2003-09-01

Identification and PID Control of Time-delayed Processes

Tony Kealy
Technological University Dublin, tony.kealy@tudublin.ie

Identification and PID Control of Time-Delayed Processes

Tony Kealy (B. Eng. in Electronic Engineering)

M. Phil.

DUBLIN INSTITUTE *of* TECHNOLOGY

Institiúid Teicneolaíochta Bhaile Átha Cliath

Supervisor: Dr. A. O'Dwyer

Dublin Institute of Technology

School of Control Systems and Electrical Engineering

Submission Date: September 2003

Declaration

I certify that this thesis which I now submit for examination for the award of Masters of Philosophy (M. Phil.), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature Tony Kealy Date 23rd January 2004

Tony Kealy

Acknowledgements

I would like to acknowledge the following people without whose help I would have had great difficulty completing this research:

Aidan O'Dwyer, for the guidance, support and friendship over the last two years;

The School of Control Systems and Electrical Engineering, Dublin Institute of Technology, for the financial support and the use of their facilities during the research programme;

My family, for helping to shape me into the person I am today;

Olivia, for her love and understanding;

My colleagues in the D.I.T., Kevin Street, namely Mark, Dan, Steven, Ruiyao, Eileen, Pauline, Charlie, Michael, Finbarr, Dec and Elmar;

Abstract

This research was conducted primarily to investigate a selection of known methods for the identification of a mathematical model for processes with an inherent time delay, and subsequently, use the parameters of the identified model to regulate the processes via the PI/PID control strategy. The research examines the level of difficulty in performing or sanctioning different system identification methods in order to re-tune the controller.

The process model identification methods are carried out on both simulated and real processes in the first part of the research. The selected parametric identification methods are chosen because they encompass open- and closed-loop techniques, first-order-plus-dead-time and second-order-plus-dead-time process models and both frequency- and time-domain identification methods.

The second part of the research considers the PI/PID control strategy to regulate the processes identified using the aforementioned process model identification methods. The link between the identification and control sections in this thesis is the tuning rule. Even though the PID control loop is the most common industrial controller, poor tuning has contributed to the control loop not working as well as it should. The thesis considers the level of expertise required and the complexity of implementing eleven representative tuning rules using different PID controller structures.

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

1.1 Background of Research

Despite rapid evolution in control hardware over the past 50 years, the PID controller remains the workhorse of process control. In process industries, more than 90% of the control loops are of the PID type (Astrom and Hagglund (1995)). Sixty years after the publication of the Ziegler-Nichols tuning rule (1942) and with numerous papers published on the tuning methods since, one might think the use of PID controllers has already met our expectations. Unfortunately, this is not the case. Surveys of Bialkowski (1993), Ender (1993), McMillan (1994) and Hersh and Johnson (1997) have revealed the following in a number of different industries:

Pulp and paper industry [over 2000 loops] (Bialkowski (1993))

- Only 20% of loops worked well
- 30% gave poor performance due to poor controller tuning
- 30% gave poor performance due to control valve problems
- 20% gave poor performance due to process and/or control system design problems

Process industries (Ender (1993))

- 30% of loops operated in manual mode
- 20% of controllers used factory tuning
- 30% gave poor performance due to sensor and control valve problems

Chemical process industry (McMillan (1994))

- Half of the control valves needed to be fixed
- Most poor tuning was due to control valve problems

Manufacturing and process industries (Hersh and Johnson (1997))

- Engineers and managers cited PID controller tuning as a difficult problem

Surveys indicate that the process control performance is, indeed, "not as good as you think" (Ender (1993)). The reality leads us to reconsider the priorities in process control research. First, an improved process and control configuration redesign (e.g., selection and pairing of input and output variables) can improve control performance. Second, control valves contribute significantly to the poor control performance. Third, and probably the easiest way to improve control performance, is to find appropriate parameters for PID controllers (Yu, 1999). The procedure of finding the controller parameters is called tuning. The control loop performs well if the PI/PID controller parameters are chosen properly. It performs poorly otherwise, e.g., the system may become unstable. PI/PID controller parameters may be chosen in two different ways. One approach is to choose some controller parameters, to observe the behaviour of the feedback system, and to modify the parameters until the desired behaviour is obtained. A second approach is the approach explored in this thesis i.e. firstly develop a mathematical model that describes the behaviour of the process. The parameters of the controller are then determined using some method for control design. The development of a mathematical model for a process is often the first step undertaken in the design of a controller. System identification is the theory, art and practise of building mathematical models of dynamical systems from observed input-output data. **Model Sets** or **Model Structures** are families of models with adjustable parameters. **Parameter Estimation** amounts to finding the "best" values of these parameters. The system identification problem amounts to finding both a good model structure and good numerical values of its parameters.

Parametric Identification Methods are techniques to estimate parameters in given model structures. Basically these techniques involve finding (by numerical search) those numerical values of the parameters that give the best agreement between the output of the model (simulated or predicted) and the measured output.

Non-Parametric Identification Methods are techniques to estimate model behaviour without necessarily using a given parameterised model set. Typical non-parametric methods include correlation analysis, which allows the estimation of an impulse response of a system, and spectral analysis, which allows the estimation of a frequency response of a system.

Identification methods may also be sorted into two further categories, open loop methods and closed loop methods. An open loop system is a system that does not use feedback. Thus the output has no effect on the signal entering the system. Figure (1.1) is an example of an open loop system.

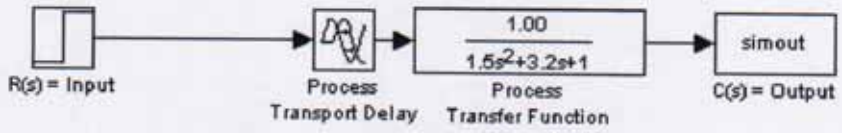

Figure (1-1) - Open Loop System

In a closed loop system, the value of the output is fed back to modify the input to the system. Figure (1.2) is an example of a closed loop system. An advantage of the closed loop system identification approach is that the identification test can be carried out while the process is running and therefore the plant does not have to be taken out of commission.

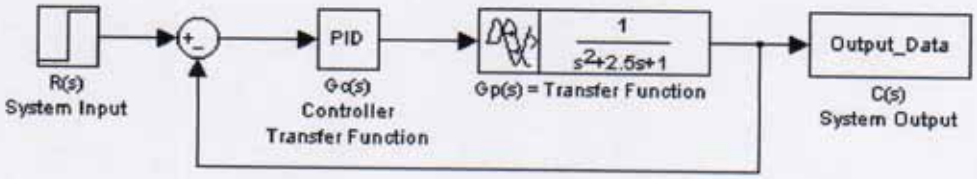

Figure (1-2) - Closed Loop System

Experimental open loop tests have the advantage of simplicity. However, the parameters identified may vary with process operating conditions and the step change size and direction. In addition, the process must be sufficiently disturbed by the change, to obtain reasonably accurate dynamic information, with the possibility that

the process may be forced outside the region of linear behaviour. There is also reluctance among plant management to permit such disturbances to be introduced for parameter estimation purposes. The process time scale must also be known in advance in order to determine when the transient response has been completed.

Some process control systems by-pass the process model identification stage. In this method of control, a test is initiated to determine two (or three) pieces of information about the process. The controller parameters are subsequently calculated directly from this information. An example of this method is the relay auto-tuning method discussed in Chapter 4. In this thesis, we are deliberately identifying a process model initially. We therefore have valuable information regarding the process under test, e.g. the ratio of time delay to time constant. Subsequently, it may be decided if PID control is appropriate for the process under test. PID would not be appropriate if, for example, the time-delay is much larger than the time constant.

It has been recognised that a first-order-plus-dead-time (FOPDT) or second-order-plus-dead-time (SOPDT) parametric model may, in general, represent process dynamics (Shaw, 1993). The FOPDT model structure is shown in equation (1.1).

$$G_m(s) = \frac{K_m e^{-d_m s}}{1 + \tau_m s} \tag{1.1}$$

K_m is the model gain, τ_m the model time constant and d_m the model time delay (or dead time). The ratio $\tau = \frac{d_m}{d_m + \tau_m}$, which has the property $0 \leq \tau \leq 1$, is called the "normalised dead time". This quantity can be used to characterise the difficulty of controlling a process.

The SOPDT model can have two structures, shown in equations (1.2) and (1.3).

$$G_m(s) = \frac{K_m e^{-d_m s}}{\tau_m^2 s^2 + 2\tau_m \zeta_m s + 1} \tag{1.2}$$

K_m and d_m are the model gain and time delay respectively, ζ_m is the model damping factor and τ_m is the reciprocal of the under-damped natural frequency of the system. The alternative SOPDT model structure is given by

$$G_m(s) = \frac{K_m e^{-d_m s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{1.3}$$

τ_1 is the first time constant and τ_2 is the second time constant.

Though the majority of control loops are PI/PID, this does not mean that there aren't better ways to control industrial processes, particularly those containing pure or dominant time delay. It is worth noting that predictive controllers are commonplace in industry, in many cases controlling time-delayed processes. There are a number of publications (and opinions), in the process control area, which would strongly advise against using PID for time-delayed processes. This thesis considers time-delayed processes where, in general, the dominant time constant is larger than the time delay.

In Chapters 2, 3 and 4 of this thesis, a process model is identified using time- and frequency-domain, open- and closed-loop methods. In Chapter 5 of this thesis, the parameters of the process model are used in the selection of the PI/PID controller parameters by way of tuning rules described by O'Dwyer (2003). The model identification techniques explored in this thesis are firstly explained and then carried out in simulation and implemented on a real plant, the PT326 process trainer from Feedback Instruments Limited. The equipment used to implement, for the remainder of this thesis, both open loop and closed loop identification techniques on the process trainer are as follows:

- PC
- MATLAB software (Version 6.0 (R12))
- SIMULINK software (Version 4.0 (R12))
- HUMUSOFT Real-Time Library (Version 3.10)

- AD 512 Data Acquisition Card
- Process Trainer PT326
- 37-Pin D-type connector, 37-Pin cable and connector block.

This thesis focuses on simulated processes and a laboratory scale pilot plant. It does not propose to address industrial systems. The simulation and laboratory study does not deal with the following:

- The presence of added noise
- The presence of process non-linearities
- Possible bias on sensors
- Higher order process dynamics

Industrial systems are expected to contain all of the above terms.

Figures (1.3) and (1.4) demonstrate the connections between the components of the system, for implementation of the identification techniques in open loop and closed loop, respectively. Figure (1.5) is a picture of the PT326 process trainer.
Chapter 2 examines the open loop identification techniques. A simulation test is carried out on all identification methods described. The simulated process is common to all the open loop techniques and is shown in equation (1.4).

$$G_p(s) = \frac{1e^{-s}}{1+s} \tag{1.4}$$

Chapter 3 examines the closed loop time-domain identification techniques and Chapter 4 discusses the closed loop relay based identification techniques. Each chapter begins with an introduction sub-section and ends with a conclusions sub-section.

The identification methods described are evaluated by comparing the model step response to the process step response in either the time-domain or the frequency-domain. Both the simulation and implementation environments are utilised in this evaluation. A formal mathematical evaluation of the methods is not discussed.

A number of quoted references in this section confirm that a considerable number of PID loops have poor performance in which merely adequate control is achieved. This is attributed to three reasons, (1) process and/or control system design problems, (2) control valve problems, (3) poor controller tuning. Apart from the possibility that PID is inappropriate in some cases, the real reasons why loops are poorly tuned are that the plant personnel don't have the expertise and/or the methods for good tuning are deemed to be too time-consuming. Its popularity is probably partly due to the fact that it will control the process reasonably well in spite of the three identified problems i.e. it has inherent robustness.

1.2 Thesis Contributions

In a report by Hersh and Johnson (1997) it is stated that engineers and managers cited PID controller tuning as a difficult problem. This thesis has attempted to investigate the particular problem of poor tuning by considering the level of expertise required and the time needed to carry out initial tests to identify a process model and subsequently, the complexity or otherwise of implementing tuning rules designed to control time-delayed processes.

The thesis evaluates open- and closed-loop, time- and frequency-domain methods of identifying a mathematical model for a time-delayed process. Both simulated processes and a real process, the PT326 process trainer, are used in this evaluation. The parameters of the process model are subsequently used to control the process using tuning rules that minimise a performance index.

The findings of this study show that the initial model identification stage is complex and time-consuming. One problem encountered is the lack of standardisation in the identification techniques. For example, there isn't a set of unified regulations regarding step size, step polarity or range of frequencies when carrying out the identification tests. This may be attributed to the fact that two processes are never going to be identical to each other. After the process model is identified, the tuning

rules are easily implemented. The study found that the PI/PID controller tuning rules were also lacking in standardisation. The author found it difficult to compare the resulting closed loop performance as the parameters of the process model used in the tuning rule are obtained using different identification tests. However, the book by O'Dwyer (2003) has addressed this problem somewhat.

Minor corrections were made to (typing) errors in the contribution by Mamat and Fleming (1995) and the SOPDT model based tuning rule by Huang *et al.* (1996).

The findings of this study have been reported in four conference papers (Kealy and O'Dwyer, 2002a, 2002b, 2003a, 2003b).

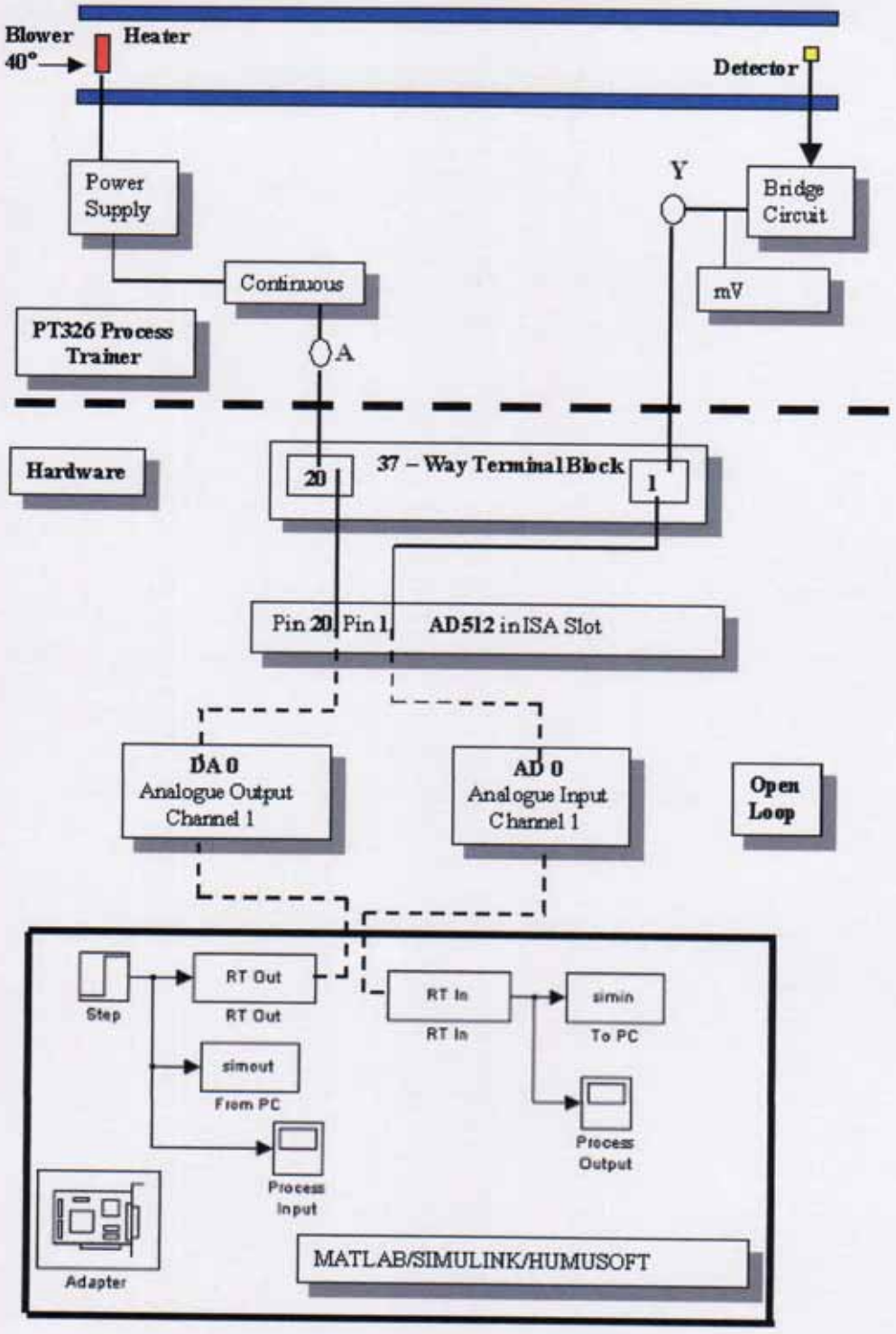

Figure (1-3) - System for implementing the identification techniques in open-loop

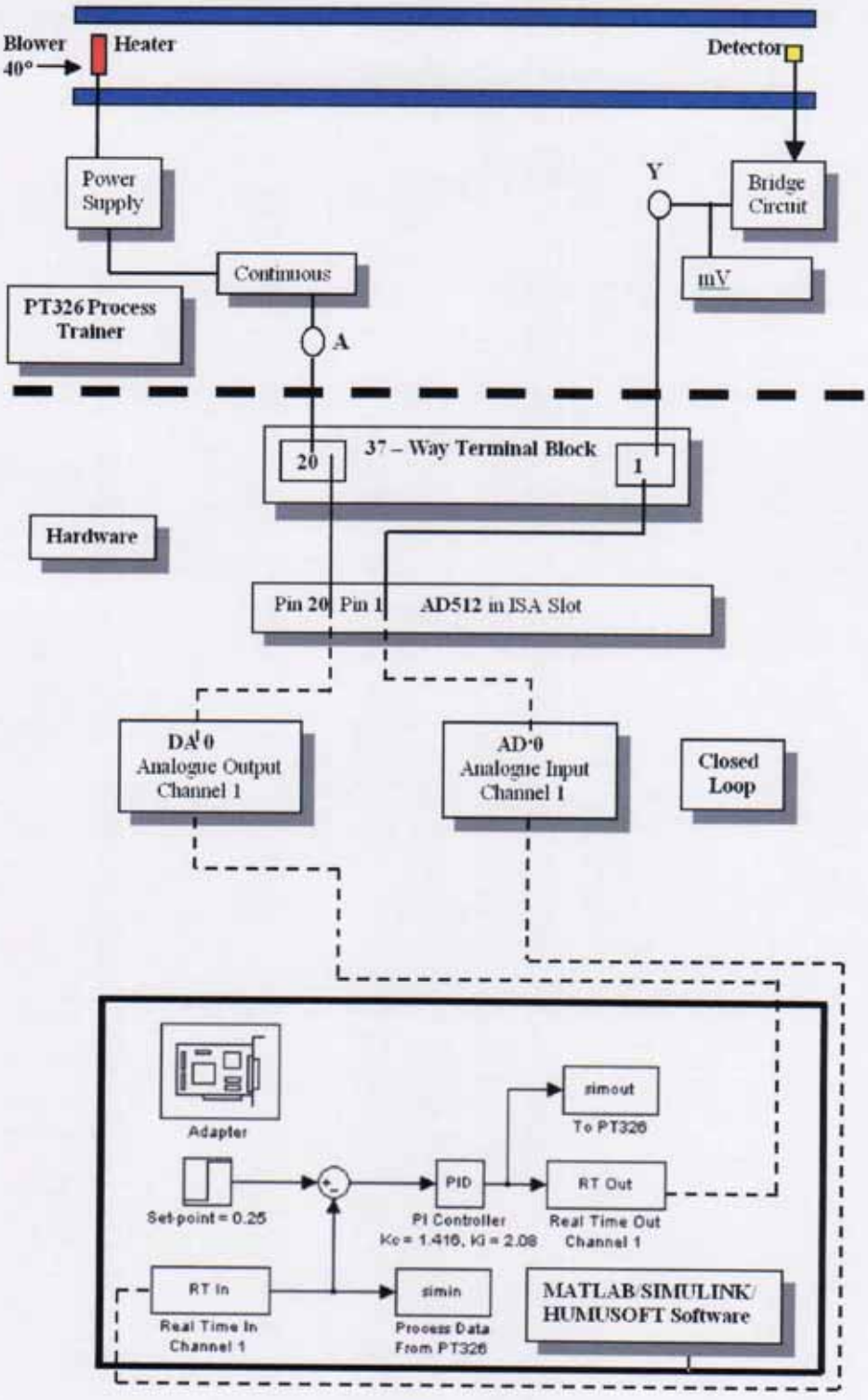

Figure (1-4) - System for implementing the identification techniques in closed loop

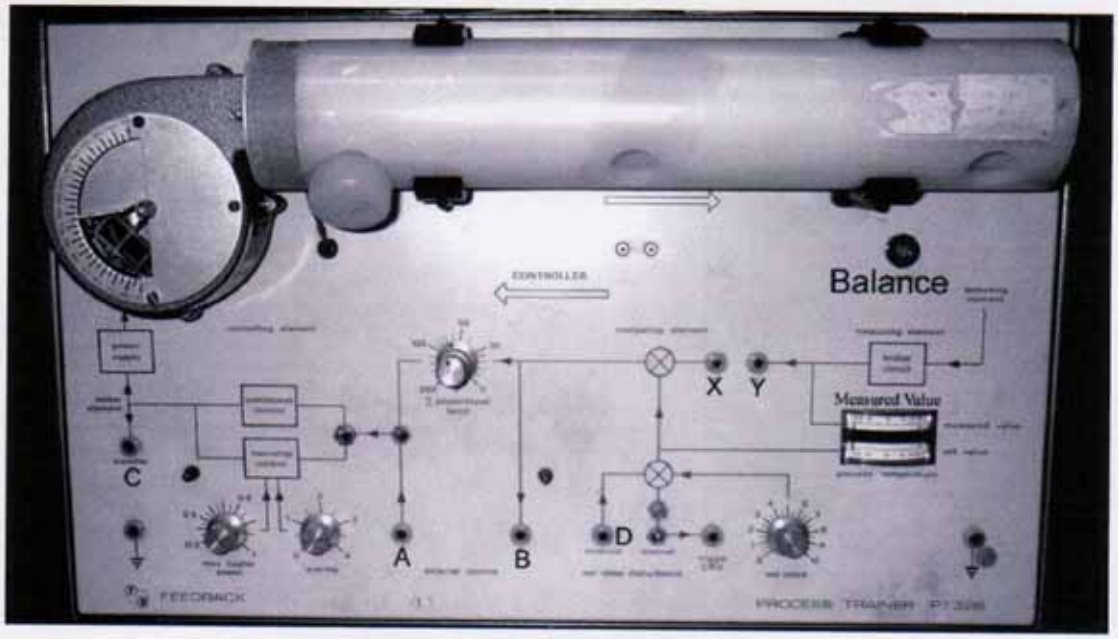

Figure (1-5) - PT326 Process Trainer

Chapter 2 : Open Loop Identification of a Process Model

2.1 Introduction: Outline of Identification Methods Investigated

One of the first process model identification guidelines was developed by Ziegler and Nichols (1942). In this method, the time constant, τ_m, and the time delay, d_m, of a first-order-plus-dead-time process model (equation (1.1)) are obtained by constructing a tangent to the experimental open loop step response at its point of inflection. The intersection of the tangent with the time axis provides an estimate of the time delay. The time constant is estimated by calculating the intersection of the tangent with the normalised value of the steady state output. The process model gain, K_m, is calculated by dividing the change in steady state output by the change in steady state input to the process. This method of calculating K_m is common to all four approaches investigated. See Figure (2.1) for demonstration of these methods.

