximum acceptable value and is thus not satisfactory. However the return of the temperature to the setpoint after the initial overshoot is quite swift and this is desired. There is a slight oscillation but these are well within the allowed tolerances.

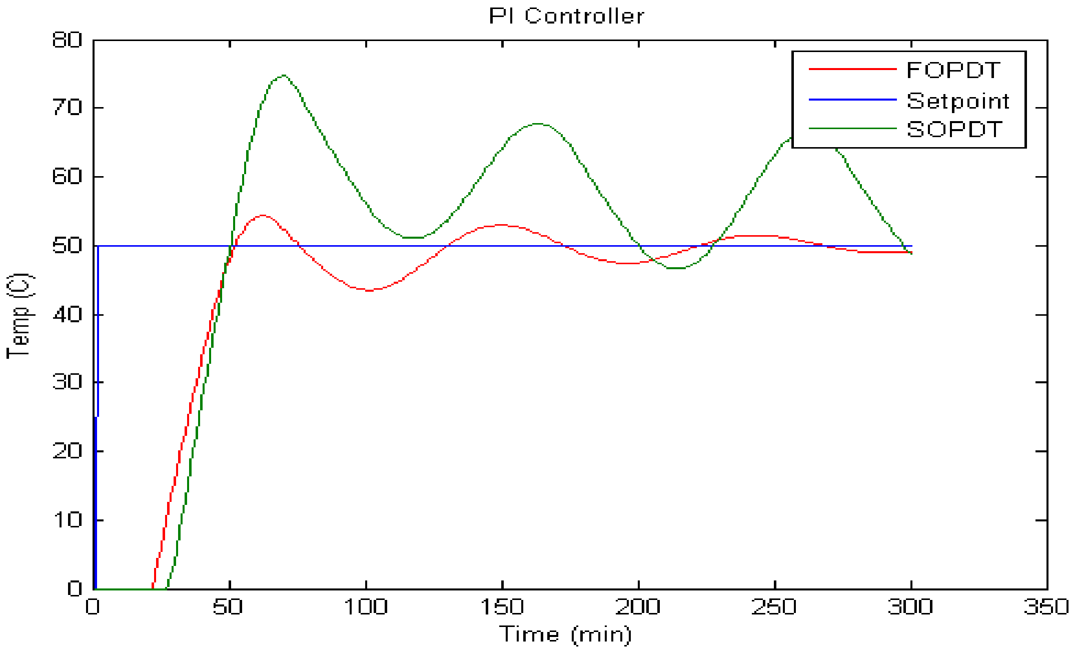

Figure 5.8 : Response to a Setpoint change as obtained from Matlab simulation

The controller used on the actual process above is then implemented on the two models using *SIMULINK*. The result from the simulation can be seen in the Figure 5.8 above where both **Model 1** (FODPT) and **Model 2** (SOPDT) are plotted**.** The response from both models is of an oscillatory nature with a considerable overshoot. The reaction of **Model 1** is much better than **Model 2** as can be clearly seen. Also **Model 2** appears to be unstable as it maintains a large oscillation. The initial response of **Model 2** is also slower than that of **Model 1.**

Again the Bode plots for the two process models are compared. The gain margin is of particular interest and is labelled in Figure 5.9.

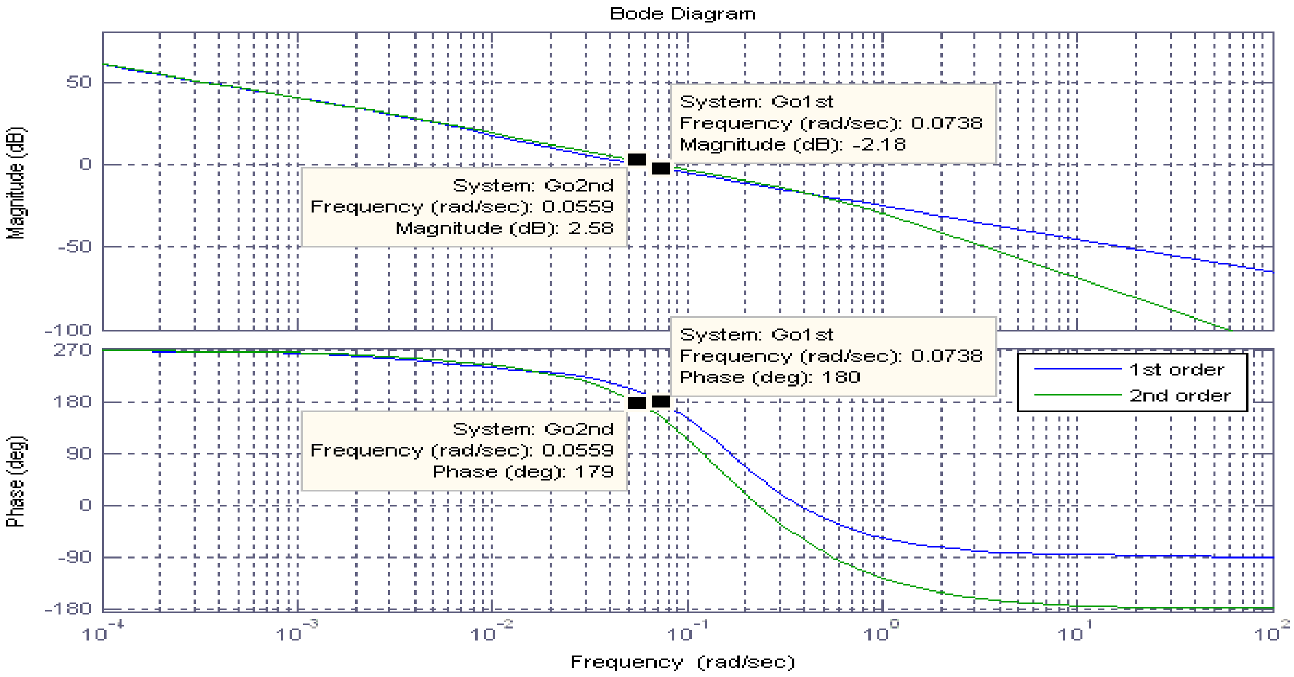

Figure 5.9 : Bode Plot

The result from this part of the experiment suggests that **Model 1** is the closest representation to the actual process. The simulation with **Model 1** appears to have more severe oscillations than that which occurs in the actual process. However it is worth noting that the Gain margin and Phase margin are from ideal and thus the controller would need further adjustment.

The next step in the testing phase is to use the controller parameters that are defined using the SOPDT tuning rules. The result from using the SOPDT defined controller on the actual system can be seen in Figure 5.10

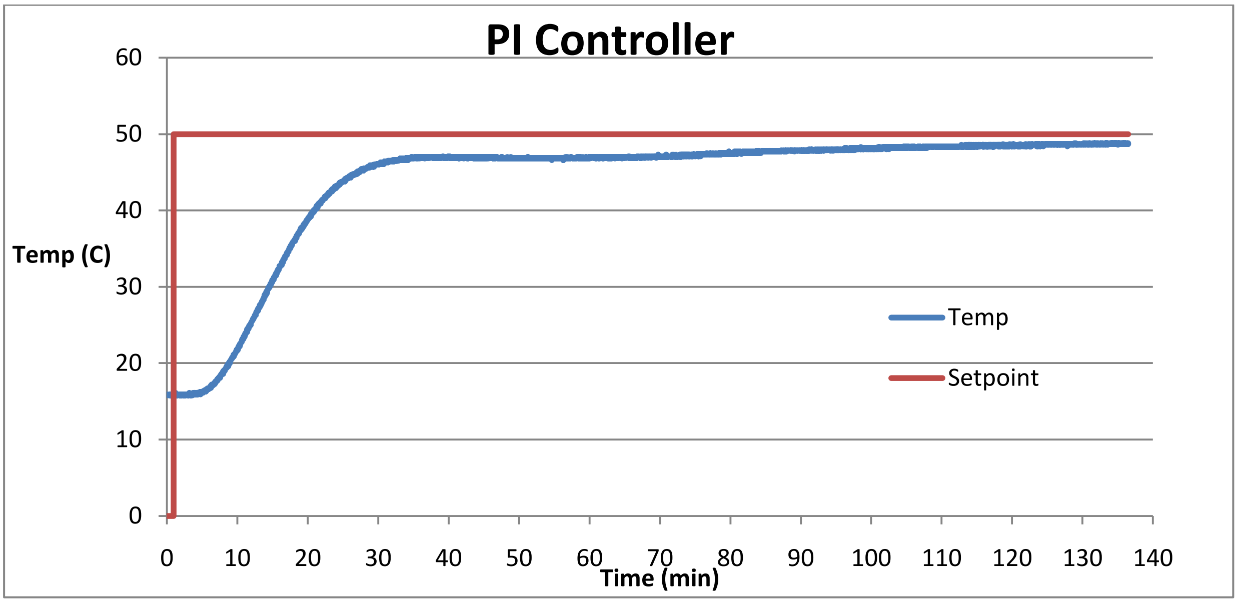

Figure 5.10 : Actual System response to a Setpoint Change

The performance of the controller in Figure 5.10 is quite sluggish however the rise time appears to be quite good and there is no overshoot. But the settling time in comparison to the other controllers is very poor. This controller is then applied to the two models and the result from the simulation can be seen in Figure 5.11.

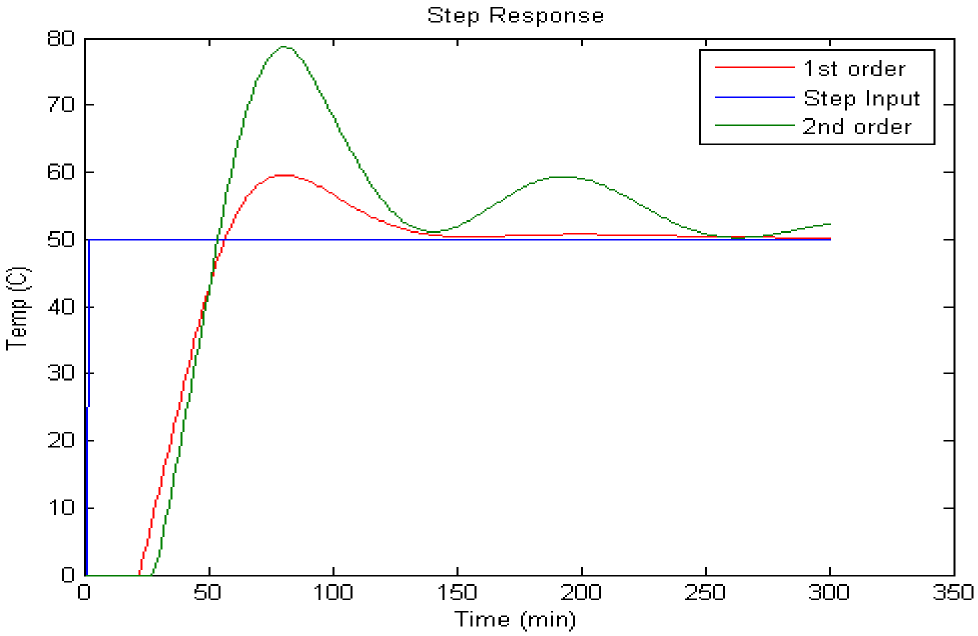

Figure 5.11 : Response to a Setpoint change as obtained from Matlab simulation

The bode plot for the two process can be seen in Figure 5.12.

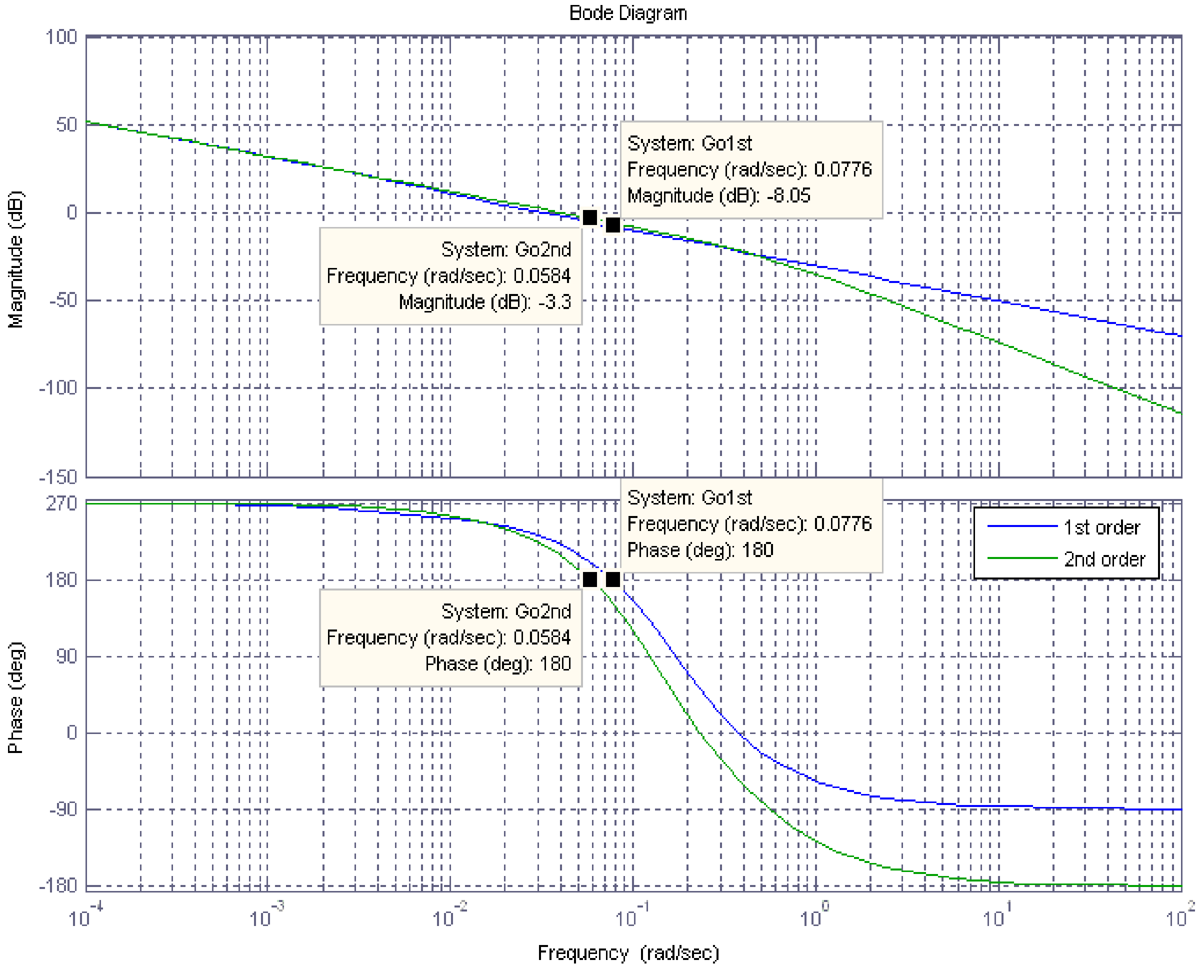

Figure 5.12 : Bode Plot

The tuning rules should give a better response from **Model 2** than **Model 1** as they are derived from **Model 2**. The controller values found from this method achieve a better response from **Model 2** than before. But the response from **Model 1** is still more desirable. This is because the methods to control a SOPDT system are not as developed as the FOPDT models.

The final phase is to identify the best possible controllers for the two models; the aim is to have as little overshoot as possible while retaining a short settling time. This fine tuning is done using Matlab where both models can be tuned simultaneously. The tuning involves a trial and error adjustment of the Proportional Gain and the Integral Time. This is the only method available as there are no tuning rules that can be applied to the criteria.

After a number of runs the controller parameters for both controllers are dialled in and the resulting system response can be seen in Figure 5.13.