An alternative process model identification technique is a "graphical" identification method. The time delay and time constant of a FOPDT process model are estimated from recording the time taken to reach different points on the response data. The time delay is taken at the time when the output has first responded to a change in input. The time constant is the time at which the output has reached 63% of the steady state output, less the time delay.

Another alternative is the "two-point" method as described by Shaw (1993); in this method, the time delay and time constant are calculated from the time taken to reach 28% and 63% of the final steady state output.

Nishikawa *et al.* (1984) describe a method of process parameter (including time delay) estimation using the 'characteristic areas' (i.e. the area underneath the step response output curve) of either the open loop or closed loop process step response. Astrom and Hagglund (1995) apply a related method, which they call the Method of

Moments, to estimate the parameters of a first-order-plus-dead-time process model. The input applied to the system in this latter method is not a step input but a pulse input.

In the frequency-domain, a FOPDT and a SOPDT process model is identified using two algorithms described by O'Dwyer (2002). The algorithms employ a two-stage approach. In the first stage, the parameters of the model are analytically determined. The second stage then determines the best parameter estimates by using a gradient algorithm to facilitate minimisation of an appropriate cost function.

This chapter firstly describes the time-domain modelling methods in detail, namely the graphical method, the two-point method, the area method and the method of moments. The methods are examined in simulation and implementation. Then the two-stage frequency-domain method is examined, again in simulation and implementation.

2.2 Time-domain modelling

The dynamics of a system can be determined from the response of the process to pulses, steps, ramps or other deterministic signals. The dynamics of a linear system are, in principle, uniquely given from such a transient response experiment. This approach requires, however, that the system is at rest before the input is applied, and that there are no measurement errors. In practice, however, it is difficult to ensure that the system is at rest. There will also be measurement errors, so the transient response method, in practice, is limited to the determination of simple models. Models obtained from a transient experiment are, however, often sufficient for PID controller tuning. The methods are also very simple to use (Astrom and Hagglund, 1995).

A static process model gives the steady state relationship between the input and the output signal. A dynamic model should give the relationship between the input and the output signal during transients. It is naturally much more difficult to capture dynamic behaviour. This behaviour is, however, very significant when discussing control problems. Fortunately, there is a restricted class of models that can often be

used for linear, time-invariant systems. Such models can also be used to describe the behaviour of control systems where there are small deviations from equilibrium. The fact that a system is linear implies that the superposition principle holds. This means that if the input u_1 gives the output y_1 and the input u_2 gives the output y_2 it then follows that the input $au_1 + bu_2$ gives the output $ay_1 + by_2$. A system is time-invariant if it is stationary or its characteristics do not change with time. A very useful property of linear time-invariant systems is that their response to an arbitrary input can be completely described in terms of the response to a simple signal. Many different signals can be used to describe a system. The time domain approaches investigated generate responses from step- or pulse-tests. The characteristics of the process responses are then used to back-calculate the parameters of an assumed process model.

2.2.1 Graphical Approach

The parameters in the first-order-plus-dead-time model shown in equation (1.1) can be determined graphically by examining the response of the system to a unit step test as shown in figure (2.1). The static gain, K_m, is obtained from the final steady-state level of the process output. Remember that the process output must be scaled with the change in the control variable. The intercept of the tangent to the step response that has the largest slope with the horizontal axis gives an estimate of d_m (See figure (2.1)). The time delay estimate, d_m, is not obtained in this manner in this thesis as the tangent is difficult to obtain. The time delay d_m can also be obtained as the time between the onset of the step and the time when the output, $s(t)$, has reached a few percent of its final value. The actual percentage value depends on the amount of noise in the system. Typical values are between 0.5% and 2%. This is how the time delay is obtained in our graphical method. There are a number of methods to determine τ_m, the process model time constant. The original Ziegler and Nichols (1942) method determines τ_m from the distance AC in figure (2.1), where the point C is the time when the tangent intersects the line $s(t) = K_m$. A second method, the so-called alternative tangent and point method, proposed by Murrill (1967), determines τ_m from

the distance AB in figure (2.1), where B is the time when the step response has reached the value $0.63K_m$. This latter approach is how the time constant is estimated in our graphical method.

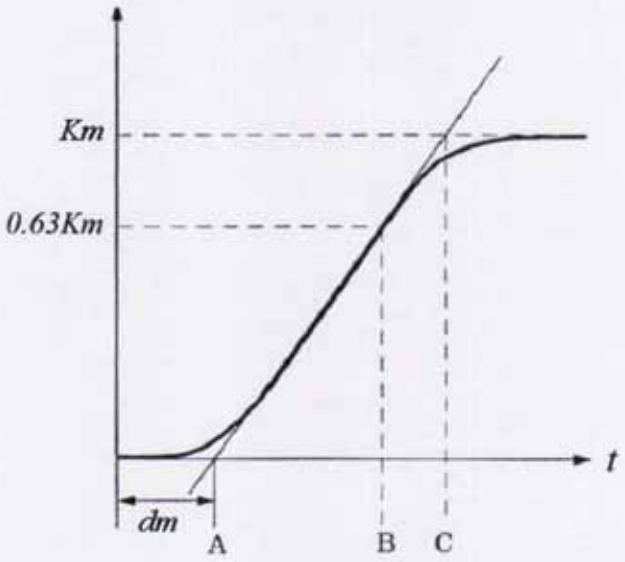

Figure (2.1) Graphical determination of three-parameter models for systems with a monotone step response (Astrom and Hagglund, 1995)

2.2.1.1 Simulation:

A simulation result shows the implementation of the method when the time delay is obtained as the time between the onset of the step and the time when the output s(t) has reached 2% of its final value; the time constant is obtained from the time when the step response has reached 63% of its final value, as described in section 2.2.1.

Figure (2.2) SIMULINK file with process parameters for graphical method

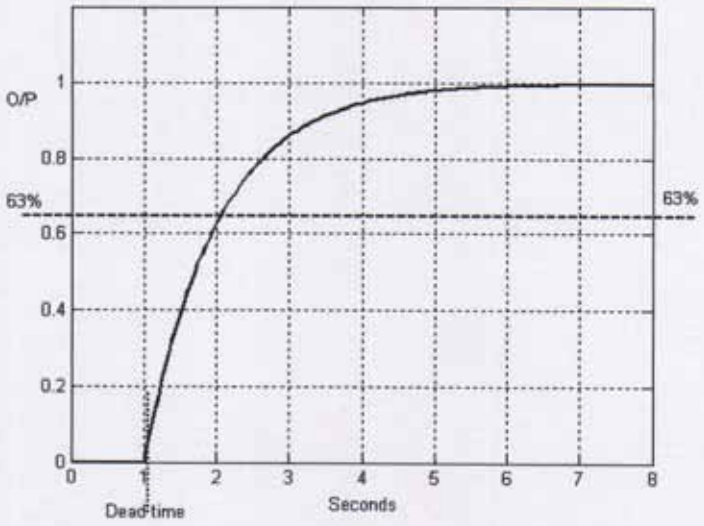

Figure (2.3) Process open loop step response from file in figure (2.2)

The program entitled "OL_Graphical_Sim_1" in Appendix 2 section 1, page A2, gives the following results:

- Model gain, K_m, = 1.00
- Model time constant, τ_m, = 1.00 seconds
- Model time delay, d_m, = 1.05 seconds

The three first-order-plus-dead-time parameter values determined by this method are now compared to the known parameter values and shown in figure (2.5).

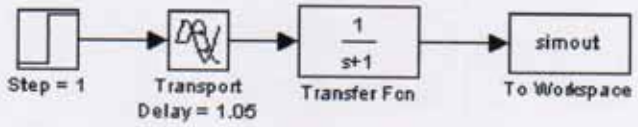

Figure (2.4) Model parameters obtained using graphical method

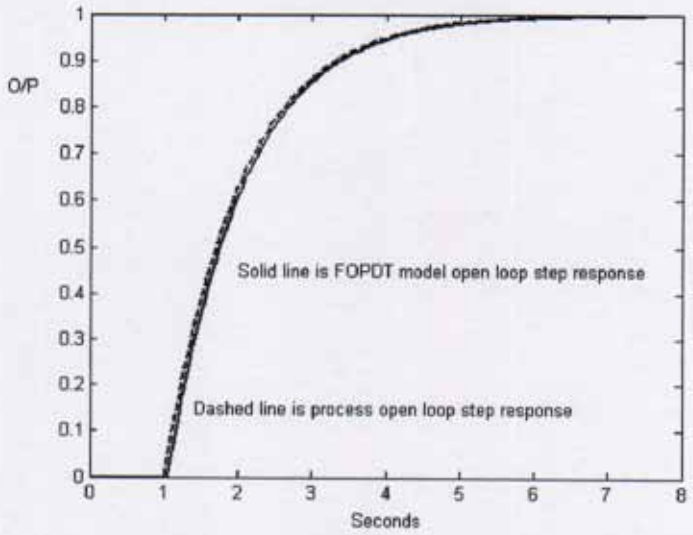

Figure (2.5) Comparison of process open loop step response and model open loop step response using graphical method to obtain model

Validation: The plot in figure (2.5) show that the open loop step response of the model obtained using the graphical method is an accurate representation of the open loop step response of the simulated process.

2.2.1.2 Implementation:

Using the MATLAB, SIMULINK and HUMUSOFT software, a step is applied to point (A) on the PT326 process trainer in figure (1.5) (point A also marked in figure (1.3)). The process output data is collected from point (Y). The process is first calibrated by sending a 0V output signal to point (A) and adjusting the "Balance" potentiometer for the input signal at point (Y) to read 0V (see figure (1.5)). After the calibration is complete, a step voltage of 2.5V is applied to the process and the resulting data used to identify the parameters of the system. The "Measured Value" indicator reads 27°C when the process output is 0V and 38°C when the process output is 2.5V. This range of values is within the linear range of the process. This is known because of previous experiments carried out on the PT326 process trainer. The file in figure (2.6) is used:

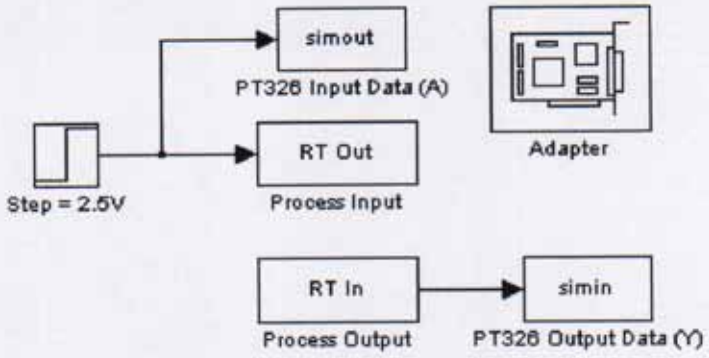

Figure (2.6) SIMULINK/HUMUSOFT file for open loop step response

The simulation parameters step size and the real-time sample time are both set to 0.01 seconds. The reason for this is to allow sufficient data points to be collected to accurately represent the process output. Note also that the step function in MATLAB has a one second delay before the step is activated. The data is plotted and shown in figure (2.7).

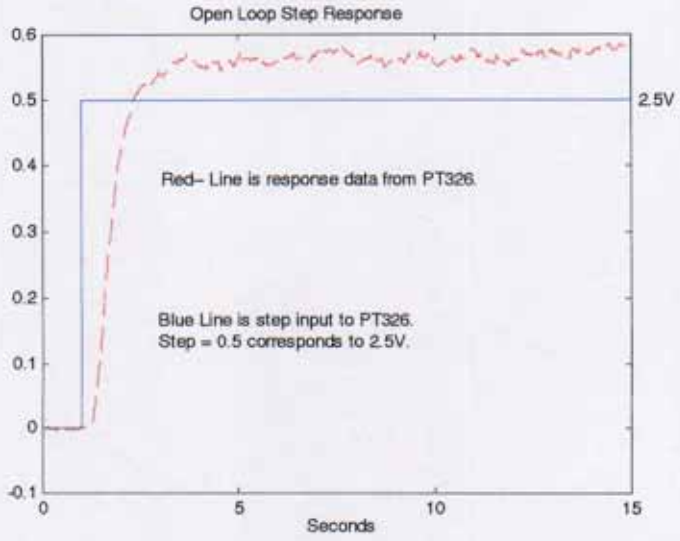

Figure (2.7) PT326 process trainer open loop step response data in red

The algorithm entitled "OL_Graphical_Imp_1" in Appendix 2 section 1, page A2, is used on the process data to determine graphically the process model gain, time delay

and time constant. The presence of noise on the process data in figure (2.7) must be allowed for when the time delay is being calculated. The time delay is obtained from the data in figure (2.7) as the time between the onset of the step and the time when the output has reached 0.87% of its final value. A noise level of less than 0.87% therefore cannot affect the time delay estimation. The results are as follows:

- Model gain, K_m, = 1.15
- Model time constant, τ_m, = 0.6 seconds
- Model time delay, d_m, = 0.26 seconds

The FOPDT model parameter values are inserted into a SIMULINK file and the open loop step response of the process is compared with the open loop step response of the model shown in figure (2.8).

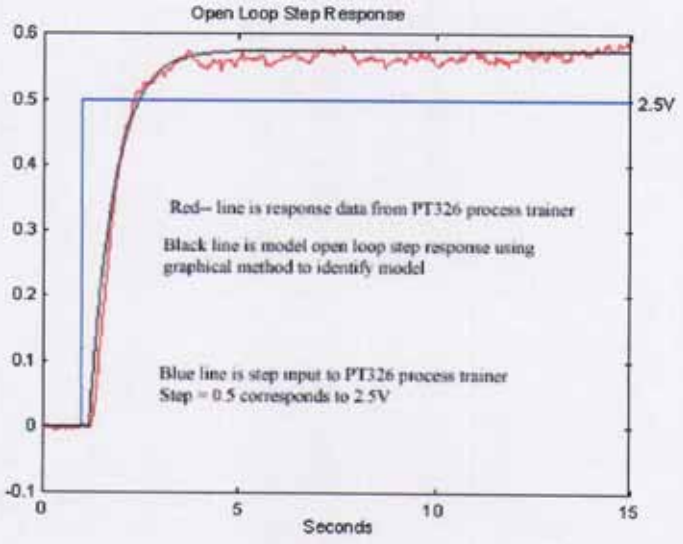

Figure (2.8) Validation of graphical method model with process open loop step response

Validation: The process and model open loop step response comparison in figure (2.8) show that the graphical method works very well in identifying a FOPDT model for the PT326 process trainer.

2.2.2 Two-point Algorithm

In the two-point algorithm approach, the steady state gain is determined as in the graphical method. The times taken for the process output to reach 28% and 63% of the final steady state output (T_{28} *and* T_{63}, respectively), are used to determine the time constant and the dead time based on solving the following simultaneous equations (Shaw, 1993):

$$T_{63} = d_m + \tau_m \tag{2.1}$$

$$T_{28} = d_m + \frac{\tau_m}{3} \tag{2.2}$$

Therefore,

$$T_{63} - T_{28} = \frac{2}{3}\tau_m \tag{2.3}$$

$$\tau_m = \frac{3}{2}(T_{63} - T_{28}) \tag{2.4}$$

Also,

$$T_{28} = d_m + \frac{T_{63} - T_{28}}{2} \tag{2.5}$$

$$d_m = T_{28} - \frac{T_{63}}{2} + \frac{T_{28}}{2} \tag{2.6}$$

$$d_m = \left(\frac{3}{2}T_{28}\right) - \frac{T_{63}}{2} \tag{2.7}$$

$$d_m = \frac{1}{2}(3T_{28} - T_{63}) \tag{2.8}$$

The two-point algorithm and the graphical approach identification methods are based on evaluation of the step response at two points only. Such methods are quite sensitive to measurement noise.

2.2.2.1 Simulation:

A simulation result shows the implementation of the method. A step is applied to the simulated process in figure (2.9) and the response data used to determine the FOPDT parameters by the two-point algorithm method.

Figure (2.9) SIMULINK file with process parameters

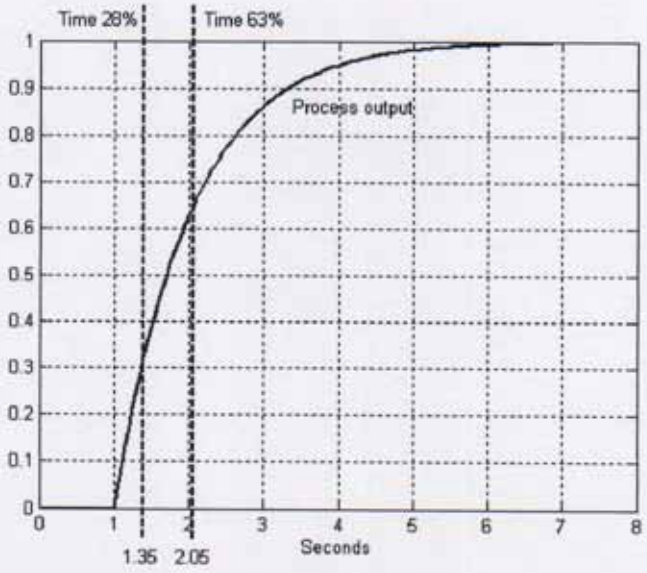

Figure (2.10) Simulated process in figure (2.9) open loop step response for two-point algorithm model identification method

The MATLAB commands in Appendix 2 section 1, page A2, entitled "OL_TP_1" determine the three first-order-plus-dead-time model parameters:

- Model gain, K_m, = 1.00
- Model time constant, τ_m, = 1.05 seconds
- Model time delay, d_m, = 1.00 seconds

These three first-order-plus-dead-time parameter values determined by the two-point method are now compared to the known parameter values and shown in figure (2.11).

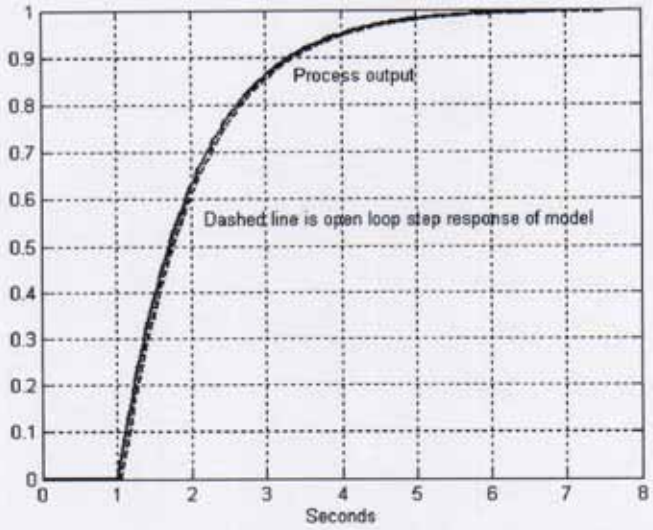

Figure (2.11) Two-point model output comparison with process output

Validation: The plot in figure (2.11) shows that the open loop step response of the model obtained using the two-point algorithm compares favourably with the open loop step response of the simulated process.

2.2.2.2 Implementation:

The two-point algorithm is applied to the step response in figure (2.7). The algorithm is entitled "OL_TP_2" in Appendix 2 section 1, page A2, and the results of the three parameters for the FOPDT model are subsequently shown.

- Model gain, K_m, = 1.15
- Model time constant, τ_m, = 0.53 seconds
- Model time delay, d_m, = 0.36 seconds

The three FOPDT process model parameters are inserted into a SIMULINK file and the model open loop step response is compared with the process trainer open loop step response (figure (2.12)).

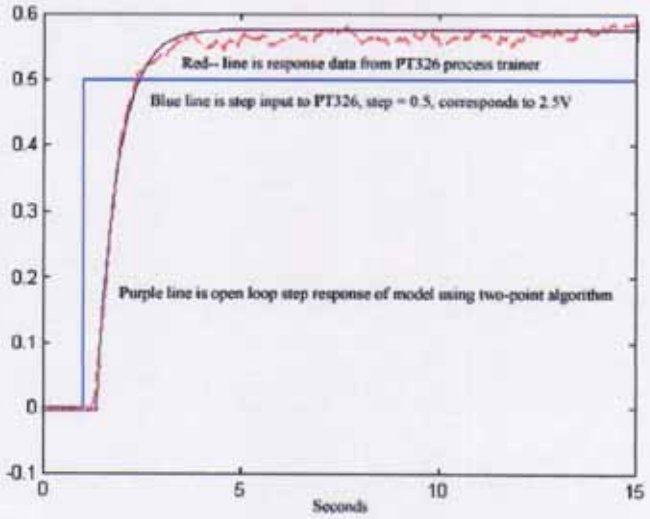

Figure (2.12) Open loop step responses of PT326 and FOPDT model using two-point algorithm identification technique

Validation: The result shown in figure (2.12) demonstrates that the open loop step response of the estimated FOPDT process model compares favourably with the open loop step response of the process. Good modelling of the process is confirmed in the "Comparison_FD" report in Appendix 2 section 3, page A63. Figure (TR_Id_6) shows the details.