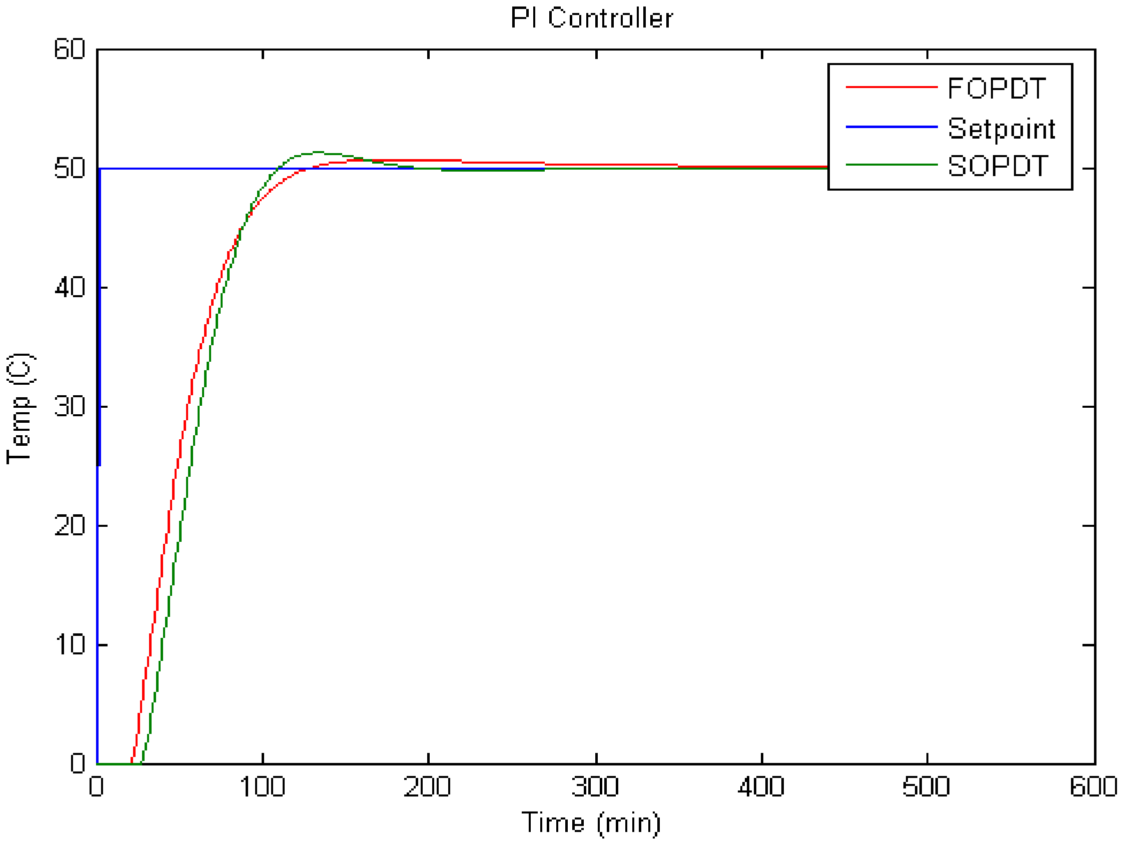

Figure 5.13 : Response to a Setpoint change as obtained from Matlab simulation

As can be seen there is a slight difference between the performances of the two models but both responses appear to have the desired shape. There is very little overshoot and also there is very little 'ringing' or oscillations, both of which are necessary to meet the specifications.

The Bode plot can be seen in Figure 5.14 where the Gain margins for both **Model 1** and **Model 2** are shown. As can be seen the two models are relatively similar at lower frequencies. The difference between both the Magnitude and Phase of the two models can be seen to differ greatly at higher frequencies. This gives an indication that **Model 2** is slightly more robust with regards to higher order disturbances.

So from the performance of the Systems response and from the Bode plot, these controllers appear to be the best controllers to meet the desired specification.

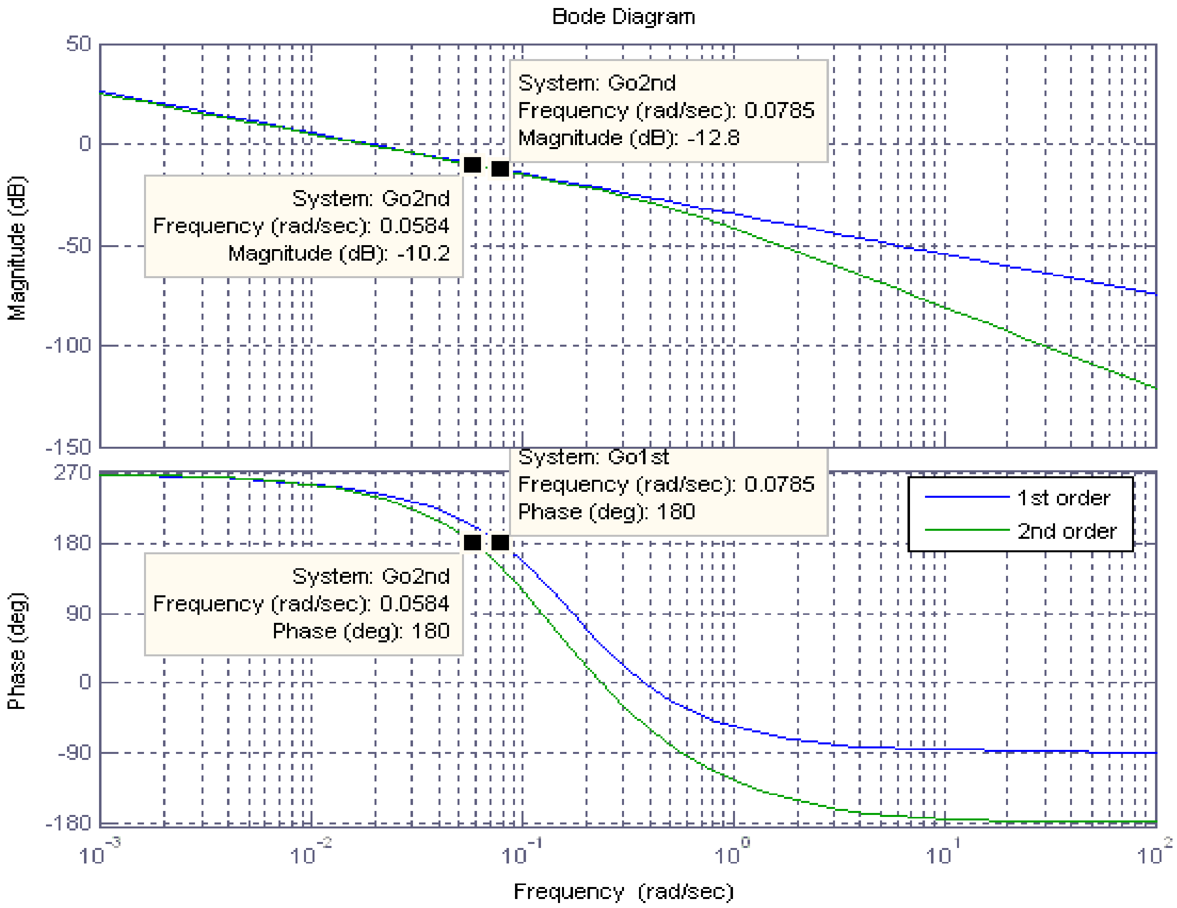

Figure 5.14 : Bode Plot

The next step is to test the controller values that are used in Figure 5.13 on the actual system and since there are two controllers with similar values the average of the two is used to implement on the actual process. The result can be seen in Figure 5.15.

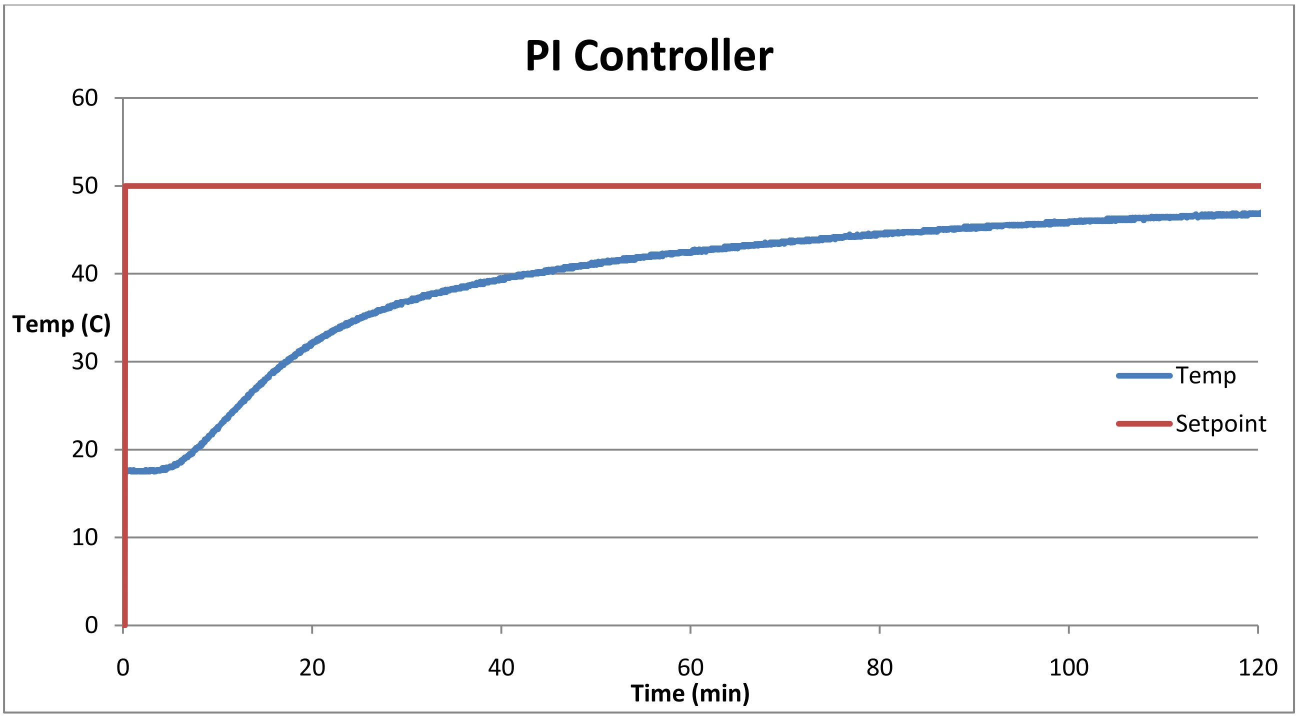

Figure 5.15 : Actual System response to a Setpoint Change

The performance of the controller is quite sluggish although it does not overshoot the setpoint. The controller appears to need an increase in the gain or a decrease in the integral time as the temperature has such a slow response.

The performance of the controller in Figure 5.13 is very similar to that as seen in Figure 5.15, for example the time to reach 45 °C in both cases is roughly 100 minutes. This shows that there is some correlation between the Empirical model and the actual system.

5.2.2 Smith Predictor Control

The next stage of testing involves the design of a Smith Predictor algorithm to try to improve the response of the system.

The best PI controller (integrating method) as achieved in the previous section is compared against the Smith Predictor with the same PI controller. The Smith Predictor is implemented using the difference equations as defined in Section 4.3. This is coded in the PLC using ladder logic and requires a large piece of memory.

The servo response of the Smith predictor can be seen in Figure 5.16, where the corresponding response from PI controller is also shown.

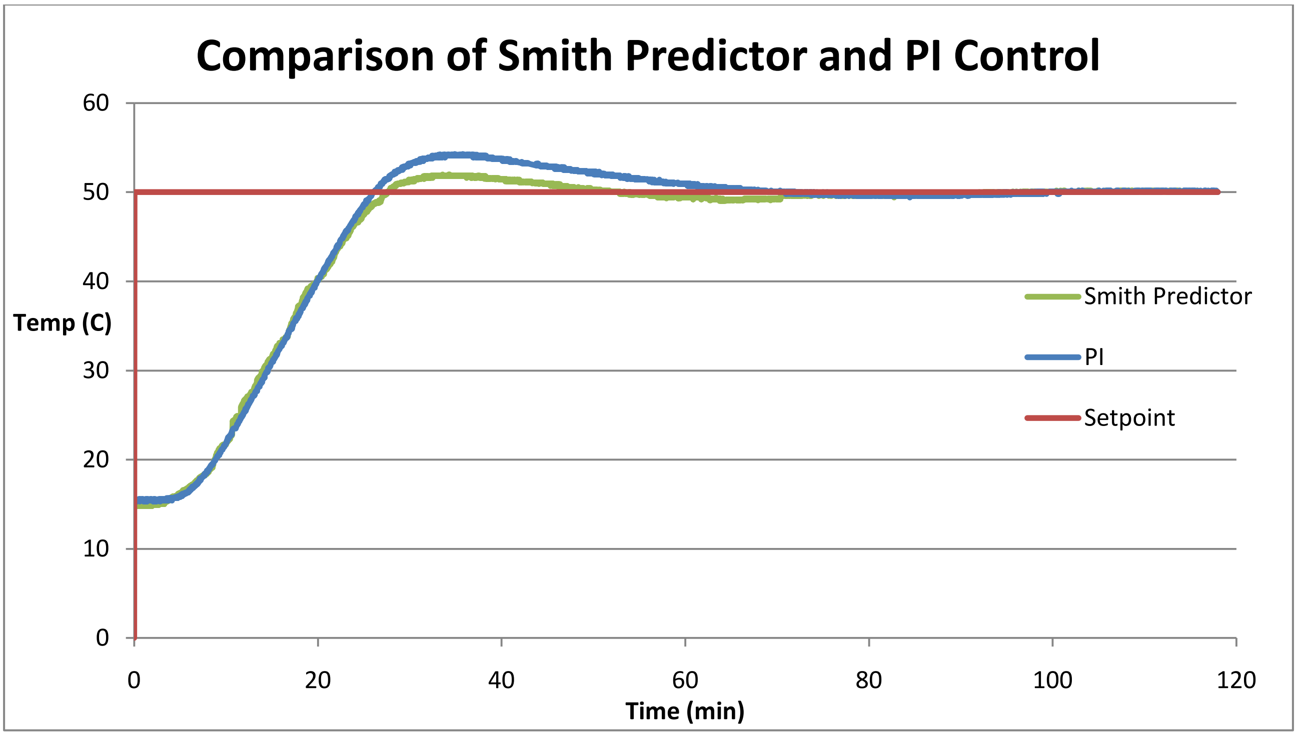

Figure 5.16 : Comparison of the Smith Predictor and PI control using the same controller on the actual system

The Smith Predictor is then applied to **Model 1** with the performance of each controller tuned to give optimal performance. The response can be seen in Figure 5.17.