2.2.3 Area Method

The third open loop time-domain method considered is the "area method" and is based on integrals of the step response (Astrom and Hagglund, 1995). The algorithm determines areas from the open loop step response data and from the resulting values, the time constant and the dead time of a FOPDT model are calculated. Figure (2.13) previews some details of the area method algorithm applied to the PT326 process trainer.

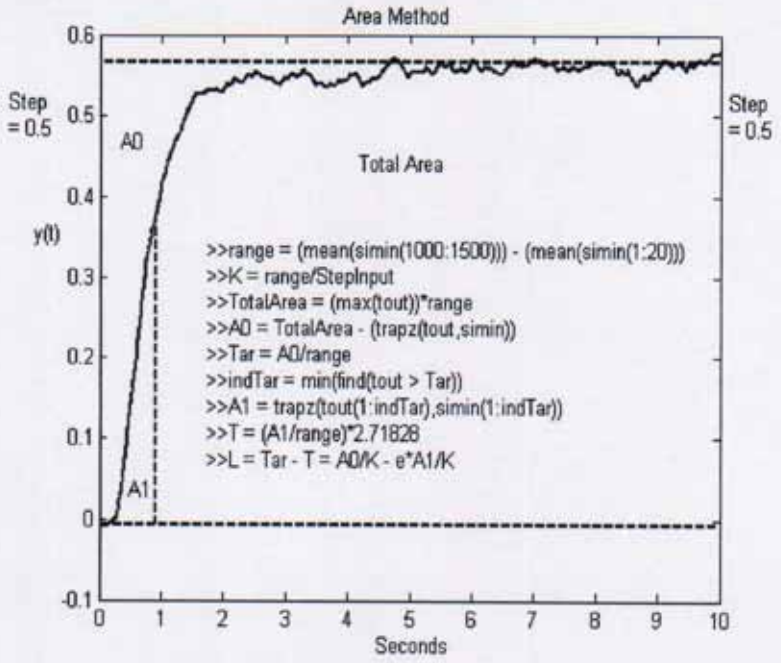

Figure (2.13) Plot of process open loop step response and area method algorithm

The commands in figure (2.13) are created using the MATLAB software with T = time constant (τ_m) and L = time delay (d_m).

The area methods are based on the calculation of areas associated with the step response. Figure (2.14) is a simulated system to demonstrate the method and figure (2.15) gives the system open loop step response with the area *A0* marked. There is a 1 second delay before the unit step input is applied to the process.

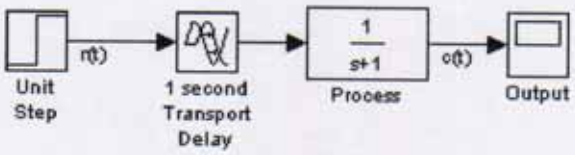

Figure (2.14) Simulated process under test for area method algorithm

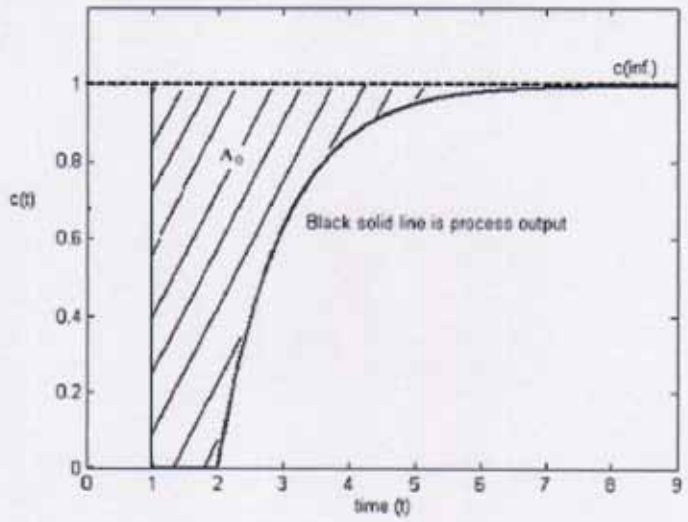

Figure (2.15) Open loop step response identifying area A_0

If the input is a unit step, and the model to be estimated is of the first-order-plus-dead-time form shown in equation (1.1), then

$c_\infty = K_m$

$c(t) = K_m(1 - e^{-(t-dm)/\tau m}).r(t)$

A_0 may easily be measured.

The "average residence time", T_{ar}, is defined as A_0/K_m. A_0 is determined as follows (Astrom and Hagglund (1995)):

Theoretically, $A_0 = \int_0^\infty [c_\infty - c(t)]dt$

i.e. $A_0 = \int_0^\infty \left[K_m - K_m\left(1 - e^{-(t-dm)/\tau m}\right)\right]dt$

$A_0 = K_m \int_0^\infty e^{-(t-dm)/\tau m} dt$

Therefore, $A_0 = K_m e^{dm/\tau m} \int_0^\infty e^{-t/\tau m} dt$

i.e. $A_0 = K_m e^{dm/\tau m} . e^{-t/\tau m} . (-\tau_m) \Big]_0^\infty$

i.e. $A_0 = -K_m \tau_m e^{dm/\tau m}[0-1]$

i.e. $A_0 = K_m \tau_m e^{dm/\tau m}$

So, $T_{ar} = \dfrac{A_0}{K_m} = \tau_m e^{dm/\tau m}$

T_{ar} may be approximated as $\tau_m \left(1 + \dfrac{d_m}{\tau_m} + \dfrac{1}{2}\left(\dfrac{d_m}{\tau_m}\right)^2 + \ldots\ldots\ldots\ldots \right)$

$\approx \tau_m \left(1 + \dfrac{d_m}{\tau_m}\right)$ if $\dfrac{d_m}{\tau_m}$ is small

Therefore $T_{ar} \approx \tau_m + d_m$

Astrom and Hagglund (1995) use the latter approximation for T_{ar}.

Figure (2.16) shows how area A_1 is determined (Astrom and Hagglund (1995)).

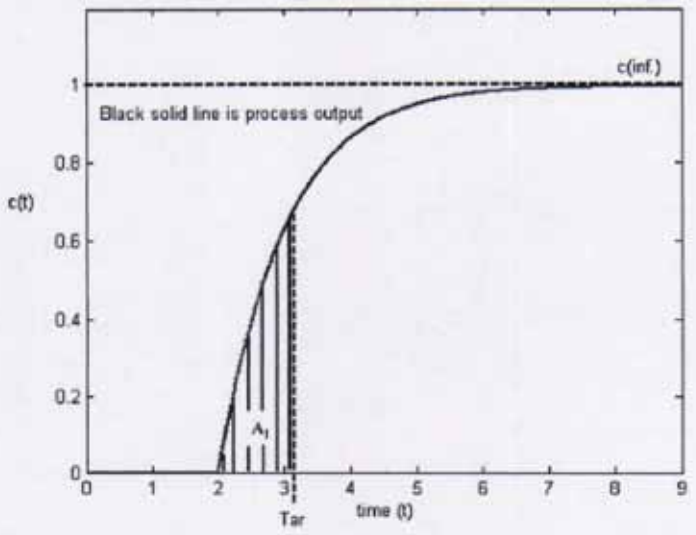

Figure (2.16) Open loop step response showing area A_1

If the area A_1 under the step response up to time T_{ar} is then determined,

$$A_1 = \int_0^{Tar} c(t)dt$$

Therefore $A_1 = \int_0^{Tar} K_m\left(1 - e^{-(t-dm)/\tau m}\right)dt$

i.e. $A_1 \cong \int_0^{dm} K_m\left(1 - e^{-(t-dm)/\tau m}\right)dt + \int_{dm}^{dm+\tau m} K_m\left(1 - e^{-(t-dm)/\tau m}\right)dt$

This expression is an approximation for A_1, as $T_{ar} \cong d_m + \tau_m$.

Therefore,

$$A_1 \cong \int_0^{\tau m} K_m\left(1 - e^{-t_1/\tau m}\right)d\,t_1, t_1 = t - d_m$$

i.e. $A_1 \cong K_m\left[t_1 + \tau_m e^{-t_1/\tau m}\right]_0^{\tau m}$

i.e. $A_1 \cong K_m\left[\tau_m + \tau_m e^{-1} - \tau_m\right] = \frac{K_m \tau_m}{e}$

Now, knowing K_m, and measuring A_1, allows us to determine τ_m:

$$\tau_m \cong \frac{e.A_1}{K_m}$$

Then $d_m \cong T_{ar} - \tau_m$

Therefore $d_m \cong \frac{A_0}{K_m} - \frac{e.A_1}{K_m}$

Some points about the area method algorithm

- The method is less sensitive to high frequency disturbances than methods where the model is determined from only a few values of the step response.
- However, the full step response needs to be stored (to calculate A_0), and the method (as described by Astrom and Hagglund (1995)) relies on d_m/τ_m being small, so that the approximation for T_{ar} is valid.

2.2.3.1 Simulation:

Figure (2.17) SIMULINK file with process parameters for area method algorithm

The file in figure (2.17) has a d_m/τ_m ratio of 1. It is worth noting that this may not be the best simulation to use to demonstrate the area method algorithm because of the approximations assumed earlier. The approximations were made on the assumption that the d_m/τ_m ratio is small, which is not the case here. However, for consistency, the simulated process used is the same throughout section 2.2.

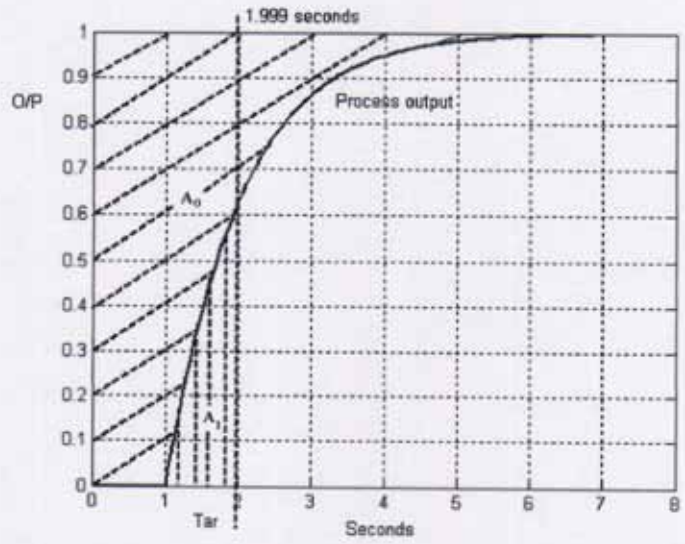

Figure (2.18) Process open loop step response for area method algorithm

The MATLAB commands in the "OL_Area_1" file in Appendix 2 section 1, page A2, determines the model parameters as follows:

- Model gain, K_m, = 1.00
- Model time constant, τ_m, = 0.99 seconds

- Approx_Time_Delay, d_m, = 1.01 seconds (using Astrom and Hagglund's approximation that $d_m = T_{ar} - \tau_m$)
- Time_Delay, d_m, = 0.70 seconds. (using a better approximation for the time delay as revealed in the development i.e. $T_{ar} = \tau_m e^{dm/\tau m}$. Note: This is a better approximation for the time delay, in general; an approximate formula is still used for the time constant).

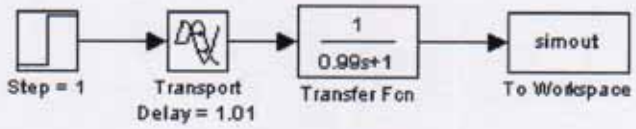

Figure (2.19) Area method process model with time delay approximation (Astrom and Hagglund (1995))

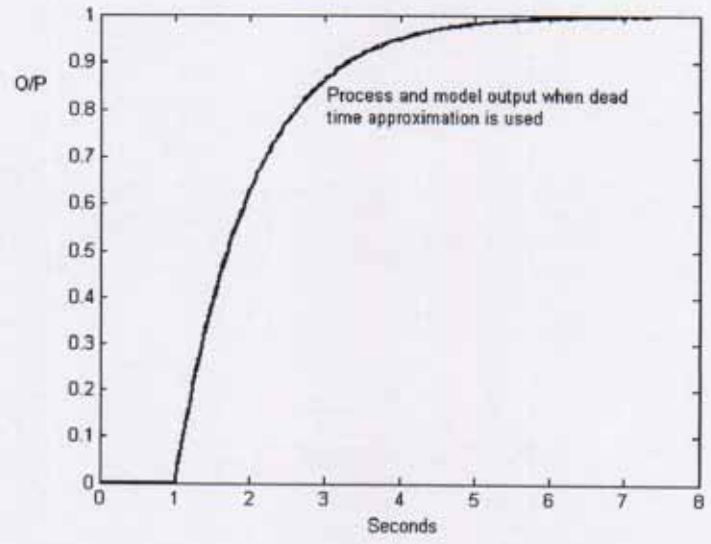

Figure (2.20) Comparison of process open loop step response with model open loop step response using area method and approximation for dead time (Astrom and Hagglund (1995))

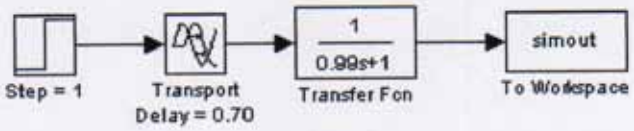

Figure (2.21) Area method process model with second time delay approximation

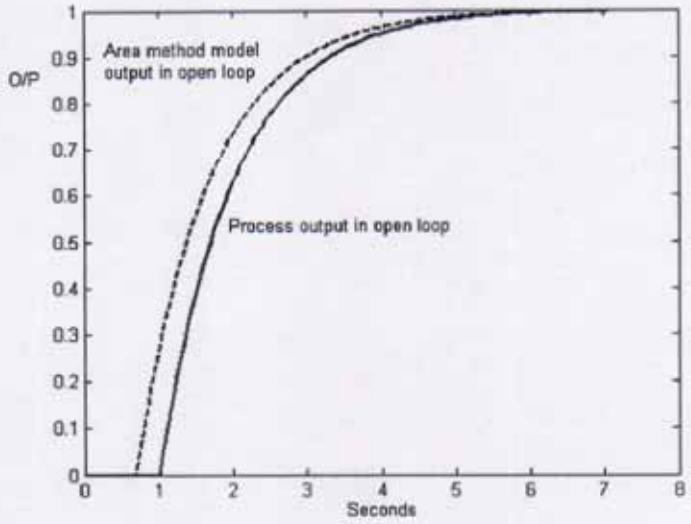

Figure (2.22) Comparison of process and model open loop step response using area method algorithm and second time delay approximation

Validation: The simulated process model gain estimate, K_m, and the simulated process model time constant estimate, τ_m, are both accurate estimations as shown in figures (2.20) and (2.22). The second time delay approximation for the process model is not as accurate as the first time delay approximation. This is not the expected result.

2.2.3.2 Implementation:

As a consequence of the results obtained in simulation, it was decided to consider the case in implementation, where the average residence time is approximately the sum of the dead time and the time constant, i.e. $T_{ar} \cong d_m + \tau_m$. The SIMULINK file in

figure (2.6) is again used, but now the step time is set to zero, i.e. no delay on the step input. The simulation is run and the response data is used by the area method algorithm. This algorithm calculates various areas from the data (figures (2.23) and (2.24)). The algorithm is shown in the file entitled "OL_Area_2" in Appendix 2 section 1, page A3.

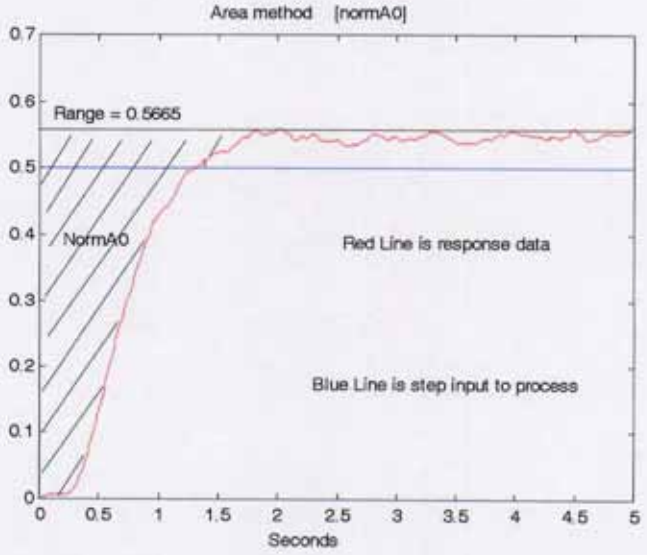

Figure (2.23) Area method showing $normA_0$

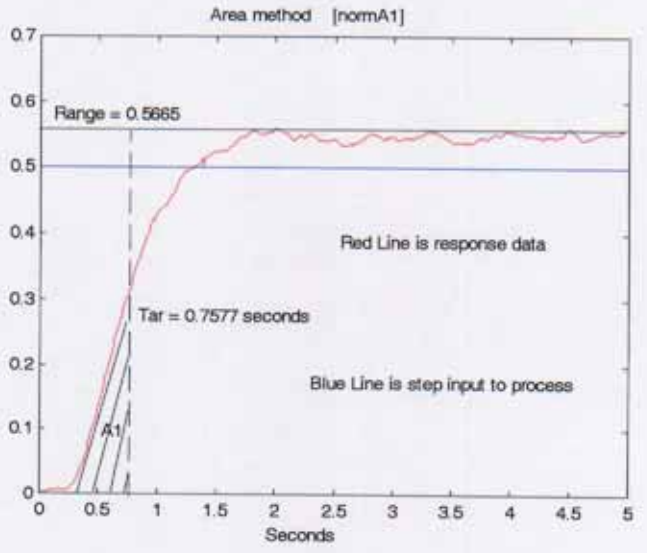

Figure (2.24) Area method showing A_1

The algorithm, of course, implements the less accurate approximation for the time delay (in general); the following results are obtained:

- Model gain, K_m, = 1.13
- Model time constant, τ_m, = 0.36 seconds
- Model time delay, d_m, = 0.40 seconds

These three FOPDT process model parameter values are inserted into a SIMULINK file and the open loop step response compared with the process open loop step response; the result is shown in figure (2.25).

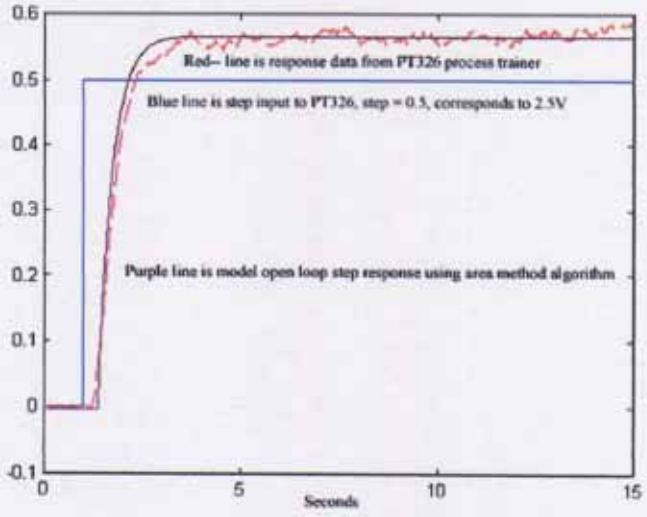

Figure (2.25) Comparison of PT326 and FOPDT model open loop step response using area method identification technique

Validation: The plot in figure (2.25) shows that, despite the approximation of the time delay detailed, the model open loop step response is a good representation of the process open loop step response. It is worth noting that all the validation comments are based on visual inspection of the process and model step responses. The accuracy of the model step responses could be more objectively evaluated by obtaining the sum of the squares of the errors between the process and model step responses, for instance.

2.2.4 Method of Moments

A drawback with the area method is that it requires storage of the step response. Area A_1 cannot be computed until area A_0 is determined. In addition, there are some process control systems where the dynamics contain integration or very long time constants. Such systems will not reach a steady state under open loop conditions. They are sometimes called systems without self-regulation. For a process with an integrator, a steady state will not be achieved when the input signal is a step, since the output will asymptotically change at a constant rate. There will be, however, a steady state output when the input is an impulse. To determine the dynamics we can, therefore, apply a short pulse to the process. This method is known as the "Method of Moments" (Astrom and Hagglund (1995)); the development outlined below is taken from their work.

Let *h(t)* be an impulse response and *G(s)* the corresponding transfer function. The functions are related through

$$G(s) = \int_0^\infty e^{-st} h(t)dt \tag{2.9}$$

The impulse response is positive for systems with monotone step responses. It can be interpreted as the density function of a probability distribution if it is normalised as follows:

$$f(t) = \frac{h(t)}{\int_0^\infty h(t)dt} \tag{2.10}$$

The quantity *f(t)dt* can then be interpreted as the probability that an impulse entering the system at time 0 will leave at time *t*. The average residence time is then

$$T_{ar} = \int_0^\infty tf(t)dt = \frac{\int_0^\infty th(t)dt}{\int_0^\infty h(t)dt} \tag{2.11}$$

We are considering the first order plus delay time model so using analytical methods, an expression for τ_m, the time constant, and d_m, the dead time, can be deduced by

assuming the average residence time, $T_{ar} \cong d_m + \tau_m$ as discussed in section 2.2.3. Then, from Astrom and Hagglund (1995),

$$\tau_m^2 = \frac{\int_0^\infty t^2 h(t)dt}{\int_0^\infty h(t)dt} - T_{ar}^2 \tag{2.12}$$

The time delay d_m can then be computed approximately, as follows:

$$d_m \cong T_{ar} - \tau_m \tag{2.13}$$

The gain K_m is given by the following equation:

$$K_m = G(0) = \int_0^\infty h(t)dt \tag{2.14}$$

2.2.4.1 Simulation:

A FOPDT process is now simulated in figure (2.26) and the response data analysed to determine a process model using the "Method of Moments" algorithm.

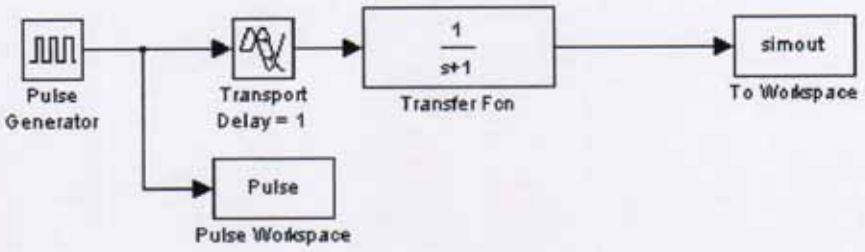

Figure (2.26) SIMULINK file with process for Method of Moments algorithm

As before, it is expected that a better simulation result would be achieved for a smaller d_m/τ_m ratio. However, for consistency with previous results, the simulated process in figure (2.26) is used. Also note that a further approximation is introduced by the use of a pulse input rather than an impulse input.

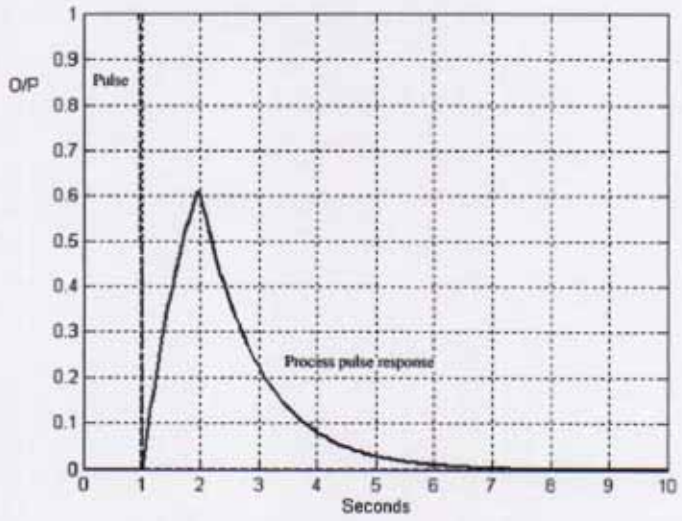

Figure (2.27) Pulse response of process

The file "OL_MoM_1" in Appendix 2 section 1, page A3, gives the MATLAB commands to obtain the following model parameters:

- Model gain, K_m, = 0.97
- Model time constant, τ_m, = 1.03 seconds
- Model time delay, d_m, = 1.46 seconds

These three first-order-plus-dead-time parameter values determined by the Method of Moments identification techniques are now compared to the known parameter values, through the open loop step responses shown in figure (2.29).

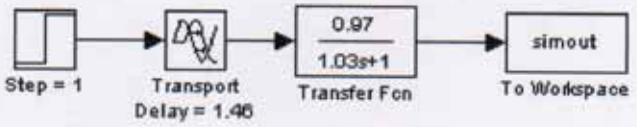

Figure (2.28) SIMULINK file with model parameters using Method of Moments

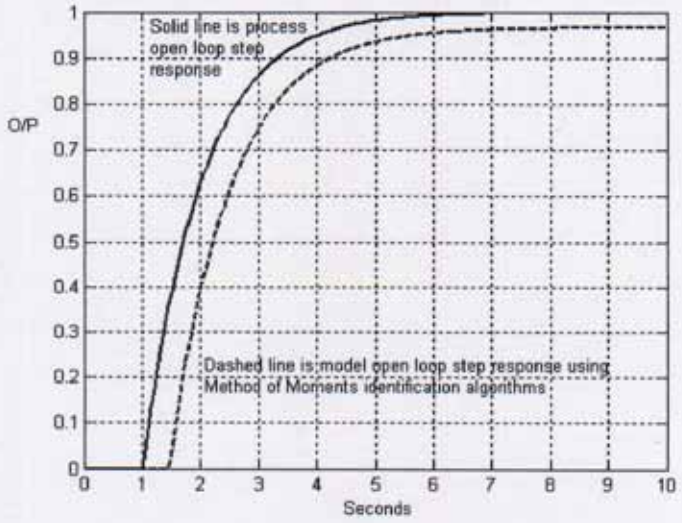

Figure (2.29) Comparison of process and model open loop step response

Validation: Figure (2.29) shows that the FOPDT model open loop step response is not close to the process open loop step response. One reason for this is the fact that the d_m/τ_m ratio is large and the approximations mentioned previously are less valid. Another problem is the difficulty in applying an impulse signal to the process in SIMULINK. Figure (2.29) demonstrates the large inaccuracy of the simulated process model time delay estimate, d_m.

2.2.4.2 Implementation:

A pulse is applied to the process trainer, PT326, using the SIMULINK file in figure (2.30) and algorithms based on equations (2.11), (2.12), (2.13) and (2.14) determine the three model parameters, process model gain, time constant and dead time. The area under the pulse signal is equal to 1, therefore the settings are as follows:

Period = 50 seconds, Duty cycle = 4%, Amplitude = 0.5, Start time = 0.

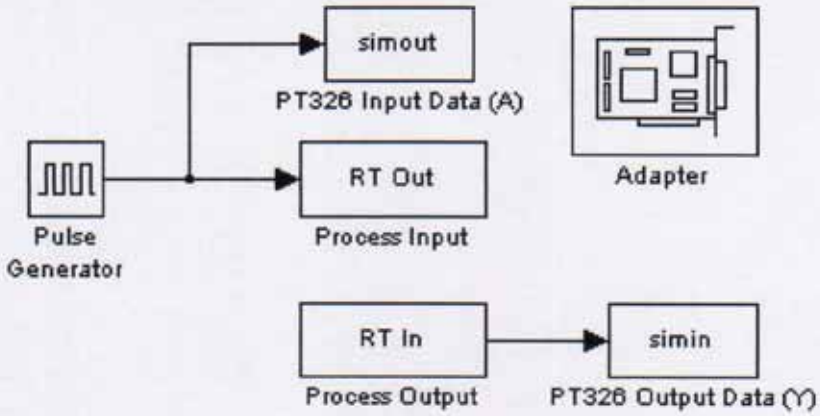

Figure (2.30) SIMULINK file used for Method of Moments

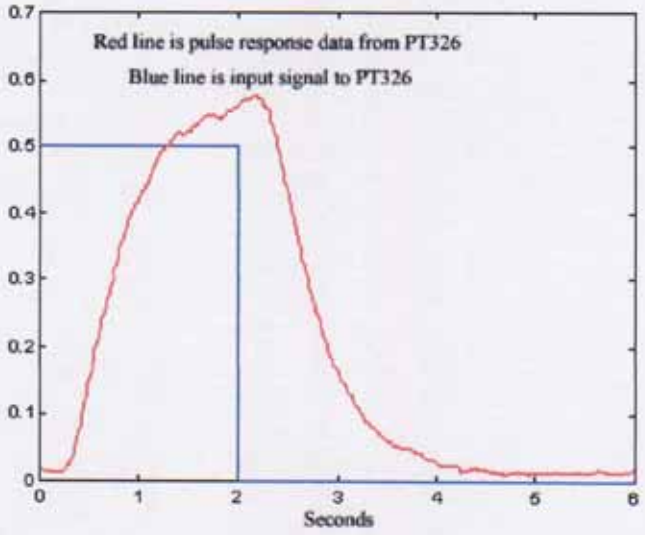

Figure (2.31) Method of Moments response data from PT326

The first order plus dead time model parameters are now determined from the algorithm entitled "OL_MoM_2" in Appendix 2 section 1, page A3, as follows:

- Model gain, K_m, = 1.15
- Model time constant, τ_m, = 0.69 seconds
- Model time delay, d_m, = 1.07 seconds

The FOPDT model parameters are inserted into a SIMULINK file and the open loop step response compared with the PT326 open loop step response in figure (2.32).

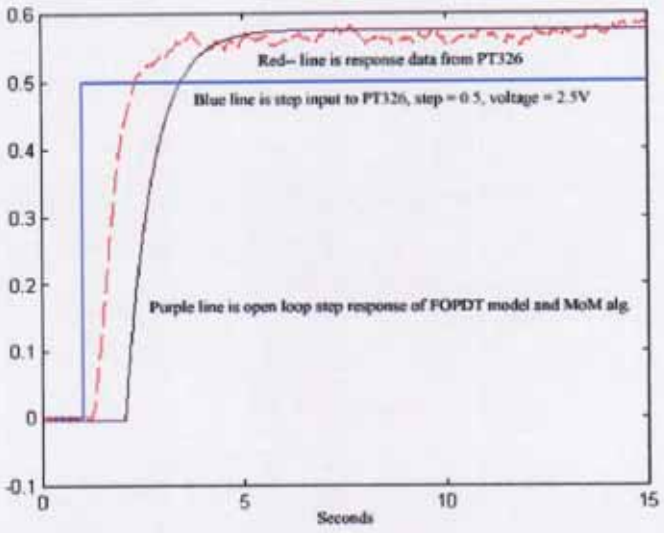

Figure (2.32) Comparison of PT326 and FOPDT model open loop step responses using the Method of Moments algorithm

Validation: Similar results to those obtained in simulation (figure (2.29)) are observed. The time delay estimation of the model, d_m, is inaccurate as can be seen from the plot in figure (2.32). The gain estimate, K_m, is quite accurate.

The area of the pulse input is equal to 1, i.e. amplitude of 0.5, for a period of 2 seconds. As a comparison, the pulse input will now be set to an amplitude of 1, for a period of 1 second and the results recorded as before. This pulse input is closer to an impulse function so better results are expected. The same SIMULINK file shown in figure (2.30) is used. The resulting plot is shown in figure (2.33).

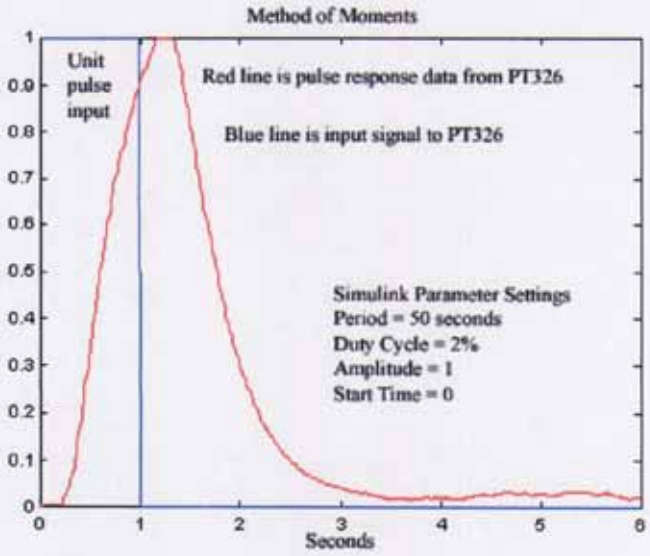

Figure (2.33) Pulse response of PT326

Using the resulting data, the "OL_Mom_3" algorithm in Appendix 2 section 1, page A3, is used to identify the three first order plus dead time model parameters of the process.

- Model gain, K_m, = 1.31
- Model time constant, τ_m, = 0.94 seconds
- Model time delay, d_m, = 0.56 seconds

When the results are compared to those obtained when the wider pulse input is used, it is seen that the parameter values differ depending on the pulse size and pulse time duration. For validation purposes, the three parameters are inserted into a SIMULINK file and the open loop step response of the process is compared with the model open loop step response (figure (2.34)).

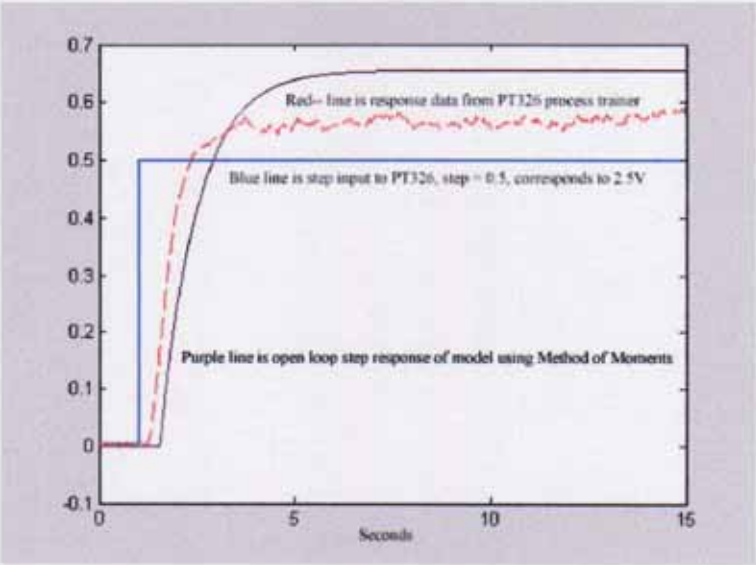

Figure (2.34) PT326 open loop step response and model open loop step response using the Method of Moments algorithm

Validation: The plot in figure (2.34) shows that the process model is inaccurate. The time delay estimation, d_m, is too large and the process model gain estimate, K_m, is also too high. When comparing the results of this test with the results from the wider pulse test, it can be seen that the time delay estimate is better using the narrower pulse settings. However, the gain estimate, in particular, is less accurate using the narrower pulse settings.

2.3 Frequency-Domain Modelling – Analytical and Gradient Approach

Identification in the frequency domain involves the estimation of the process frequency response over an appropriate frequency range, followed by the estimation of the model parameters. The process frequency response may be measured in open loop by recording the output of the process as a sine wave input varies in frequency and then determining the magnitude and phase from the input-output data at each frequency. The model parameters are estimated by a two-stage approach, combining an analytical approach and a gradient approach, as detailed by O'Dwyer (2002). The three parameters of the FOPDT model, equation (1.1), are analytically calculated as follows:

$$K_m = \frac{|G_p(j\omega_1)||G_p(j\omega_2)|\sqrt{\omega_2^2 - \omega_1^2}}{\sqrt{|G_p(j\omega_2)|^2 \omega_2^2 - |G_p(j\omega_1)|^2 \omega_1^2}} \tag{2.15}$$

$$\tau_m = \frac{1}{\omega}\sqrt{\frac{K_m^2}{|G_p(j\omega)|^2} - 1} \tag{2.16}$$

$$d_m = \frac{1}{\omega}\left[-\phi_p(j\omega) - \tan^{-1}(\omega\tau_m)\right] \tag{2.17}$$

ω_1 and ω_2 are two test frequencies; $|G_p(j\omega_1)|$ and $|G_p(j\omega_2)|$ are the magnitudes of the frequency response at ω_1 and ω_2 respectively; ϕ_p is the phase of the frequency response at test frequency ω.

The gradient approach is subsequently employed to determine the most accurate model parameters. The gradient method examines the cost function, J, to determine the best model estimate. The cost function, J, is the mean sum of the squares of the error between the process and the model outputs. An important requirement is that J must be unimodal i.e. J must have no local minima. The algorithm determines the partial derivative of the cost function, with respect to the three FOPDT parameters K_m, τ_m and d_m, at the initial estimate and subsequent estimates. The final and most

accurate estimated value, in a least squares sense, is in the trough of the cost function curve. Full details are available in the paper by O'Dwyer (2002).

The analytical and gradient approach is now examined in simulation and implementation. The simulated process is the same as the process used for the time-domain modelling and the PT326 process trainer is used for the implementation tests.

2.3.1 Simulation:

To determine the parameters of the FOPDT and SOPDT models using the frequency-domain methods, ten frequency domain experiments are carried out and the results recorded as shown in Table (2.1). The SIMULINK file in figure (2.35) is used to input sine-waves of varying frequencies to the simulated process and the resulting plot in figure (2.36) is an example of the data obtained from one such test:

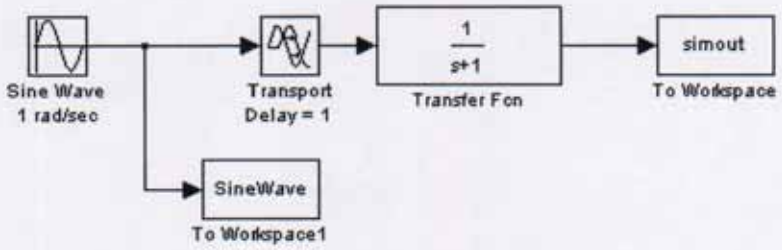

Figure (2.35) SIMULINK file to determine frequency response

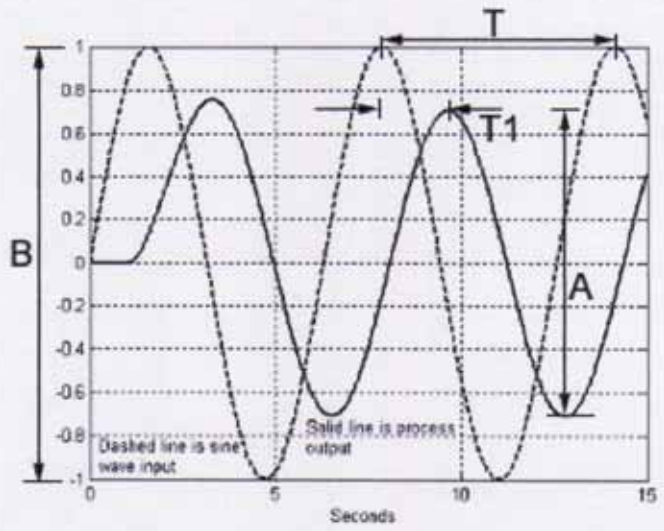

Figure (2.36) Input and output signals to process

From the data shown in figure (2.36), Magnitude $= \frac{A}{B}$, Phase $= -\frac{T1}{T} \times 2\pi$ radians.

Simulation step size = 0.05. This allows 20 samples to be recorded every second for the workspace data array. This is sufficient to plot the data accurately.

Frequency	Magnitude	Phase
Rads/sec		Radians
0.1	0.99	-0.20
0.2	0.98	-0.40
0.4	0.93	-0.77
0.5	0.90	-0.97
0.6	0.87	-1.14
0.8	0.81	-1.48
1	0.76	-1.86
2	0.45	-3.20
4	0.27	-5.40
5	0.20	-6.45

Table (2.1) Summary of results

2.3.1.1 FOPDT Model

The frequency, magnitude and phase values shown in Table (2.1) are next entered into the gradient method MATLAB program entitled "FreqGradFOPDT_Sim_1" in Appendix 2 section 1, pages A4 – A10. This program is the work of Dr. A. O'Dwyer (2002). For example, the first frequency value is entered as follows:

(ω_1) w1 = 0.1;

The corresponding magnitude value is then entered:

G1 = 0.99;

The corresponding phase value is entered:

phi1 = -0.20;

The remaining nine values are subsequently entered in the appropriate spaces. It is found that equations (2.15), (2.16) and (2.17) work reasonably well if (O'Dwyer, 2002):

1. K_m is determined from magnitude data recorded at least a decade apart in frequency.
2. τ_m is determined when the magnitude of the process, $|G_p(j\omega)|$, is in a range of 0.25 to 0.75 times the gain, K_m, calculated.
3. d_m is determined when the magnitude of the process is less than 0.5 times the gain, K_m, calculated.

Results:

- (Model gain, K_m) Kpavg = 0.999
- (Model time constant, τ_m) Tcpavg = 0.95
- (Model time delay, d_m) Tdpavg = 1.03

The first-order-plus-dead-time model parameters are now used to generate a nyquist plot for comparison with the simulated process; this plot is shown in figure (2.37). The MATLAB code to generate figure (2.37) is shown in Appendix 2 section 1, page A10, and is entitled "FD_1A".

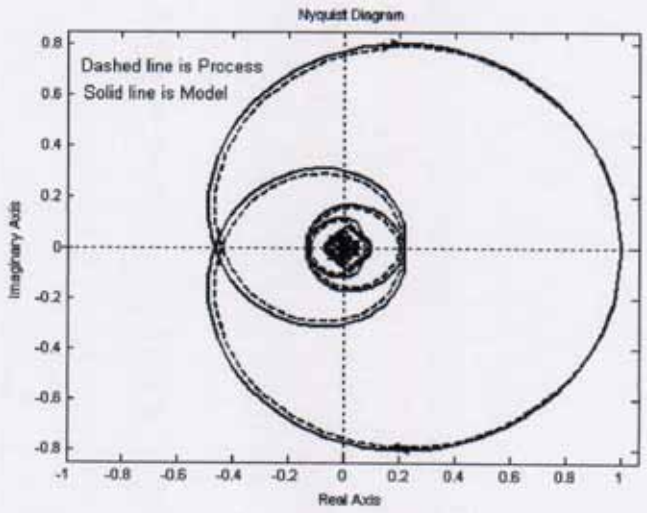

Figure (2.37) Nyquist plot of simulated process and FOPDT model

Validation: The nyquist plot of the process and the FOPDT model in figure (2.37) indicate that the modelling method works well. The low frequency simulated process gain estimate, K_m, is very accurate.

2.3.1.2 SOPDT Model

The values from Table (2.1) are entered, in the same way as for the FOPDT model, into the gradient method MATLAB program entitled "FreqGradSOPDT_Sim_1" in Appendix 2 section 1, pages A11 – A22, to obtain the second-order-plus-dead-time model parameters.

Results:

- (Model gain, K_m) ans = 1.01
- (a_{m1}) ans = 0.98
- (a_{m2}) ans = 3.28e-004
- (Model time delay, d_m) ans = 1.03

The second-order-plus-dead-time model structure is shown in equation (2.18).

$$G_m(s) = \frac{K_m e^{-sdm}}{1 + a_{m1}s + a_{m2}s^2} \tag{2.18}$$

The SOPDT model is therefore

$$G_m(s) = \frac{1.01e^{-1.03s}}{1 + 0.98s + 0.000328s^2}$$

The second-order-plus-dead-time model parameters are now used to generate a nyquist plot for comparison with the parameters of the process and shown in figure (2.38). The MATLAB code is shown in "FD_2A" in Appendix 2 section 1, page A22.