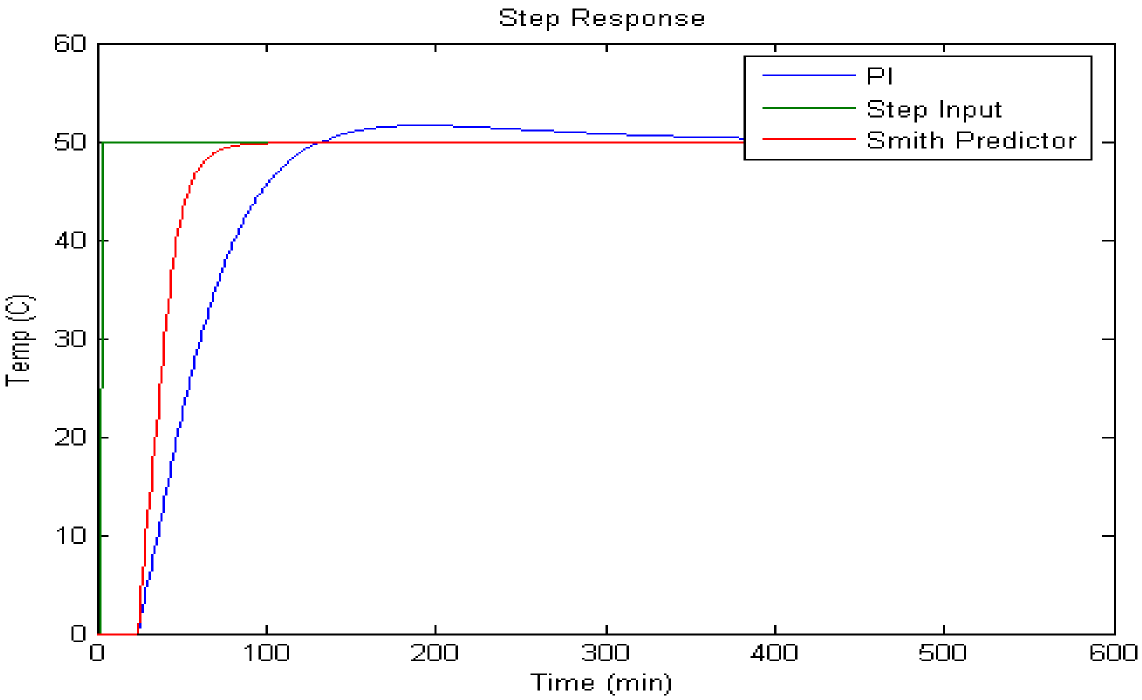

Figure 5.17 : Response to a Setpoint change as obtained from Matlab simulation

The *SIMULINK* model used to simulate the two controllers can be seen in Figure 5.18.

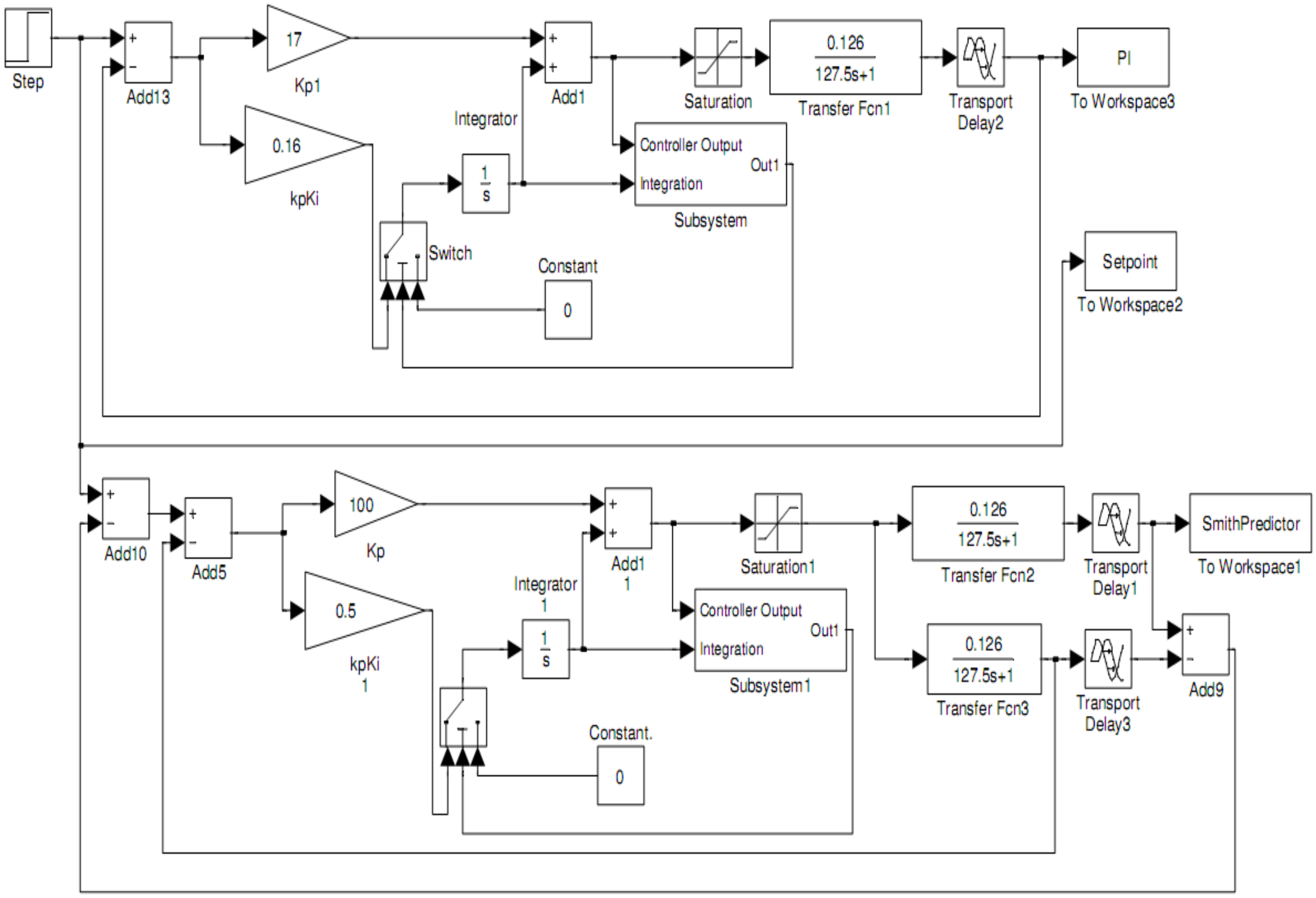

Figure 5.18 : ***SIMULINK*** **Model**

The next step was to simulate **Model 2** to investigate the performance of the Smith Predictor against the best PI controller achieved in the previous section.

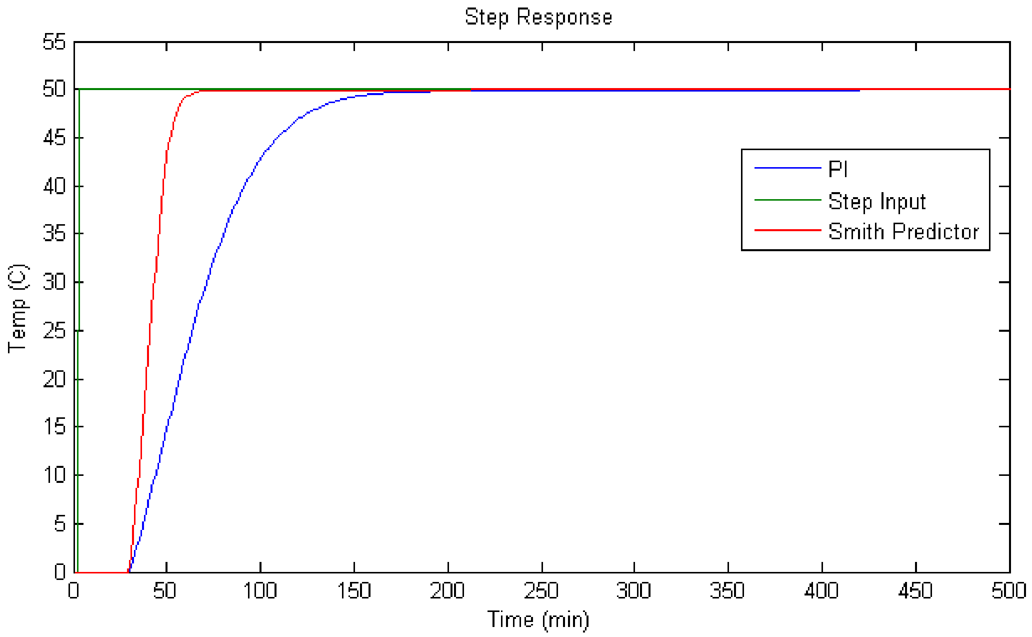

Figure 5.19 : Response to a Setpoint change as obtained from Matlab simulation

The *SIMULINK* model used to simulate the Smith Predictor and the PI controller can be seen in Figure 5.20.