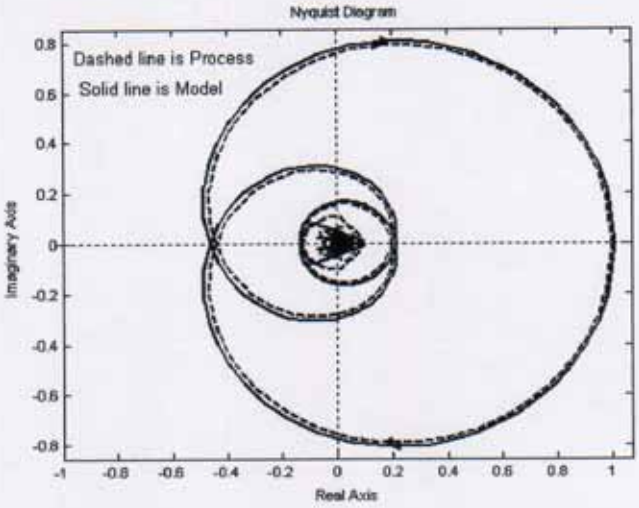

Figure (2.38) Nyquist plot of simulated process and SOPDT model

Validation: As with the FOPDT model, the plot in figure (2.38) validates the modelling technique. The "fit" between process and SOPDT model is very close. The SOPDT model program takes a longer time to execute than the FOPDT model program. The initial values of the estimated parameters are determined by averaging the values found in a purely analytical approach. The final and best model parameter estimates are then calculated using the gradient approach.

2.3.2 Implementation:

The frequency response of the Process Trainer is obtained by transmitting sine waves of constant magnitude and varying frequency to the input of the process, and plotting the output from the process. The SIMULINK file to achieve this is shown in figure (2.39).

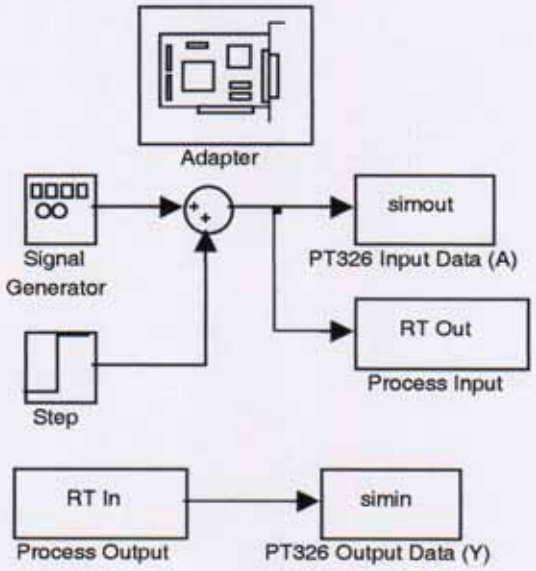

Figure (2.39) SIMULINK file used to help to obtain the frequency response of PT326

The step size in figure (2.39) is set to 0.3. This corresponds to a voltage of 1.5 Volts. This is the offset to the process that ensures that no clipping of the signal takes place. The signal generator output is set to a sine wave of amplitude 0.25 with the frequency, in radians/second, varying between 0 radians/second to 20 radians/second. Figure (2.40) shows the input and output data recorded, as an example, when the input frequency is set to 4 radians/second.

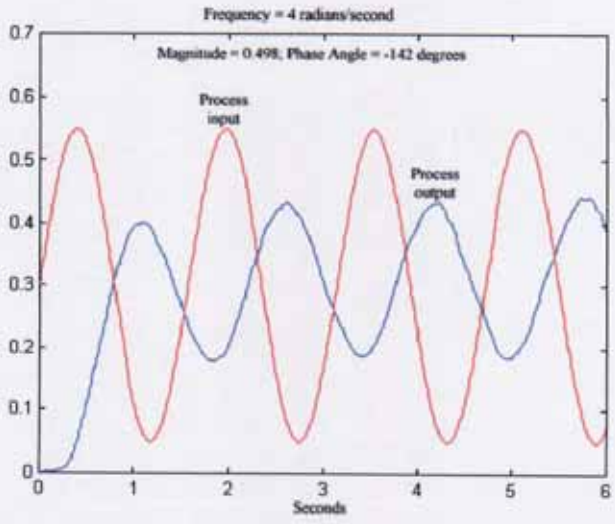

Figure (2.40) Sine wave response of PT326 when input signal is 4 radians/second

Thirty-five different frequencies are examined between 0 and 20 radians/second. Table (2.2) shows the results of these experiments.

Frequency (Rads/Sec)	Freq. (Hertz)	1/Freq. (Time,sec.)	Magnitude (Op/Ip)	Magnitude (dB)	Phase (Deg.)	Phase (Rads.)	Point (On plot)
0	0		1.17	1.39	0	0.00	1
0.1	0.02	62.89	1.14	1.13	-2.3	-0.04	2
0.2	0.03	31.44	1.13	1.04	-4.6	-0.08	3
0.3	0.05	20.96	1.11	0.88	-6	-0.10	4
0.4	0.06	15.72	1.10	0.83	-18	-0.31	5
0.5	0.08	12.57	1.12	0.96	-23	-0.40	6
0.6	0.10	10.47	1.09	0.71	-30	-0.52	7
0.7	0.11	8.97	1.05	0.46	-31	-0.54	8
0.8	0.13	7.85	1.08	0.70	-33	-0.57	9
0.9	0.14	6.98	1.05	0.39	-37	-0.64	10
1	0.16	6.28	1.07	0.55	-45	-0.78	11
1.3	0.20	5.02	1.00	0.01	-57	-0.99	12
1.5	0.24	4.19	1.00	0.03	-61	-1.06	13
2	0.32	3.14	0.84	-1.52	-83	-1.44	14
2.5	0.40	2.51	0.72	-2.89	-100	-1.74	15
3	0.48	2.09	0.67	-3.50	-117	-2.04	16
3.5	0.56	1.79	0.57	-4.94	-136	-2.37	17
4	0.64	1.57	0.50	-6.05	-142	-2.48	18
4.5	0.72	1.39	0.40	-7.91	-171	-2.98	19
5	0.80	1.25	0.37	-8.73	-177	-3.09	20
5.3	0.84	1.19	0.39	-8.14	-180	-3.14	21
5.5	0.88	1.14	0.34	-9.42	-189	-3.29	22
6	0.96	1.04	0.32	-12.32	-199	-3.47	23
6.5	1.04	0.96	0.28	-10.96	-202	-3.53	24
7	1.11	0.89	0.24	-12.32	-216	-3.77	25
7.5	1.19	0.83	0.22	-12.97	-223	-3.89	26
8	1.37	0.72	0.21	-13.63	-229	-3.99	27
8.5	1.35	0.74	0.19	-14.32	-233	-4.06	28
9	1.43	0.69	0.17	-15.15	-258	-4.51	29
9.5	1.51	0.66	0.15	-16.57	-261	-4.55	30
10	1.59	0.62	0.14	-17.22	-286	-4.99	31
12.5	1.99	0.51	0.08	-21.51	-300	-5.24	32
15	2.39	0.42	0.06	-24.79	-320	-5.58	33
17.5	2.79	0.36	0.05	-26.57	-360	-6.28	34
20	3.18	0.31	0.04	-27.72	-412	-7.19	35

Table (2.2) Experimental data results for frequency response of PT326

The results in table (2.2) enable the Process Trainer nyquist and bode plots to be drawn (See figures (2.41) and (2.42)). The MATLAB commands to draw the nyquist and bode plots from experimental data are in Appendix 2 section 1, page A22, entitled "PT326_Ny_1" and "PT326_Bode_1" respectively.

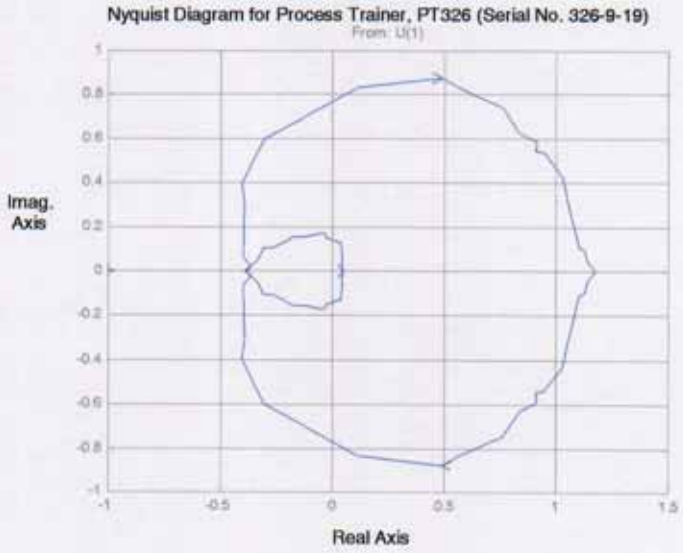

Figure (2.41) Nyquist plot for PT326 Process Trainer using data in Table (2.2)

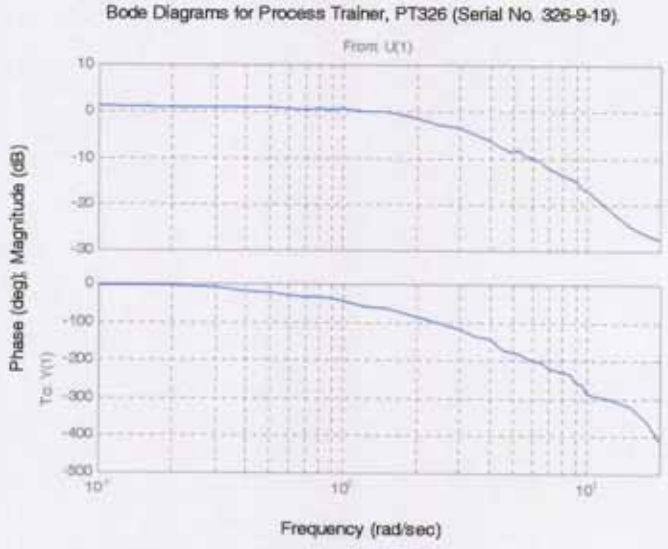

Figure (2.42) Bode plot for PT326 Process Trainer using data in Table (2.2)

2.3.2.1 FOPDT Model

The program by O'Dwyer (2002) to identify a FOPDT process model requires ten values of frequencies and the corresponding magnitude and phase data to be entered at the start of the program. Therefore, ten appropriate values from Table (2.2) are chosen for this identification procedure. The ten frequency values chosen are as follows:

$\omega_1 = 0.4$ rads/sec. $\omega_2 = 0.6$ rads/sec. $\omega_3 = 0.9$ rads/sec. $\omega_4 = 1.25$ rads/sec.

$\omega_5 = 2.5$ rads/sec. $\omega_6 = 4$ rads/sec. $\omega_7 = 5$ rads/sec. $\omega_8 = 6$ rads/sec.

$\omega_9 = 9$ rads/sec. $\omega_{10} = 12.5$ rads/sec.

The corresponding magnitude and phase data is then taken from Table (2.2) and the program is then executed.

Results:

>> FreqGradFOPDT_Sim_1

- (Model gain, K_m) Kpavg = 1.13
- (Model time constant, τ_m) Tcpavg = 0.61 sec.
- (Model time delay, d_m) Tdpavg = 0.34 sec.

These three FOPDT process model parameters are now put in transfer function form, equation (1.1), to generate a model nyquist plot to compare the trajectory with the process nyquist plot; the results are shown in figure (2.43). The MATLAB commands to draw a nyquist or bode plot from frequency response data is shown in a file entitled "PT326_Nyquist_Bode_Plot" in Appendix 2 section 1, page A22.

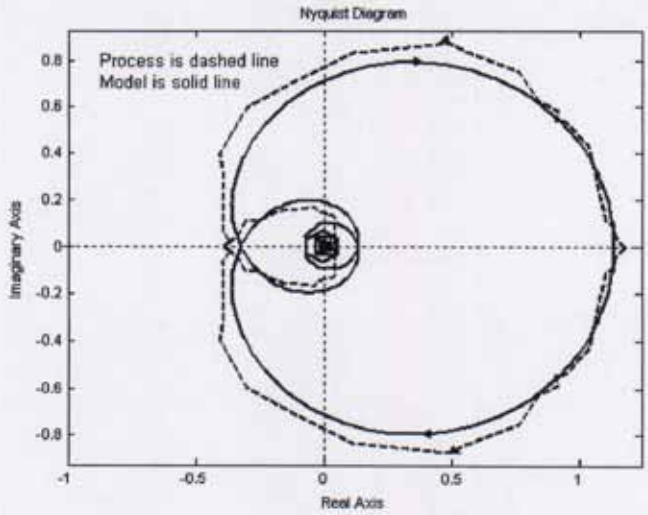

Figure (2.43) Nyquist plot of process (dashed line) and process model (solid line)

Validation: The plot in figure (2.43) indicates that the FOPDT model is a good model of the PT326 process trainer. The low frequency gain is particularly accurate.

2.3.2.2 SOPDT Model

The SOPDT model structure is shown in equation (2.18).

$$G_m(s) = \frac{K_m e^{-sdm}}{1 + a_{m1}s + a_{m2}s^2} \tag{2.18}$$

The two-stage approach, combining an analytical and gradient method, is also used to obtain the parameters of a SOPDT model. The ten frequency values that were entered for the FOPDT model program from Table (2.2) are now entered into the SOPDT program written by O'Dwyer (2002). The program is entitled "FreqGradSOPDT_Sim_1" in Appendix 2 section 1, pages A11 – A22.

Results:

>> FreqGradSOPDT_Sim_1

- ans = 1.13 (K_m)
- ans = 0.5704 (a_{m1})
- ans = 0.0775 (a_{m2})

- ans = 0.23 (d_m)

The process SOPDT model transfer function now becomes:

$$G_m(s) = \frac{1.13e^{-0.23s}}{0.0775s^2 + 0.5704s + 1} \tag{2.19}$$

A different SOPDT model structure is shown in equation (1.3),

$$G_m(s) = \frac{K_m e^{-d_m s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{1.3}$$

As an extra piece of information, the two time constants in equation (1.3), τ_1 and τ_2, can be deduced from equation (2.19) using MATLAB. See the program "SOPDT_Roots" in Appendix 2 section 1, page A23. This program gives the following results:

$\tau_1 = 0.22$ seconds

$\tau_2 = 0.35$ seconds

The SOPDT model transfer function in equation (2.19) is used to plot the nyquist and bode plots of the model on the same plots as the nyquist and bode plot of the PT326 process trainer, to validate the model. The resulting plots are shown in figure (2.44) and figure (2.45). The MATLAB commands to draw the bode plot in figure (2.45) are in "SOPDT_FD_Bode" in Appendix 2 section 1, page A23.

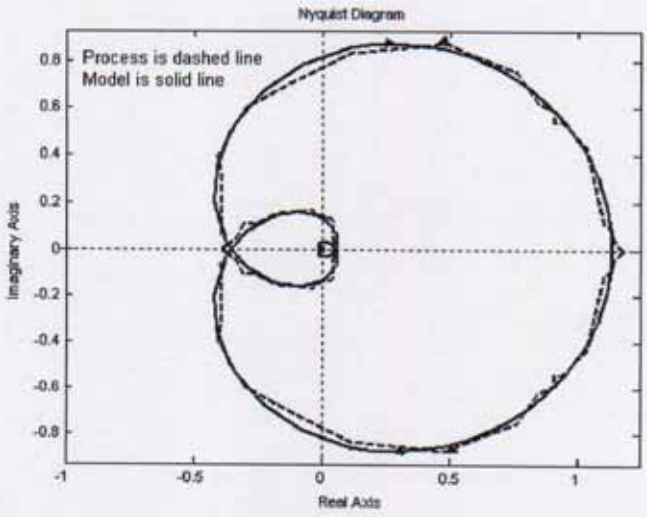

Figure (2.44) Nyquist plot of PT326 (dashed line) with SOPDT model (solid line)

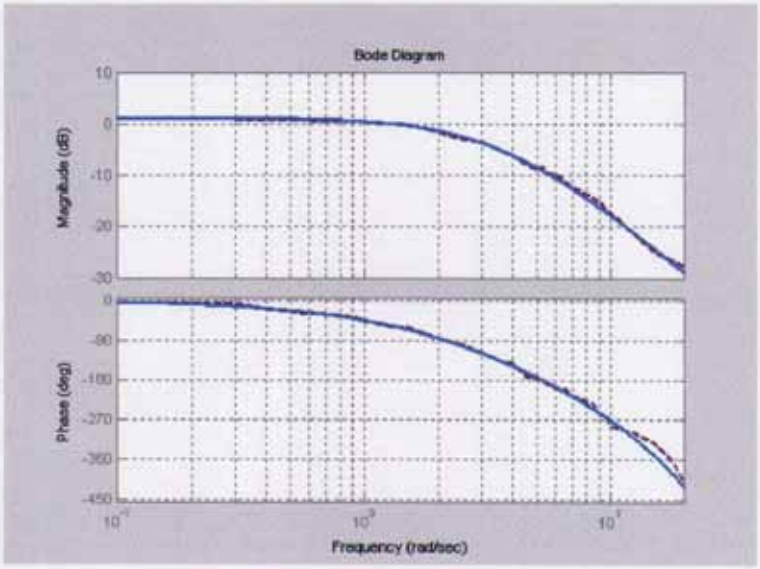

Figure (2.45) Bode plot of PT326 process trainer (Black dashed line) and SOPDT process model (Blue solid line)

Validation: The plots in figures (2.44) and (2.45) show that the SOPDT process model provides very good fitting of the PT326 process trainer. This is true even at high frequencies. The model gain estimate is the same as the process gain. The result proves that the method works very well in identifying a SOPDT model for a process with a time delay. This result is confirmed in a conference paper by Kealy and O'Dwyer (2003b) that compares model identification techniques on a real process. This paper can be viewed in Appendix 2 section 6.4, page A195.

2.4 Case Study – Feedback Procon pH Rig

The pH measurement variable is expressly designed to report the activity of hydrogen ions in an aqueous solution. The true concentration of the hydrogen ions may differ from its measured activity at pH levels below 2, where ion mobility may be impeded. Most pH loops have control points in the pH 2 – 12 range, however, where activity and concentration are essentially identical. Consequently, the pH measurement will be used herein as an indication of the concentration of hydrogen ions [H^+] in solution. The pH loop has been generally regarded as the most difficult single loop in process

control, for many reasons. First, the response of pH to reagent addition tends to be non-linear. Second, the sensitivity of pH to reagent addition in the vicinity of the set-point tends to be extreme, in that a change of one pH unit can result from a fraction of a percent change in reagent addition. Thirdly, the two relationships above are often subject to change, especially when treating wastewater. And finally, reagent flow requirements may vary over a range of 1000:1 or more, especially when treating wastewater (Shinskey, 1988).

The pH rig used in the experiments is the Procon pH Process Control Trainer, 38 Series, from Feedback Instruments Limited shown in figure (2.46). The effluent to be treated is stored in the effluent holding tank and fed to a circulating pump. The liquid passes through a manual flow control valve and variable flow-meter before entering the reaction vessel via an inlet. After being mixed with reagent, the liquid exits through an overflow to a treated fluid tank. The pH of the liquid in the process vessel is monitored using the pH probe. The reagent is stored in the reagent holding tank and fed via an inlet to the circulating pump. The liquid passes through a manual flow control valve, variable area flow-meter, solenoid valve and servo valve before entering the reaction vessel.

What is recognised as the non-linearity in pH control loops is the process titration curve, the relationship between measured pH and the amount of acid or base reagent added to a solution. The reagents in the experiments are acids or bases of known concentration, pH4 and pH10, added to the solution in the reaction vessel (see figure (2.46)). The pH measuring electrode and associated instrumentation is calibrated to give an output of 4mA when immersed into a pH4 (acid) solution and 20mA when immersed into a pH10 (base) solution.

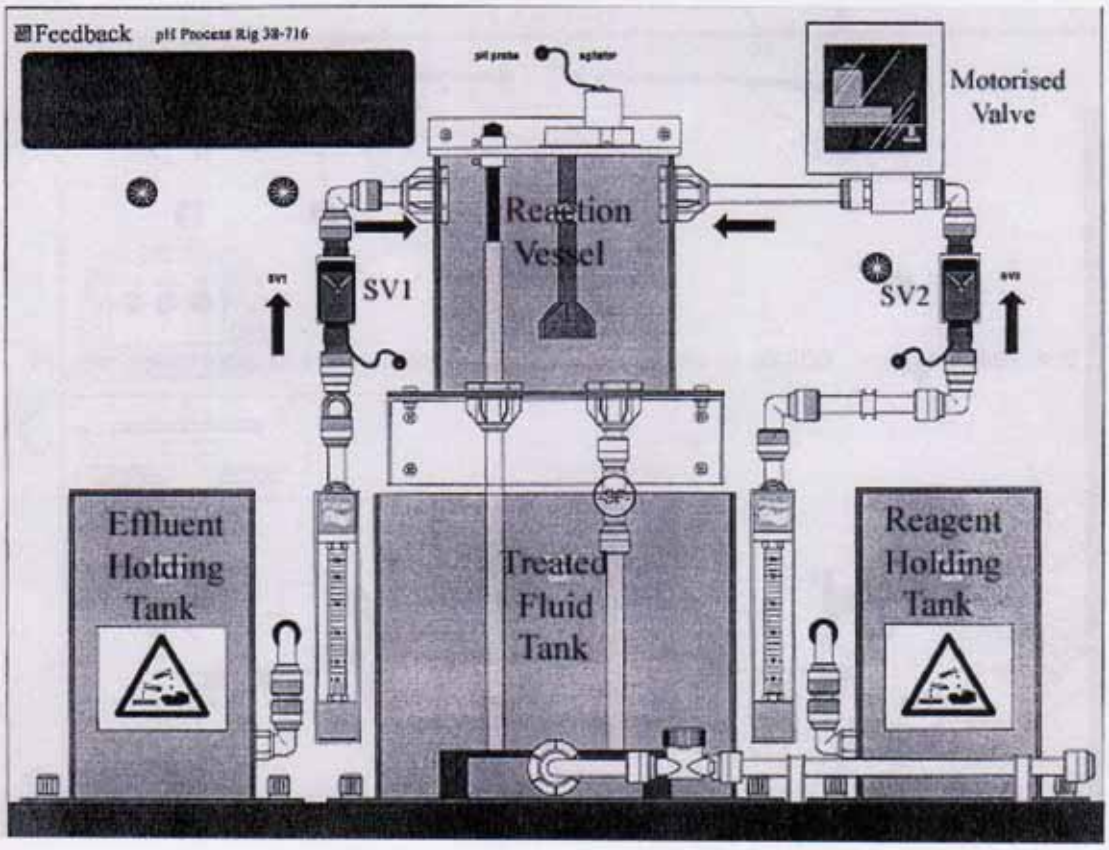

Figure (2.46) pH Process Rig

The titrations are conducted batch-wise by adding reagent incrementally to a measured volume of process solution in the reaction vessel. The first titration curve is generated by depositing 40ml of pH10 solution and zero pH4 solution into the reaction vessel. Increments of pH4 solution are subsequently added and the pH value recorded when the pH value settles down. The result of this experiment is shown in figure (2.47). The data for generating "Titration Curve 1" is shown in Appendix 2 section 2, page A25, in Table (TC_1).

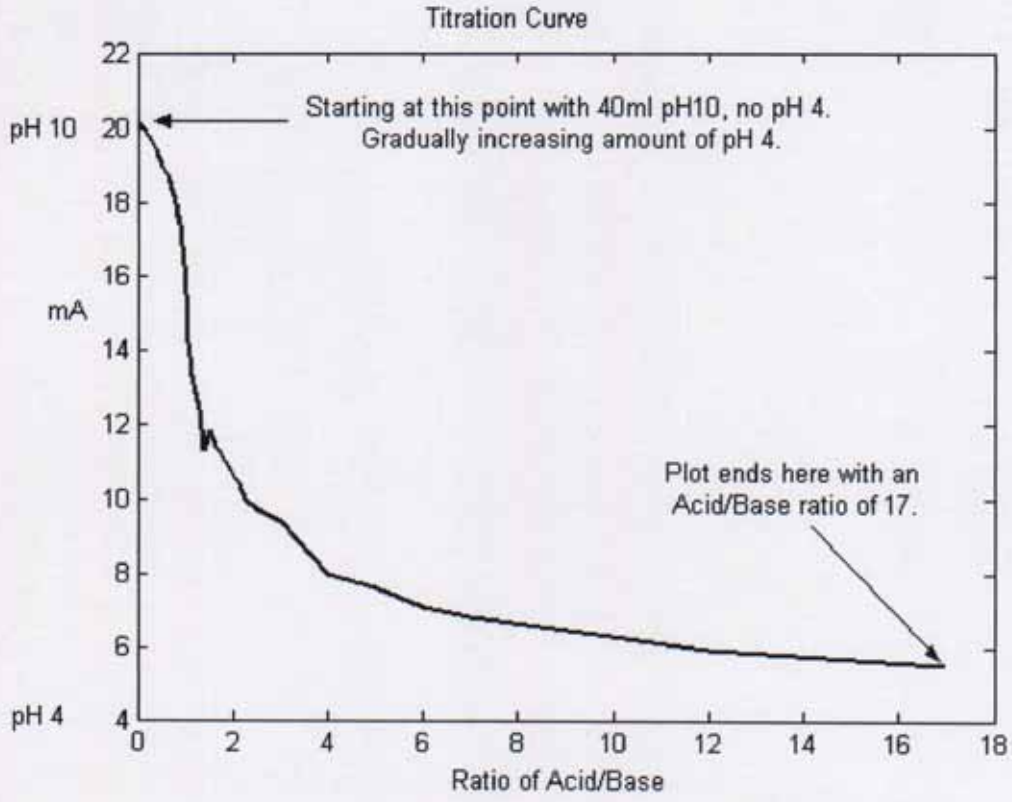

Figure (2.47) Titration Curve 1

The second titration curve is generated by depositing 40ml of pH4 solution and zero pH10 solution in the reaction vessel. Increments of pH10 are then subsequently added and the pH recorded. The result of this experiment is shown in figure (2.48). The data for generating "Titration Curve 2" is shown in Appendix 2 section 2, page A26, in Table (TC_2). Both titration curves are plotted on the same plot and shown in figure (A2.1) on page A27 in Appendix 2.

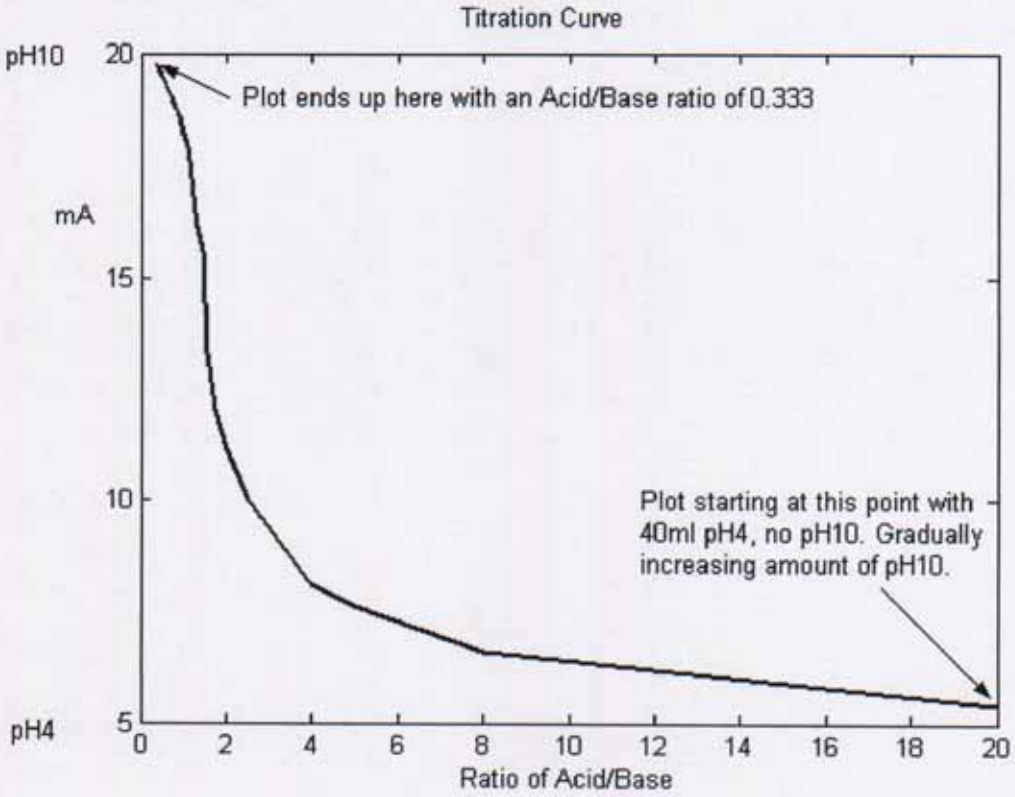

Figure (2.48) Titration Curve 2

The titration curves convert the acid/base ratio on the input to the mA output in a non-linear fashion. The titration curves are used in the calculation of the process model gain parameter, K_m, in the identification stage. When the reagent to be added to the solution has a lower pH than the one measured in the vessel, Titration Curve 1 is used, otherwise, Titration Curve 2 is used. Because of the non-linear characteristics of the pH process, the process model identification stage takes on a different path than the ones previously described in sections 2.2 and 2.3. It is decided that the pH process is "piece-wise" linear. This means that the process is linear within small ranges in its overall operating range. Eight experiments are carried out, encompassing different pH starting values of the solution, different volumes of reagent added to the solution and higher/lower pH values of reagent added to the solution, to obtain the dynamic model parameters of the process under eight different operating regimes.

The experiments are carried out by (1) measuring the pH of the solution in the reaction vessel (2) adding a known quantity and value of pH buffer to the solution (3) recording the pH output of the solution versus time, as the buffer is added in a batch manner. The recorded data output is then used to obtain FOPDT models of the process. The models are determined using three of the open loop identification techniques discussed previously, namely

- Two-point algorithm
- Graphical method
- Area method

An equaliser (characteriser), which uses the inverse of the titration curve, has not been used in this case to linearise the titration curve.

The gain of the process, at each of the eight operating conditions, is calculated as follows, using the second entry in Tables (2.3), (2.4) and (2.5) as an example. A SIMULINK block diagram of the overall system is shown in figure (2.49).

- There is 850ml of pH7 solution to start with in the tank. At a pH of 7, the acid/base ratio is 1. From titration curve 1 in figure (2.47), when the acid/base ratio is 1, the current output is 14mA.
- Next, 850ml of pH4 solution is added. The acid/base ratio is now (2/1) = 2. This is based on the assumption that we start one volume of pH4 solution and one volume of pH10 solution, giving one volume of pH7 solution. Next, one volume of pH4 solution is added, hence the 2:1 ratio.
- The current output is then determined as 10.6mA from figure (2.47). The current output difference is (14 – 10.6)mA = 3.4mA. The non-linear gain is therefore (3.4/2) = 1.7 (figure (2.49)).
- The change in pH due to the addition of 850ml of pH4 solution is 1.11, as measured by the pH sensor. The pH rig gain is (pH_Change/Acid:Base_Ratio) = (1.11/2) = 0.56 (figure (2.49)).
- The process gain is then determined as (0.56/1.7) = 0.33.

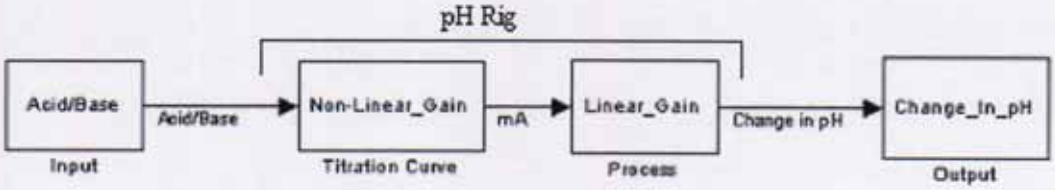

Figure (2.49) Block Diagram for Calculating Process Model Gain

The process model time delay and time constant estimated values are calculated in the ways previously described in sections 2.2.1, 2.2.2 and 2.2.3. These experiments are carried out using the "Graphical", "Two-Point Algorithm" and "Area" methods of process model identification.

The results of these twenty-four experiments are plotted in Appendix 2, section 2, Technical Report pH Rig, on pages A24 – A51. The results are summarized in Tables (2.3), (2.4) and (2.5). The experiments were performed using MATLAB/SIMULINK and HUMUSOFT data acquisition software together with the required hardware described in chapter 1. The sampling time was 0.05 seconds. In Tables (2.3), (2.4) and (2.5), Km = K_m, process model gain estimate; dm = d_m, process model time delay, and tm = τ_m, process model time constant. The ratio of time delay to time constant, d_m/τ_m, is also given in the last column.

		2-Pt Algo.	2-Pt Algo.	2-Pt Algo.	dm/tm
Starting	Adding	Km	dm	tm	Ratio
850ml pH7	850ml pH10	1.50	1.88	1.13	1.67
850ml pH7	850ml pH4	0.33	6.40	1.50	4.27
1670ml pH7	850ml pH10	0.46	2.30	2.10	1.10
1670ml pH7	850ml pH4	0.16	1.28	0.68	1.89
850ml pH8.3	850ml pH10	0.35	2.85	1.20	2.38
850ml pH5.6	850ml pH4	0.33	3.68	1.88	1.96
850ml pH5.5	850ml pH10	0.30	1.93	0.83	2.33
850ml pH8.1	850ml pH4	0.25	1.73	0.53	3.29

Table (2.3) Two-point algorithm models

Starting	Adding	Graphical Km	Graphical dm	Graphical tm	dm/tm Ratio
850ml pH7	850ml pH10	1.50	2.10	1.55	1.35
850ml pH7	850ml pH4	0.33	7.40	1.80	4.11
1670ml pH7	850ml pH10	0.46	3.60	1.80	2.00
1670ml pH7	850ml pH4	0.16	2.30	0.65	3.54
850ml pH8.3	850ml pH10	0.35	3.50	1.55	2.26
850ml pH5.6	850ml pH4	0.33	4.75	1.80	2.64
850ml pH5.5	850ml pH10	0.30	2.90	0.85	3.41
850ml pH8.1	850ml pH4	0.25	2.70	0.55	4.91

Table (2.4) Graphical method models

Starting	Adding	Area Km	dm	tm	dm/tm Ratio
850ml pH7	850ml pH10	1.50	1.61	5.42	0.30
850ml pH7	850ml pH4	0.33	5.27	6.66	0.79
1670ml pH7	850ml pH10	0.46	2.71	3.55	0.76
1670ml pH7	850ml pH4	0.16	2.97	3.98	0.75
850ml pH8.3	850ml pH10	0.35	1.84	2.40	0.77
850ml pH5.6	850ml pH4	0.33	2.80	3.92	0.71
850ml pH5.5	850ml pH10	0.30	1.53	2.91	0.53
850ml pH8.1	850ml pH4	0.25	1.93	2.20	0.88

Table (2.5) Area method models

The results shown in Tables (2.3), (2.4) and (2.5) indicate that there are different model gains, model time delays and model time constants for different operating conditions on the pH rig. Therefore, it would not be realistic to identify just one model for this process. A representative example of the results obtained is demonstrated in figure (2.50) showing that the identification algorithms worked reasonably well.

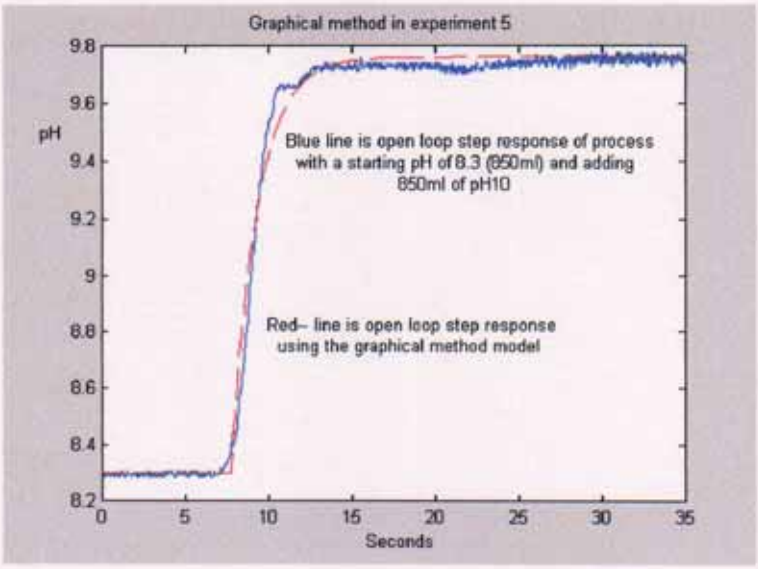

Figure (2.50) Open loop step response of process and FOPDT process model using the Graphical identification method

Figure (2.50) shows the accurate fitting of the process, under the conditions described in figure (2.50), and the FOPDT process model open loop step response. In general, the Graphical and Two-Point identification methods obtained satisfactory process models. The Area method of process model identification was less satisfactory. This is consistent with previous results. All the process and model open loop step responses are demonstrated in Appendix 2 section 2, pages A28 – A51.

2.5 Conclusions

Identification of a process model in open loop is a relatively straightforward exercise. In general, simple FOPDT process models are obtained using these techniques (SOPDT process models are also possible as described in the frequency-domain method). Care must be taken with the input signal to ensure that the output is operating in the linear region. Therefore, size of input signal and polarity of input signal are issues that need to be considered. A difficulty with open loop identification on a real process is that the process generally has to be taken out of commission for the experiments to take place.

Of the methods described in Chapter 2, the graphical and two-point methods in the time-domain, and the analytical and gradient approach in the frequency-domain were satisfactory in identifying accurate process models both in simulation and implementation. The area method was less satisfactory in identifying accurate process models and the method of moments was the least satisfactory of the open loop methods. A problem with the method of moments approach is the necessity of applying a pulse input to the system in the MATLAB/SIMULINK environment.

Chapter 3 : Closed Loop Time-domain Identification of a Process Model

3.1 Introduction

In 1982, Yuwana and Seborg (1982) proposed a simple on-line algorithm that used the closed-loop servo step response and the Pade approximation of the dead-time element to evaluate the parameters of a first-order-plus-dead-time process model. Yuwana and Seborg (1982) estimated the process model parameters from the under-damped process output obtained under proportional control. Bogere and Ozgen (1989) extended the Yuwana and Seborg method to estimate a second order plus dead time (SOPDT) process model as many processes are better represented by the SOPDT approximation. The test is also carried out in closed loop under proportional control. The identification method can be readily applied regardless of whether the response obtained during closed loop identification is under-damped or over-damped. Lee (1989) modified the identification algorithms by matching the dominant poles of the closed-loop model to the poles of the observed process response. This modification enables the method to be used for processes with large dead times. Mamat and Fleming (1995) proposed a method to identify a first-order-plus-dead-time process model in closed loop under PI control. The controller parameters are chosen so that the response is under-damped. Suganda *et al.* (1998) described a method for establishing second order plus dead time process model parameters under closed loop PI control. It can be readily applied regardless of whether the response obtained during closed loop identification is under-damped or over-damped.

This chapter investigates the closed loop identification methods of Bogere and Ozgen (1989), Mamat and Fleming (1995) and Suganda *et al.* (1998) in turn. These three methods are chosen because the popular FOPDT and SOPDT process models are identified using the Proportional (P) and Proportional/Integral (PI) control algorithms. The methods are firstly explained and then the identification techniques are carried out on a simulated process and a real process.

3.2 Process Under Proportional Control

The first closed loop identification technique investigated is based on a paper by Bogere and Ozgen (1989). The method identifies a SOPDT model shown in equation (1.3), introduced in Chapter 1. The test is carried out in closed-loop under proportional control, as shown in figure (3.1).

$$G_m(s) = \frac{K_m e^{-d_m s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{1.3}$$

K_m is the process model gain, d_m is the process model time delay and the two time constants are denoted by τ_1 and τ_2. The proportional gain is set so that the process output has an oscillatory response as shown in figure (3.2). The method is based on a method originally proposed by Yuwana and Seborg (1982). Consider a conventional feedback control loop. The process transfer function, $G_p(s)$, and the unmeasurable disturbances on the output, $d(s)$, are not precisely known (See figure (3.1)).

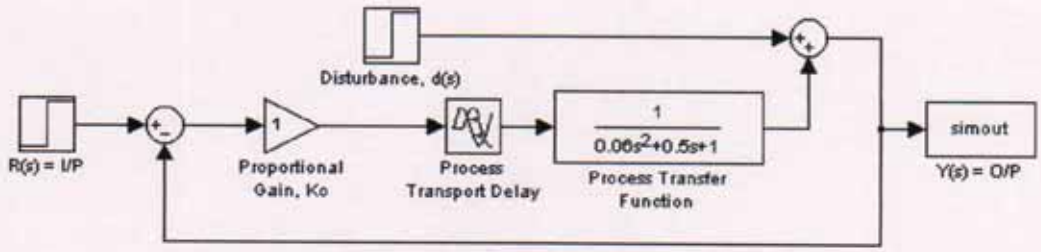

Figure (3.1) Conventional Feedback Control Loop

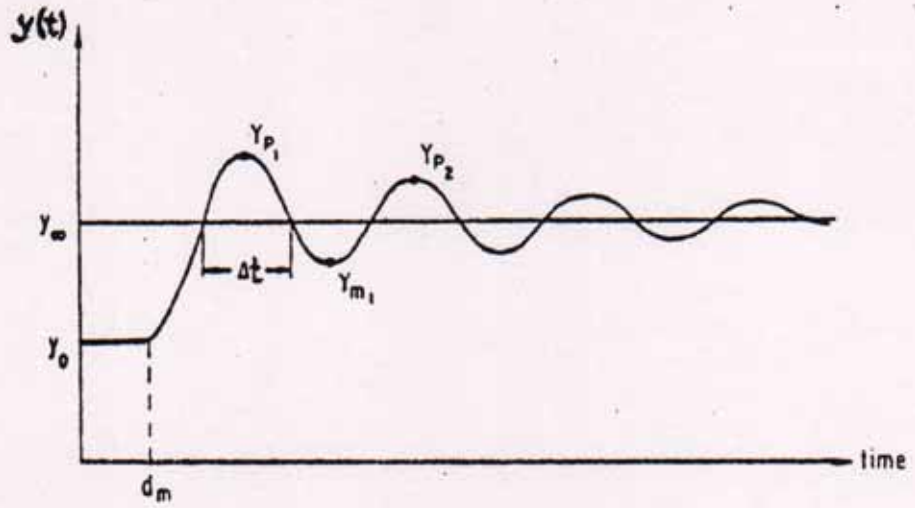

Figure (3.2) Underdamped transient response, for a step input

In the identification stage, the controller is switched to proportional control mode, such that controller $C(s) = K_c$. The closed loop transfer function, for set point changes, after substituting equation (1.3) for the process model transfer function, becomes

$$\frac{Y(s)}{R(s)} = \frac{K e^{-d_m s}}{(\tau_1 s+1)(\tau_2 s+1)+K e^{-d_m s}} \tag{3.1}$$

where $K = K_m K_c$. The process model parameters for equation (1.3), K_m, τ_1 and τ_2, can be analytically determined from the closed loop response data, if an approximation for the time delay in the denominator of equation (3.1) is made. The time delay term approximation used is a three-term modified Taylor approximation. This approximation is

$$e^{-d_m s} = 1 + a d_m s + b d_m^2 s^2 \tag{3.2}$$

where a = -0.8647 and b = 0.226 are the optimal values that effectively lump the higher order terms. The modified Taylor approximation is more accurate than the original three-term Taylor series approximation and the first order Pade approximation (Bogere and Ozgen, 1989). Substituting the modified-Taylor

approximation for the time delay into the denominator of equation (3.1), and rearranging the denominator results in

$$\frac{Y(s)}{R(s)} = \frac{K' e^{-d_m s}}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{3.3}$$

where

$$K' = \frac{K}{K+1} \tag{3.4}$$

$$\tau^2 = \frac{\tau_1 \tau_2 + K b d_m^2}{(1+K)} \tag{3.5}$$

$$\zeta = \frac{\tau_1 + \tau_2 + aK d_m}{2\sqrt{(1+K)(\tau_1 \tau_2 + K b d_m^2)}} \tag{3.6}$$

For a step input of magnitude A, such that $R(s) = A/s$, and a suitable K_c setting, an oscillatory response is obtained as shown in figure (3.2). The measurable quantities Δt and Y_0, Y_{p1}, Y_{p2}, Y_{m1} and Y_∞ on the response curve are used to determine the four SOPDT process model parameters. The process model gain, K_m, is obtained using equation (3.7).

$$K_m = \frac{|Y_\infty - Y_0|}{K_c(A - |Y_\infty - Y_0|)} \tag{3.7}$$

The time delay, d_m, is taken directly as the time interval between the time when the set-point input is made to the process and the time when the output from the process begins to respond to the input. The damping coefficient, ζ, in equation (3.3) is the average value of ζ_1, equation (3.8) and ζ_2, equation (3.9):

$$\zeta_1 = \frac{-\log_e(\alpha_1)}{\left(\pi^2 + (\log_e(\alpha_1))^2\right)^{1/2}} \tag{3.8}$$

$$\zeta_2 = \frac{-\log_e(\alpha_2)}{\left(4\pi^2 + (\log_e(\alpha_2))^2\right)^{1/2}} \tag{3.9}$$

where

$$\alpha_1 = \frac{Y_\infty - Y_{m1}}{Y_{p1} - Y_\infty} \tag{3.10}$$

and

$$\alpha_2 = \frac{Y_{p2} - Y_\infty}{Y_{p1} - Y_\infty} \tag{3.11}$$

The two time constants, τ_1 and τ_2 in the SOPDT process model, shown in equation (1.3), are determined using equations (3.12) and (3.13) respectively.

$$\tau_1 = \alpha + \beta \tag{3.12}$$

$$\tau_2 = \alpha - \beta \tag{3.13}$$

where

$$\alpha = \left(\frac{\Delta t}{\pi}\right)\zeta\sqrt{1-\zeta^2}(1+K) - 0.5aK\,d_m \tag{3.14}$$

and

$$\beta = \left(\beta_1 + \beta_2 + \beta_3\right)^{\frac{1}{2}} \tag{3.15}$$

with

$$\beta_1 = \left(\frac{\Delta t}{\pi}\right)^2(1-\zeta^2)(1+K)\left(\zeta^2(1+K)-1\right) \tag{3.16}$$

$$\beta_2 = \left(\frac{\Delta t}{\pi}\right)\zeta\sqrt{1-\zeta^2}(1+K)(-a)K\,d_m \tag{3.17}$$

$$\beta_3 = K\,d_m^2\left(0.25K\,a^2 + b\right) \tag{3.18}$$

Note that in equations (3.14), (3.16), (3.17) and (3.18), $K = K_m K_c$. It is also worth noting that the method works on an approximation of the delay term in equation (3.1) so therefore this approximation is a possible source of errors.