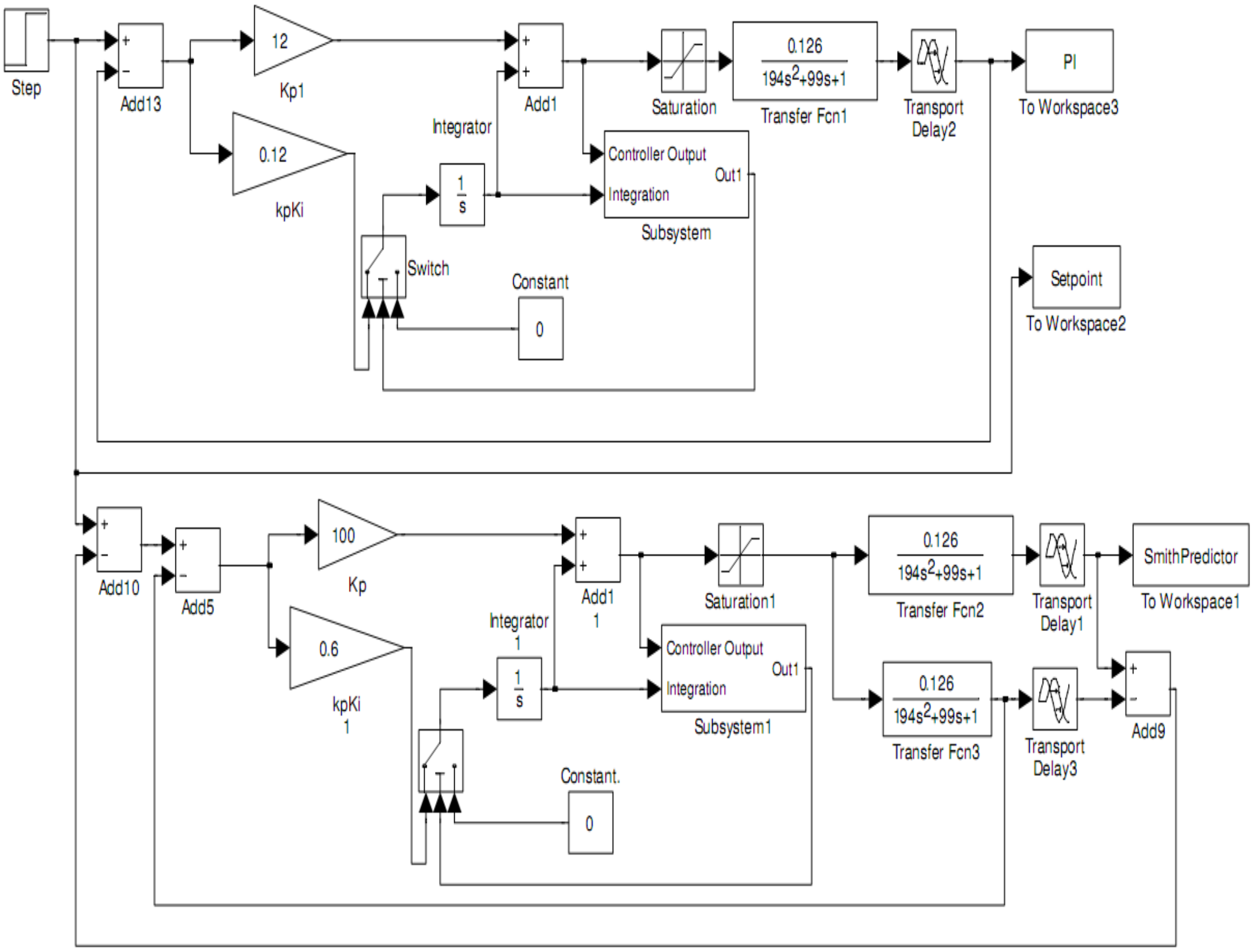

Figure 5.20 : *SIMULINK* Model

As can be seen the Smith Predictor is of the standard form where again the approximation is assumed to match the process exactly without any modelling error.

The Smith Predictor is applied to the two models and this is compared to the optimal controller as achieved in the previous section. This shows that the Smith Predictor offers an increase in performance even in process where the dead time is small compared to the process time constant, this also verifies the result seen in the experiment involving the laboratory model.

Table 1: Comparison of tuning rules and system performance

Parameter Estimation Method	Controller		Model 1	Model 2	Actual Process
Zeigler-Nichols (FOPDT Rules)	$G_c(s) = 45.5\left[1 + \frac{1}{65.89s}\right]$	%OS	61.2%	Unstable	8.8%
		10-90% Rise Time	17.2 mins		16.3 mins
		2% Settling Time	241 mins		57.5 mins
		Steady State Error	0%		0%
		IAE	2152		18302*
		GM	1.6245		
		PM	28.14°		
Integrating Method	$G_c(s) = 57.2\left[1 + \frac{1}{66.6s}\right]$	%OS	85.2%	Unstable	8.6%
		10-90% Rise Time	14 mins		15.5 mins
		2% Settling Time	246 mins		60.3 mins
		Steady State Error	0%		0%
		IAE	2204		17939*
		GM	1.295		
		PM	17°		
SOPDT Rules	$G_c(s) = 31\left[1 + \frac{1}{99s}\right]$	%OS	19.1%	56.9%	0%
		10-90% Rise Time	27.4 mins	27.5 mins	19 mins
		2% Settling Time	132 mins	398 mins	136 mins
		Steady State Error	0%	0%	0%
		IAE	2346	2346	22182*
		GM	2.53	1.47	
		PM	50.4°	28.4°	
Fine Tuning based on Model 1	$G_c(s) = 19\left[1 + \frac{1}{118.75s}\right]$	%OS	1.3%	2.79%	0%
		10-90% Rise Time	54.8 mins	54.8	64.42 mins
		2% Settling Time	111 mins	136 mins	206 mins
		Steady State Error	0%	0%	0%
		IAE	2696	2873	32953*
		GM	4.19	3.42	
		PM	66.77°	61.2°	
Fine Tuning based on Model 2	$G_c(s) = 14\left[1 + \frac{1}{100s}\right]$	%OS	3.59%	2.48%	0%
		10-90% Rise Time	59.4 mins	54.8 mins	66.9 mins
		2% Settling Time	173 mins	147 mins	223 mins
		Steady State Error	0%	0%	0%
		IAE	3121	2946	35102*
		GM	3.643	3.253	
		PM	60.12°	62.83°	

5.3 Summary

In summary this chapter has covered all the results from the laboratory testing and from the simulations. It can be seen that the model of the system does not represent the actual process as closely as we would desire. Also the implementation of the different control strategies on the PLC shows that the ideal controllers are hard to realise in real life.

Due to these two imperfections the control of the actual process requires some manual tuning and also some experience to achieve a desired level of performance. This manual tuning is uneconomical especially where the process time constant is large.

The direct comparison of the three identification methods and the associated tuning rules can be seen in Figure 5.21. It can be seen that differences between the three methods are small with the FOPDT method providing the best performance.

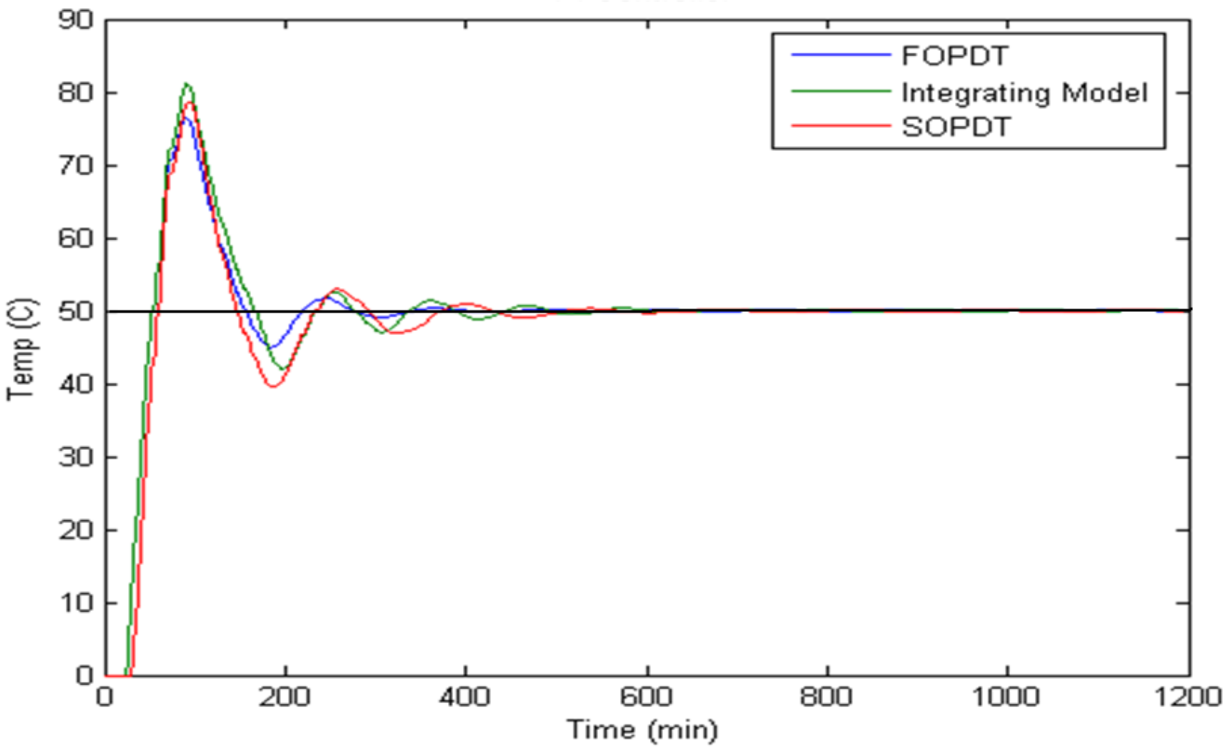

Figure 5.21 : Comparison of the different System Identification Methods

This can then be compared to the implementation of the actual system which can be seen in Figure 5.22 where the performance of the integrating methods appears to have the best performance of the three.

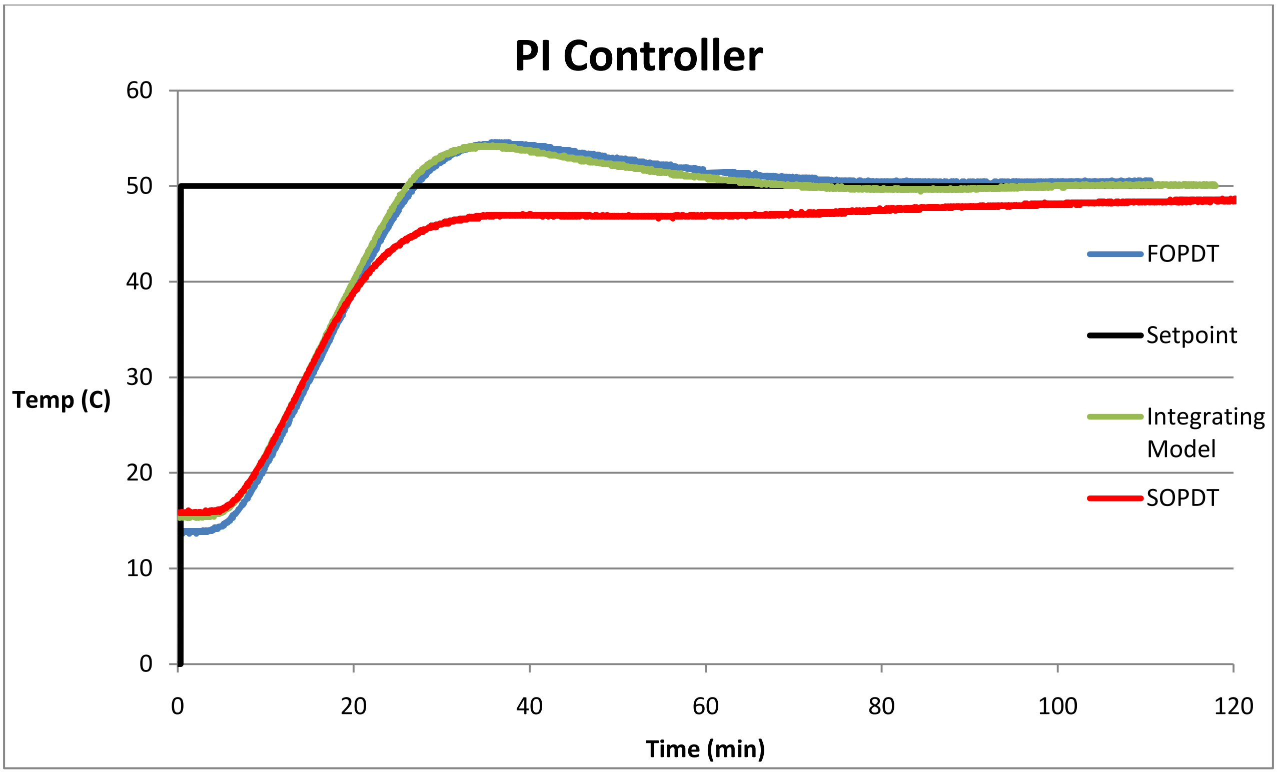

Figure 5.22 : Comparison of the System Identification methods as implemented on the system

The SOPDT method appears to be overdamped and takes a considerable length of time to reach the setpoint.

So in summary the FOPDT method appears to achieve the best performance in the simulation, while the performances of the SOPDT and Integrating method are not far off the FOPDT's performance. However when the three methods are implemented on the actual system the performance of the Integrating method appears to be the best with the FOPDT method offering a similar level of performance. The performance of the SOPDT method is not satisfactory due to its sluggish response.

6. Conclusions and Discussion

6.1 Introduction

This section describes the conclusions and achievements that can be drawn from this project. The results have been expressed in the previous section and they will be discussed further here. There were a number of aims for this project initiating from the commissioning of a Bench Scale Batch Reactor leading to the comparison of a number of temperature control algorithms. The process is then simulated using Matlab and *SIMULINK*. The goal for the project is to investigate if the correlation between the simulation and the discrete implementation is accurate enough to provide an aid to improving temperature control. If the models are accurate enough then the controller can be tuned using the simulation software before transferring to the physical plant. This would save valuable down time and also be more cost effective as tuning on the plant is hugely time consuming.

6.2 System Performance

There are two sections of the project the first being the system identification methods, while the second is the control of the actual process and the simulation experiments using *SIMULINK*.

6.2.1 Discussion on the system identification methods

There are a large number of methods that can be used in the empirical modelling of process, and these can be arranged into either on-line or off-line methods.

The off-line methods can also be referred to as open loop modelling. This is because the controller is disconnected and the process is in open loop, then the controller output is set to a constant value and the controlled variable is measured. There are both advantages and disadvantages associated with this method. The advantages of this approach are that it is simple to implement and it does not require any special code or software. Also there has been a lot of research carried out in this area which means that it is well known. Nonetheless there are disadvantages which include the risk that the process could go into an unsafe mode of operation because there is no control acting on it. However the problems and limitations of this approach are well documented and thus can be easily avoided. The open loop is still used quite open and it is necessary when a new process is being initialised.

The on-line methods, in the simplest form, use the controller to drive the systems to the edge of stability to identify a model for the system. The Zeigler-Nichols Ultimate Cycle is one of the most common online tuning methods however this does method does not return a model of the system.

The Ultimate Cycle method is used to derive controller parameters directly from the gain and period of oscillation of the process. There are also more advanced methods that can be used to derive controller parameters although they are not examined here, due to lack of state of the art software and hardware, they do exist and can be used to derive accurate control on the touch of a button. Having said that the advanced methods do not always provided accurate control as they have physical limitations.

There are methods to identify a model as described by Yuwana and Seborg (4) from an online approach but this is implemented in a different manner and can be used to define an accurate model for the system as shown by Kealy and O' Dwyer (5).

All of the online methods require a controller to be already in place and this is not always possible for example, when a new system is installed then the first step is to identify a controller and for which the open loop methods are crucial.