3.2.1 Simulation:

As a test of the accuracy of the method, a step input is applied to the simulated process shown in figure (3.3). The gain setting of 2 ensures an underdamped response as shown in figure (3.2). The second order process has a process gain equal to 1 and a time delay equal to 1 second.

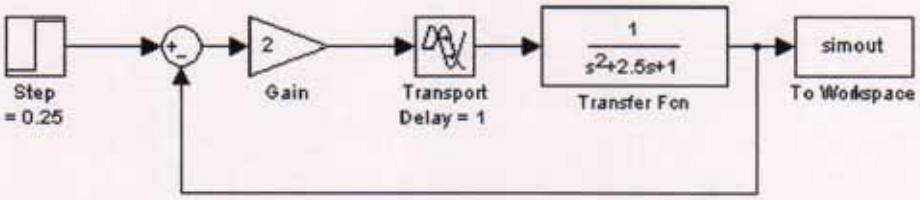

Figure (3.3) Simulated process to test the Bogere and Ozgen (1989) method

The MATLAB commands for the Bogere and Ozgen method are included in Appendix 2 section 3, page A53, entitled "CL_BandO_1"; the following SOPDT parameter values are deduced:

- Model gain, K_m, = 1.00
- Model time delay, d_m, = 1.20 seconds
- First time constant, τ_1, = 2.53 seconds
- Second time constant, τ_2, = 0.58 seconds

These four parameter values for the second-order-plus-dead-time process model in equation (1.3) are inserted into SIMULINK files and the model responses compared to the process responses. The model parameters for the closed loop test are shown in figure (3.4):

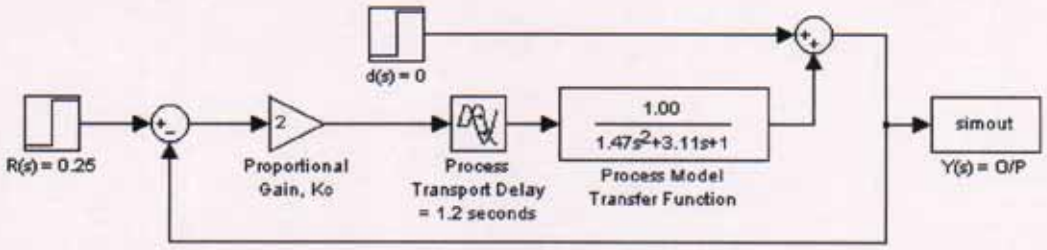

Figure (3.4) SIMULINK file with model parameters using Bogere and Ozgen method

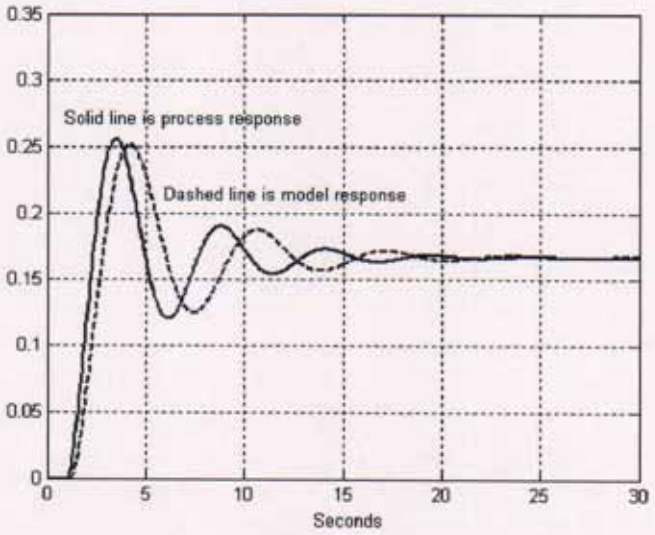

Figure (3.5) Closed loop step response comparison of process and model

Next, an open loop step test is carried out on the process and the model for further validation of the method. The SIMULINK files in figure (3.6) and (3.7) are used for this purpose. The resulting plot is shown in figure (3.8).

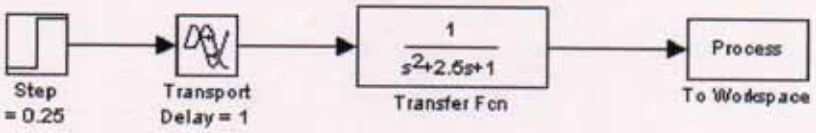

Figure (3.6) Open loop step test on SOPDT process

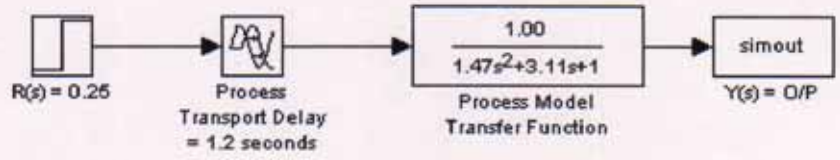

Figure (3.7) Open loop step test on SOPDT model of process in figure (3.6)

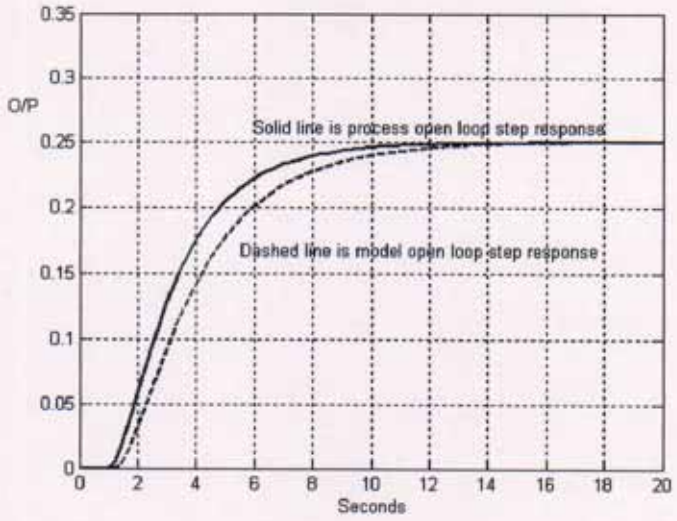

Figure (3.8) Open loop step response of process and model

The frequency-domain is now used as a further alternative validation of the model. Figure (3.9) shows the comparison of the nyquist plot for the process with the nyquist plot of the SOPDT process model. Figure (3.10) shows the bode plot of process and model. The MATLAB program to draw the nyquist and bode plots shown in figures (3.9) and (3.10) is given in Appendix 2 section 3, page A53, entitled "CL_BandO_2".

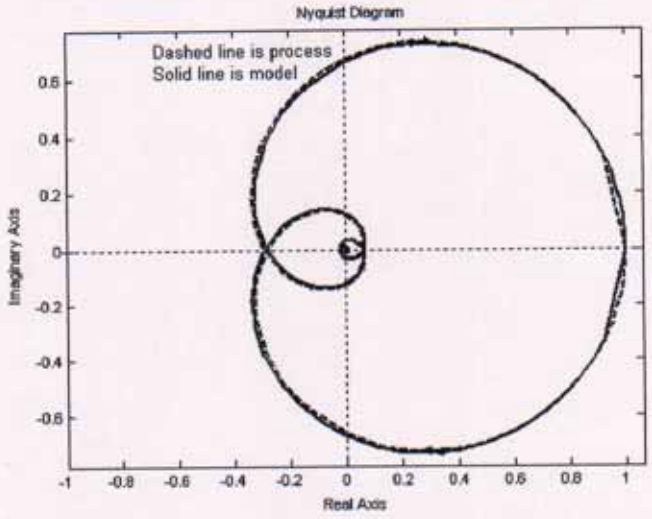

Figure (3.9) Nyquist plot of SOPDT process and model using Bogere and Ozgen (1989) identification method

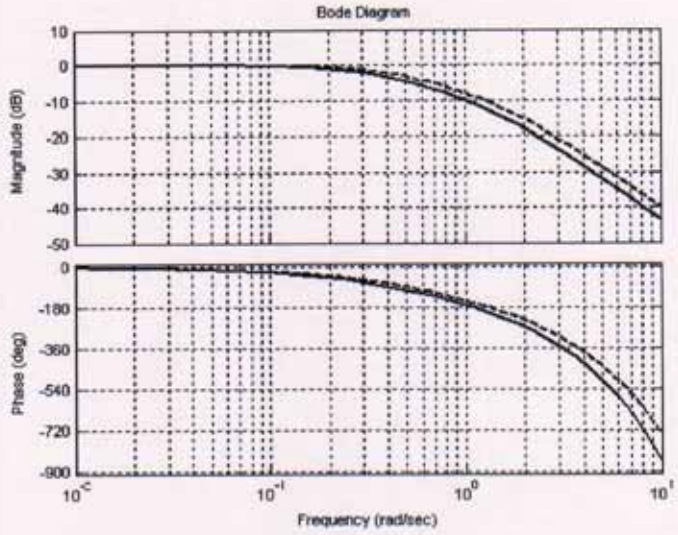

Figure (3.10) Bode plot of SOPDT process and model using Bogere and Ozgen (1989) identification method

The solid line is the bode plot of the model. The dashed line is the bode plot of the process.

Validation: The results of the plots in figures (3.5), (3.8), (3.9) and (3.10) show that the identified model is a reasonable replica of the SOPDT process. The simulated process model gain estimate, K_m, is the same as the process gain, K_p. This is very desirable. The time delay estimate of the simulated process model, d_m, is slightly longer than the simulated process time delay, d_p.

3.2.2 Implementation

The Real-Time Toolbox and Data Acquisition equipment are used as in previous experiments to identify a second order plus dead time model of the PT326 process trainer (Serial number 326-9-19). The SIMULINK file in Figure (3.11) is used. The command signal of 0.25 corresponds to a voltage step of 1.25 Volts and a temperature change from 25°C to 33°C. This size of step input ensures that the process trainer is operating in the linear range. This is desirable as the models identified are linear models. The response data is analysed to identify a SOPDT process model. The response data is shown in figure (3.12).

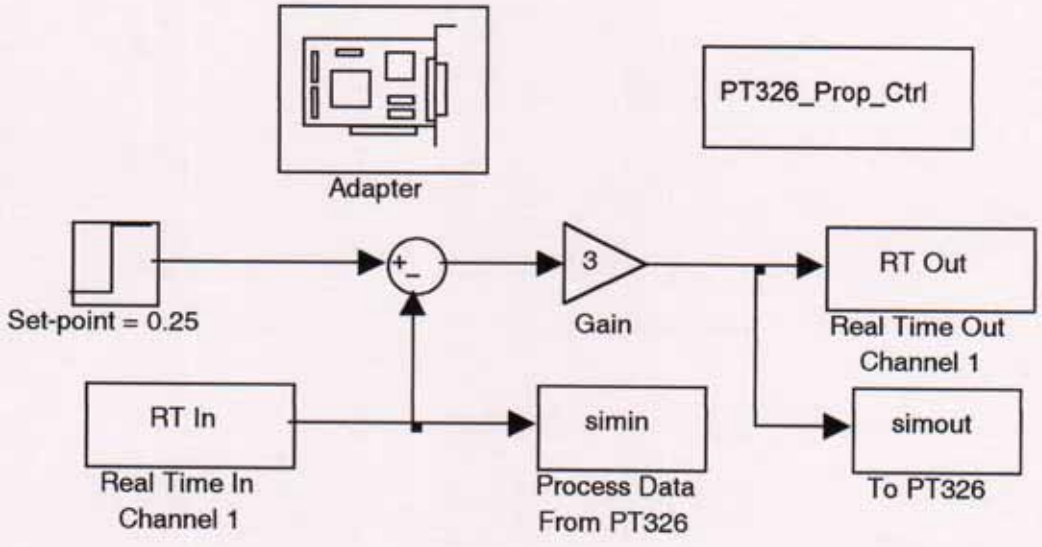

Figure (3.11) Proportional only controller for process identification

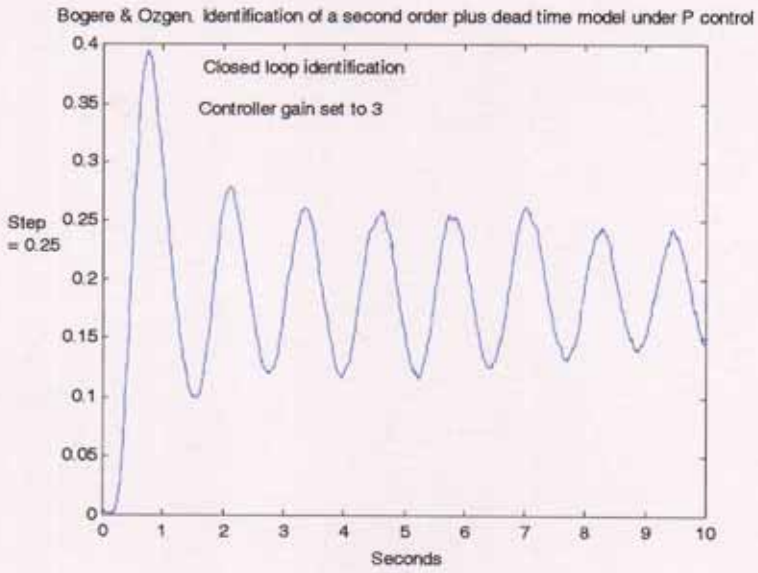

Figure (3.12) Plot of response data from PT326 Process Trainer to step input

Despite the fact that Figure (3.12) shows a non-linear response i.e. the decay ratio is non-uniform from peaks 1-2 and 2-3, it was decided to use the method to attempt to identify a SOPDT process model. From the data in Figure (3.12), the following results are obtained.

Y_{P1}, output value at first peak = 0.3960	[Figure (3.12)]
Y_{P2}, output value at second peak = 0.2681	[Figure (3.12)]
Y_{M1}, output value at first valley = 0.103	[Figure (3.12)]
Y_{∞}, output value at infinity = 0.2	[Figure (3.12)]
Δt = 0.72 seconds	[Figure (3.12)]
A = 0.25	[Figure (3.11)]
$\boldsymbol{K_m = 0.86}$	[Equation (3.7)]
K = 2.58	[$K = K_m K_c$]
$\boldsymbol{d_m = 0.25}$ **seconds**	[Figure (3.12)]
α_1 = 0.4949	[Equation (3.10)]
α_2 = 0.4393	[Equation (3.11)]

$\zeta_1 = 0.2185$ [Equation (3.8)]

$\zeta_2 = 0.1298$ [Equation (3.9)]

$\zeta = 0.1742$ [$(\zeta_1 + \zeta_2)/2$]

$\alpha = 0.4625$ [Equation (3.14)]

$\beta_3 = 0.1369$ [Equation (3.18)]

$\beta_2 = 0.0944$ [Equation (3.17)]

$\beta_1 = -0.1740$ [Equation (3.16)]

$\beta = 0.2393$ [Equation (3.15)]

$\tau_1 = \mathbf{0.70\ seconds}$ [Equation (3.12)]

$\tau_2 = \mathbf{0.22\ seconds}$ [Equation (3.13)]

The model parameters are inserted into a SIMULINK file as shown in figure (3.13). One (closed loop) validation result is shown to demonstrate the applicability of the method.

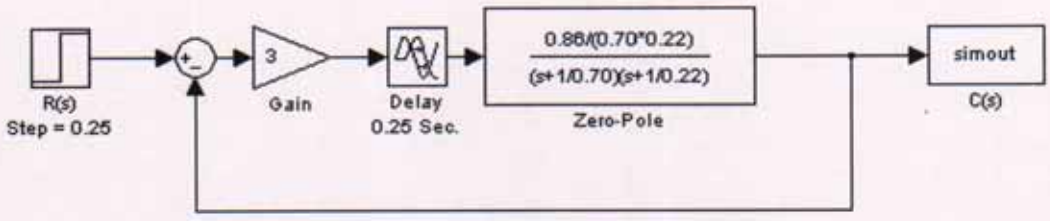

Figure (3.13) SIMULINK file with second order plus dead time model parameters

The program is run for 10 seconds and the results plotted and compared with the data from the real process shown in figure (3.11). The resulting plot is shown in figure (3.14).

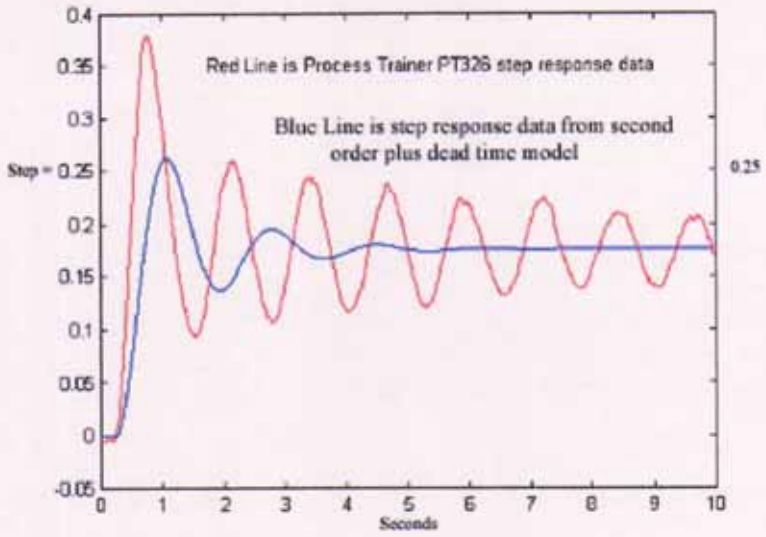

Figure (3.14) Comparison of closed loop step response of SOPDT model with closed loop step response of PT326 process trainer

Note the presence of the steady state error inherent in proportional controllers. The plot in figure (3.14) shows that the model captures some of the dynamics of the process though somewhat inaccurately. The process model gain estimate, K_m, is not large enough. However, Bogere and Ozgen's method is easily implemented. It is important to generate a well-defined underdamped closed loop step response to obtain the relevant points on the data.