6.2.2 Discussion on the performance of the control algorithms

There are number of controller algorithms chosen for analysis in this project and these are namely PI, Cascaded PI-P and the Smith Predictor.

The PID controller is used as the benchmark controller here as it is the most common. It offers a high level of performance for minimal tuning. However the fine tuning need to meet the required specification is largely time consuming as it is a trial and error process. But as the PID controller is so common there is a large quantity of knowledge on how to tune the controller efficiently and this can be used to reduce the time and effort required to tune the controller.

The cascaded control structure is also a very common control strategy, and as it (in most cases) contains two PID loops offers similar level of performance to the single loop. However the cascade control does not offer any improvements in the servo response of the system. The cascaded control strategy offers large improvements in the regulatory response of the process over the single loop controller. The cascaded control structure is easily implemented as it is essentially two PID controllers combined. The cascade controller implemented here did not offer an improvement even when it is tuned correctly which also is largely time consuming.

The design of the original process was expected to have a large delay due to the capacitance of the Stainless Steel structure. In which case a dead time compensator would offer a large improvement over single loop control. However when the initial experiments were implemented it was evident that the time delay was not enough due to the large time constant. However the implementation of the dead time compensator showed that despite the large time constant it could offer slight improvements in performance.

The improvement in performance was first analysed using Matlab/*SIMULINK,* where it was shown to offer some improvement. This was then implemented on the laboratory model and the improvement in performance was very minimal. In this case the effort required to implement a Smith Predictor was not rewarded with a large increase in controller performance.

6.3 Conclusions

The main conclusion that can be drawn from this report is the fact that the use of more advanced control strategies does not necessarily offer an improvement. Also the empirical modelling method for the systems leaves a lot to be desired as the accuracy cannot be depended upon to produce models that can be used for tuning.

The conclusion is that the Integrating method can be used instead of the previous methods with minimal if any loss in performance while reducing the time needed to obtain controller parameters.

The use of the Smith Predictor can be seen to improve the performance of the system however this improvement is limited due to the ratio of time delay to the process time constant. The improvements offered by the smith predictor are not just limited to the system performance but the structure can also negate some of the modelling errors.

6.4 Future Work

There are a number of areas closely linked to this title that could lead to future work; this is because this project has covered a number of ideas. The main area in which future research is warranted is the initial system identification and tuning methods for discrete controllers that can be implemented efficiently on a PLC structure.

The area of controller tuning has been covered very well in published papers however the initial start up procedure can cause some problems which are not researched to the same extent. These problems are caused by the large initial offset which leads to an undesirable overshoot. The simple method to overcome this problem is to tune the controller to be more efficient and constraint at the start up phase however the consequences of doing this is a deterioration in the performance of the controller at the desired operating region.

Also the identifications methods (a few of the many available are covered earlier) used to model the system are not as accurate as desired. The models produced are not accurate to tune the controller to meet the specified criteria. The lack of accuracy in the model means that there is some fine tuning that has to be completed in order meets the criteria. A method to reduce the time consuming trial and error tuning could be developed to work specifically on batch reactors.

6.5 Summary

In summary the project has been completed with a number of findings and achievements. The final product is a bench scale batch reactor that is fully automated and is also operating with optimal efficiency. The laboratory testing of the different strategies proved to be hugely time consuming compared to the simulation. However the laboratory testing is necessary to highlight the performance of the actual system. The PLC is a very common method to implement the controllers described in this project.

The results found in the simulations cannot be expected to be found when the controller is implemented on a PLC on the actual process; this is down to a number of reasons. The first of which is the fact that the controller is implemented in a discrete time manner while the plant is of a continuous time nature. This approximation leads to discrepancy between the simulated version and the physical controller. The next reason is down to the fact the model of the physical plant is at best a loose approximation and thus the tuning rules obtained from the model can only have a limited degree of accuracy.

This project has shown a number of different identification methods that can be used to describe a process while also the differences between the control methods and the tradeoffs that exist between the different approaches.

Bibliography

1. *PID tuning rules for SOPDT systems: Review and some new results.* **Rames C. Panda, Cheng-Ching Yu, Hsiao-Ping Huang.** Taiwan : ISA, 2003.

2. *Introduction to Feedback.* **Martin, Leslie A.** s.l. : Massachusetts Institute of Technology, 1997.

3. **Lipták, Béla G.** *Process Control and Optimization.* New York : CRC Press, 2006.

4. *A new method for on-line controller tuning.* **Seborg, D.E. and Yuwana, M.** s.l. : AIChE J., 1982.

5. *Closed Loop Identification Of A First Order Plus Dead Time.* **O'Dwyer, Aidan and Kealy, Tony.** Cork : ISSC, 2002.

6. *Method for on-line identification of a first order plus dead-time process model.* **Mamat, R. and Fleming, P.J.** No.15, s.l. : Electronic Letters, 1995, Vol. Vol. 31.

7. *The Dynamics of Automatic Control.* **Oldenbourg, R. C. and Sartorius, H.** New York, USA : The American Society of Mechanical Engineers, 1948.

8. *Process dynamics and control.* **Seborg, D.e., Edgar, T.f. and Mellichamp, D.A.** Newyork : Wiley & Sons, 1989.

9. *Process Control: Modeling, Design and Simulation.* **Bequette, B. Wyane.** s.l. : Pretince Hall, 2003.

10. *Optimum Settings for Automatic Controllers.* **Ziegler, J. G. and Nichols, N.B.** New York : ASME, 1942.

11. *When to use cascade control.* **P. R. Krishnaswamy, G. P. Rangaiah, R. K. Jha, and P. B. Deshpande. "When to use cascade.** s.l. : Ind. Eng. Chem. Research, 1990.

12. *New Hybrid Control Approach for Cascade Temperature Control in Catalyst Regeneration Process.* **A. J. Modak, S. B. Deshpande and P. V. Dhawas.** India : IEEE, 2006.

13. *Robust Cascade Control System Design for Central Air-conditioning System.* **Jiangjiang Wang, Youyin Jing and Chunfa Zhang.** China : IEEE, 2008.

14. *PID Controller Tuning To Obtain Desired Closed Loop Responses for Cascade Control Systems.* **Par, Yongho Lee and Sunwon.** Korea : Ind. Chem. Research , 1998.

15. *Design of PLC-based Smith Predictor for Controlling Processes with Long Dead Times.* **Vodencarevic, Asim.** Hong Kong : Imecs, 2010, Vol. 2.

16. *Monitoring and assement of control performance for single loop systems.* **Huang, H.p. and Jeng, J.C.** s.l. : Ind. Eng. Chem. Res., 2002, Vol. 41.

17. *Display Systems For Production Engineers and Managers.* **An IProdE Guide.** s.l. : The Institute of Production Engineers, 1997.

18. *Chemical Process Control: An Introduction to Theory and Practice.* **Stephanopoulos, G.** New Jersey : Prentice-Hall, 1984.

19. *Automatic Control Systems.* **Kuo, B.C.** New Jersey : Prentice-Hall, 1991, Vol. 6th edition.

20. *Control System Engineering.* **Palm, W.J.III.** New York : John Wiley & Sons, 1986.

21. *Some Remarks on Padé-Approximations.* **Vajta, M.** Veszprém, Hungry : Tempus-Intcom Symposium, 2000.

22. The Chemical Engineers' Resource Page. [Online] GlobalSpec. [Cited: 23rd September 2010.] www.cheresources.com.

23. **Engineering, Glendoher.** Engineering Calculations. *Glendoher Engineering.* [Online] [Cited: 2 November 2010.] http://eng-calculations.com/.

24. **G. F. Franklin, J. D. Powell, and M. Workman.** *Digital Control of Dynamic Systems.* 1998. 3rd ed..

25. **Smith, Carlos A.** *Automated Continunous Process Control.* New York : John Wiley, 2002.

26. *Introduction to S88.* **Brown, David W.** Tokyo : JBF, 2007.

27. *A Design Approach for Feedback-Feedforward Control Systems.* **Behnia-Willison, Fariba, et al.** Christchurch, New Zealand : IEEE, 2009.