3.3 Process Under Proportional Plus Integral Control

3.3.1 FOPDT Model

In earlier years, the first-order-plus-dead-time (FOPDT) process model is estimated from the process reaction curve obtained from an open loop step response of the process, with the risk of process runaway. Some of these open loop techniques have been explored in Chapter 2. As has been discussed, Yuwana and Seborg (1982) developed a method to approximate a process by a FOPDT model from the under-damped closed-loop step response data; the closed loop system was under

proportional control. The practical advantages of the Yuwana and Seborg (1982) method, and subsequent variations of the method, are that they require only a single closed-loop test and the algorithms are simple. The main disadvantage is that the test is performed under proportional control, which introduces steady-state offset during testing. A method described by Mamat and Fleming (1995) is used to identify a first-order- plus-dead-time model in closed-loop under PI control. Consequently, steady-state offset is eliminated. Since most of the controllers in industry are inherently PI controllers, previous knowledge of the operation of the controllers on the plant can be useful when selecting the test PI parameters, K_c and T_i. The model structure is shown in equation (1.1). The PI controller transfer function is shown in equation (3.19).

$$G_c(s) = K_c(1+\frac{1}{T_i s}) \tag{3.19}$$

If the PI controller parameters K_c and T_i are chosen such that the closed-loop response is under-damped, as shown in figure (3.15), then by using a 1st order Pade approximation for the dead-time term, $e^{-d_m s}$, in the denominator of the closed loop transfer function, the closed-loop response can be approximated by a second order plus dead-time transfer function (Mamat and Fleming (1995)):

$$G_{CL}(s) = \frac{C(s)}{R(s)} = \frac{K e^{-ds}}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{3.20}$$

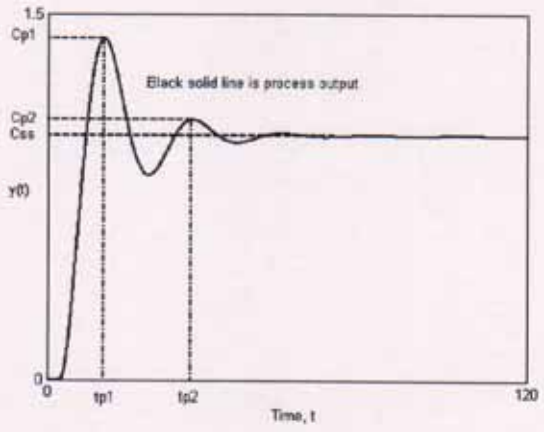

Figure (3.15) Typical under-damped closed-loop servo step response under PI control

From the closed loop step response data, five characteristic points, C_{p1}, C_{p2}, C_{ss}, t_{p1} and t_{p2}, are used to determine the second order plus dead-time approximation, equation (3.20), and subsequently, the frequency response of the closed-loop system. Knowing the dynamics of the closed-loop system and the dynamics of the controller, the open-loop dynamics of the process can be determined by separating the dynamics of the controller from the closed-loop dynamics. The equations to determine the closed loop SOPDT approximation parameters in equation (3.20), K, d, τ and ζ are as follows, where A is the magnitude of the set-point change as in section 3.2 (Mamat and Fleming (1995)):

$$K=\frac{C_{ss}}{A};\rho=-\frac{1}{2\pi}\ln\left[\frac{C_{p2}-C_{ss}}{C_{p1}-C_{ss}}\right];\zeta=\sqrt{\frac{\rho^2}{1+\rho^2}};$$
$$\tau=\frac{(t_{p2}-t_{p1})\sqrt{1-\zeta^2}}{2\pi}; \qquad (3.21)$$
$$d=\frac{S_c}{C_{ss}}-2\zeta\tau;S_c=\int_0^\infty[C_{ss}-C(t)]dt$$

with C_{ss}, C_{p2}, C_{p1}, t_{p1} and t_{p2} illustrated in figure (3.15). The frequency-domain is now used to determine critical points of the system. The phase crossover frequency, ω_c, and the magnitude at the phase crossover frequency, M, are the two pieces of information extracted from the frequency response data to use in subsequent calculations. It can be shown (Mamat and Fleming (1995)), that at the phase crossover frequency ω_C

$$|G_cG_p(j\omega_c)|=\frac{|G_{cl}(j\omega_c)|}{1+|G_{cl}(j\omega_c)|} \qquad (3.22)$$

$$\angle G_cG_p(j\omega_c)=0 \qquad (3.23)$$

$\angle G_cG_p(j\omega_c)$ is the phase angle of the loop transfer function at ω_c . Substituting Equations (3.19) and (1.1) into Equations (3.22) and (3.23), and solving for d_m and τ_m, the parameters of the FOPDT model, K_m, τ_m and d_m, are given by equations (3.24), (3.25) and (3.26).

$$K_m=\frac{T_i}{K_cS_c}C_{ss} \qquad (3.24)$$

$$\tau_m = \frac{\sqrt{(1+M)^2 (K_c K_m)^2 (1+T_i^2 \omega_c^2) - M^2 T_i^2 \omega_c^2}}{M \omega_c^2 T_i} \tag{3.25}$$

$$d_m = \frac{1}{\omega_c}\left[\tan^{-1}(\omega_c T_i + \tan^{-1}(\frac{1}{\tau_m \omega_c}) \right] \tag{3.26}$$

The equation for determining τ_m is a corrected version of the equation given by Mamat and Fleming (1995), as described originally by Kealy and O'Dwyer (2002a). The step-by-step procedure for developing equations (3.24) to (3.26) is now demonstrated:

$$G_c(s) = K_c(1 + \frac{1}{T_i s}) \tag{3.19}$$

$G_p(s)$ is modelled by $G_m(s)$:

$$G_p(s) \equiv G_m(s) = \frac{K_m e^{-d_m s}}{1 + \tau_m s} \tag{1.1}$$

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s}\right) = \frac{K_c(T_i s + 1)}{T_i s}$$

$$|G_c G_m(j\omega_c)| = \frac{|G_{cl}(j\omega_c)|}{|1 + G_{cl}(j\omega_c)|}$$

$$|G_{cl}(j\omega_c)| = M$$

$$|G_c G_m(j\omega_c)| = \frac{K_c K_m \sqrt{T_i^2 \omega_c^2 + 1}}{T_i \omega_c \sqrt{1 + \tau_m^2 \omega_c^2}} = \frac{M}{1+M}$$

Cross Multiply: $K_c K_m \sqrt{T_i^2 \omega_c^2 + 1}(1+M) = T_i \omega_c \sqrt{1 + \tau_m^2 \omega_c^2} M$

Divide both sides by $T_i \omega_c M$: $\sqrt{1 + \tau_m^2 \omega_c^2} = \frac{(1+M) K_c K_m \sqrt{T_i^2 \omega_c^2 + 1}}{M T_i \omega_c}$

Square both sides: $1 + \tau_m^2 \omega_c^2 = \frac{(1+M)^2 K_c^2 K_m^2 (T_i^2 \omega_c^2 + 1)}{M^2 T_i^2 \omega_c^2}$

Take 1 from both sides: $\tau_m^2 \omega_c^2 = \frac{(1+M)^2 K_c^2 K_m^2 (T_i^2 \omega_c^2 + 1)}{M^2 T_i^2 \omega_c^2} - 1$

Equivalent to: $\tau_m^2\omega_c^2 = \dfrac{(1+M)^2 K_c^2 K_m^2 (T_i^2\omega_c^2+1)}{M^2 T_i^2 \omega_c^2} - \dfrac{M^2 T_i^2 \omega_c^2}{M^2 T_i^2 \omega_c^2}$

Equivalent to: $\tau_m^2\omega_c^2 = \dfrac{(1+M)^2 K_c^2 K_m^2 (T_i^2\omega_c^2+1) - M^2 T_i^2 \omega_c^2}{M^2 T_i^2 \omega_c^2}$

Divide both sides by ω_c^2, i.e. multiply by $1/\omega_c^2$:

$$\tau_m^2 = \frac{1}{\omega_c^2}\left(\frac{(1+M)^2 K_c^2 K_m^2 (T_i^2\omega_c^2+1) - M^2 T_i^2 \omega_c^2}{M^2 T_i^2 \omega_c^2}\right)$$

Equivalent to: $\tau_m^2 = \dfrac{(1+M)^2 K_c^2 K_m^2 (T_i^2\omega_c^2+1) - M^2 T_i^2 \omega_c^2}{M^2 T_i^2 \omega_c^4}$

Square root of both sides: $\tau_m = \sqrt{\dfrac{(1+M)^2 K_c^2 K_m^2 (T_i^2\omega_c^2+1) - M^2 T_i^2 \omega_c^2}{M^2 T_i^2 \omega_c^4}}$

Equivalent to: $\tau_m = \dfrac{\sqrt{(1+M)^2 K_c^2 K_m^2 (T_i^2\omega_c^2+1) - M^2 T_i^2 \omega_c^2}}{M\omega_c^2 T_i}$

Equivalent to: $\tau_m = \dfrac{\sqrt{(1+M)^2 (K_c K_m)^2 (1+T_i^2\omega_c^2) - M^2 T_i^2 \omega_c^2}}{M\omega_c^2 T_i}$

The $(1+M)^2$ term was missing from equation (14) in the Mamat and Fleming (1995) paper, page 1298. Also, τ_m in equation (3.26) was written as τ, in this paper.

In summary, the equations to determine the model parameters are:

$$K_m = \frac{T_i}{K_c S_c} C_{ss} \tag{3.24}$$

$$\tau_m = \frac{\sqrt{(1+M)^2 (K_c K_m)^2 (1+T_i^2\omega_c^2) - M^2 T_i^2 \omega_c^2}}{M\omega_c^2 T_i} \tag{3.25}$$

$$d_m = \frac{1}{\omega_c}\left[\tan^{-1}(\omega_c T_i) + \tan^{-1}\left(\frac{1}{\tau_m \omega_c}\right)\right] \tag{3.26}$$

3.3.1.1 Simulation

To test the validity of the proposed method, a "known" process, shown in equation (3.27), is simulated using the MATLAB/SIMULINK software and the identification parameter results compared with the "correct" values. This is the same process as shown in equation (1.4).

$$G_P(s) = \frac{e^{-s}}{1+s} \tag{3.27}$$

This process is in closed-loop with a PI controller (see figure (3.16)) where the proportional gain is set to 1 and the integral time is set to 1 second.

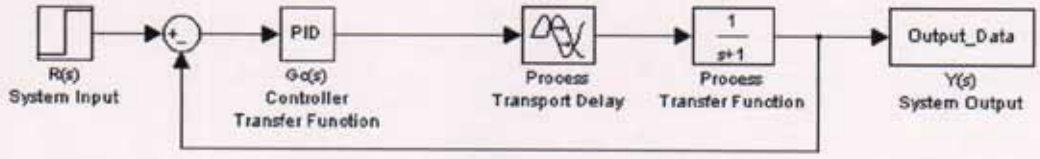

Figure (3.16) Block diagram of standard feedback control system

A step input, R(s) = 1, is applied to this system and the resulting output data is used to determine the parameters of a second order plus dead-time **approximation** of the closed-loop system in the time domain. The parameters of this approximation are calculated using the characteristic points C_{p1}, C_{p2}, C_{ss}, t_{p1} and t_{p2} as shown in figure (3.15) and equation (3.21). The plot of the output response is shown in figure (3.17).

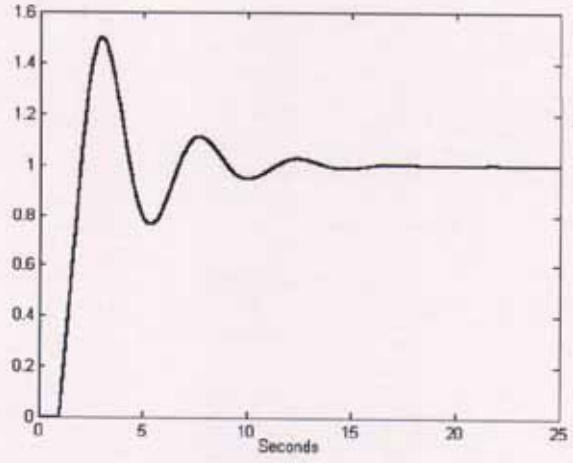

Figure (3.17) Closed loop step response of process in equation (3.27)

The values are determined as follows:

$C_{p1} = 1.5008$, $C_{p2} = 1.1122$, $C_{ss} = 1$, $t_{p1} = 3$ seconds, $t_{p2} = 7.71$ seconds.

Then, from equation (3.21), the following parameters are calculated:

$K = 1$, $\rho = 0.2381$, $\zeta = 0.2316$, $\tau = 0.7292$, $Sc = 1.0001$ and $d = 0.6623$ seconds.

The K, ζ, τ and d values are inserted into equation (3.20) to give the closed-loop second order approximation of the overall system. The frequency domain is now used to determine critical points of the system. The frequency response of the second order approximation is obtained using the **bode** command in MATLAB. The MATLAB commands to draw the plot are shown in figure (3.18).

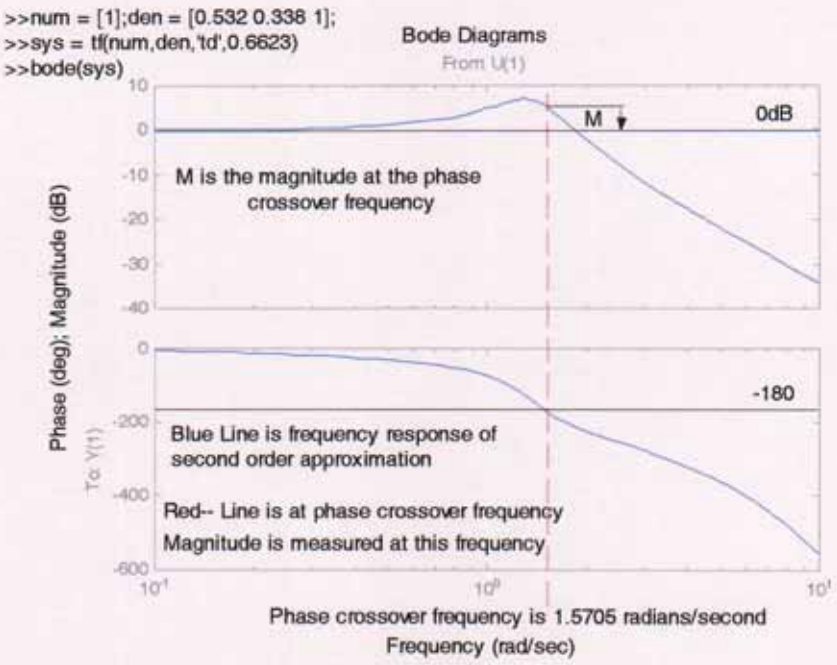

Figure (3.18) Frequency response plot of second order approximation

From figure (3.18), the phase crossover frequency and the magnitude at the phase crossover frequency are obtained using the following MATLAB commands:

>>[mag,phase,w] = bode(sys,w);

>>[gm,pm,wcp,wcg] = margin(mag,phase,w)

gm = 0.62: pm = -31.14: wcp = 1.57: wcg = 1.83.

From the bode plot in figure (3.18), the magnitude of the gain, ***M***, at the phase crossover frequency, ω_c = 1.57 rads/sec., is equal to $1/g_m$ = 1/0.62 = 1.62. The FOPDT model parameters estimated (by applying equations (3.24) to (3.26), with the FOPDT model parameters estimated using Mamat and Fleming's equations in brackets for comparison) are as follows:

- Model gain, K_m, = 1.00 (0.99).
- Model time delay, d_m, = 1.10 (0.99).
- Model time constant, τ_m, = 1.04 (1.04).

The "correct" value for each of these parameters is 1. The three estimated parameter values of the FOPDT model are inserted into a MATLAB/SIMULINK file and the model open-loop step response compared with the simulated process open-loop step response. A Nyquist plot of the FOPDT model and process is also drawn for validation of the proposed method.

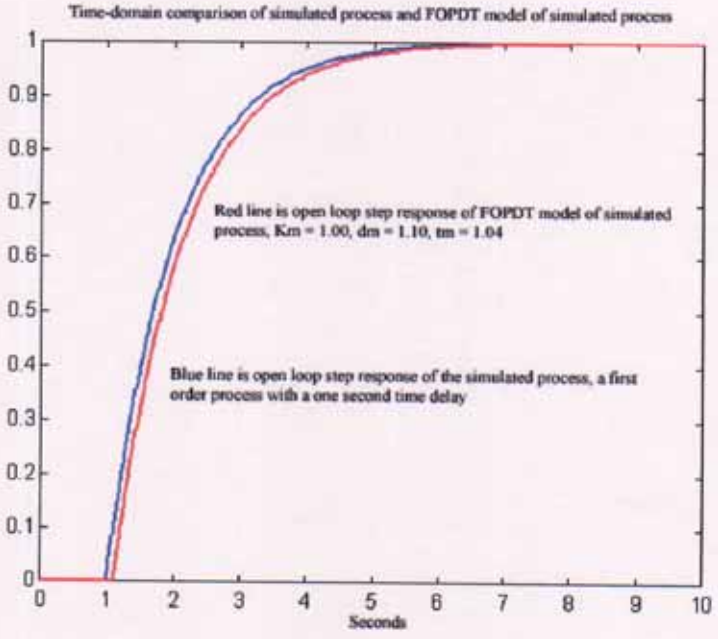

Figure (3.19) Step response of process in equation (3.27) and FOPDT model

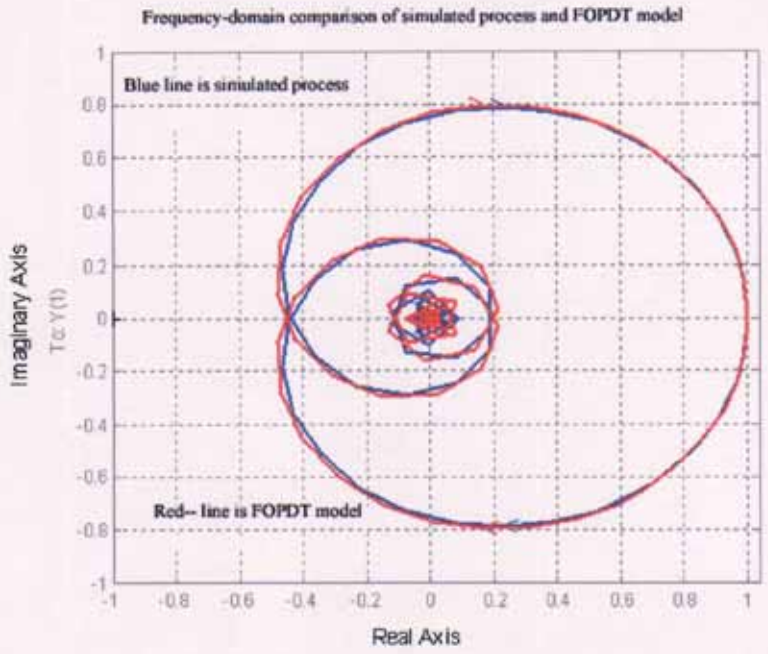

Figure (3.20) Nyquist plot of process in equation (3.27) and FOPDT model

Validation: From the time- and frequency-domain comparisons in figures (3.19) and (3.20), it can be seen that the quality of the "fit" between the process in equation (3.27) and the first order plus dead-time model of the process is reasonable. The modelling method gives a better process model gain estimate, K_m, than the Mamat and Fleming (1995) method. Mamat and Fleming (1995) give a better process model time delay estimate, d_m. The process model time constant estimate, τ_m, is the same in both cases. The results also show that the dynamics of the model replicate the dynamics of the process quite accurately. This is to be expected, however, as the process and the model are both of the first-order-plus-dead-time type.

A second simulated process (also considered by Mamat and Fleming (1995)) is then considered using the same methods and the results compared as before. This is a third order plus delay process, the transfer function of which is shown in equation (3.28).

$$G_P(s) = \frac{e^{-3s}}{(s+1)^2(1+2s)} \tag{3.28}$$

The PI controller values, $K_c = 0.6$ and $K_i = 0.2$ ($K_i = K_c/T_i$), ensures an under-damped closed-loop step response. The parameter values determined for the second order approximation (Equation (3.20)) are $K = 1$, $\zeta = 0.2636$, $\tau = 3.1642$ and $d = 3.3292$. From a bode plot, the phase crossover frequency, $\omega_c = 0.3529$ rads./sec. and magnitude, $M = 1.53$, are determined. Using equations (3.24), (3.25) and (3.26), the following first order plus dead-time model parameter values are obtained, with the results from Mamat and Fleming's equations in brackets for comparison:

- Model gain, K_m, = 1.00 (1.00)
- Model time delay, d_m, = 4.38 (4.69)
- Model time constant, τ_m, = 2.58 (2.59)

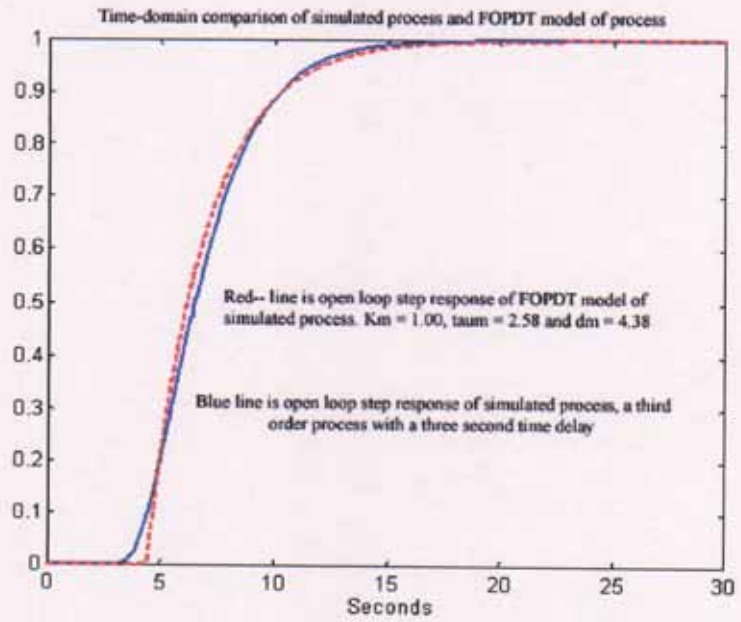

Figure (3.21) Open loop step response of process in equation (3.28) and FOPDT model

Validation: The time-domain open loop step response of the third-order-plus-dead-time process in equation (3.28) is shown in blue in figure (3.21). The first-order-plus-dead-time model of the process is shown by the red--line in figure (3.21). It can be seen that the model and process step response are close, so the model is valid.

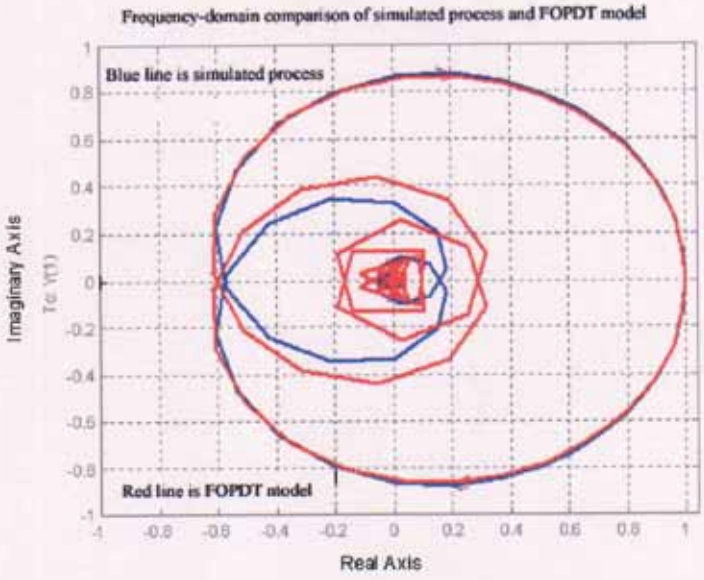

Figure (3.22) Nyquist plot of process in equation (3.28) and FOPDT model

Validation: The comparison of the nyquist plot of the third-order-plus-dead-time process in equation (3.28) with the nyquist plot of the first-order-plus-dead-time process model shows that the model is accurate, especially at low frequencies. The process model gain estimate, K_m, is very accurate.

3.3.1.2 Implementation

The method described in this section is now implemented on a real process (PT326 process trainer). Compared with the open-loop system identification methods, closed-loop identification methods are often more desirable in industrial applications because they cause less disruption to the operation of the system. The SIMULINK file in figure (3.23) is used in the closed loop identification process.

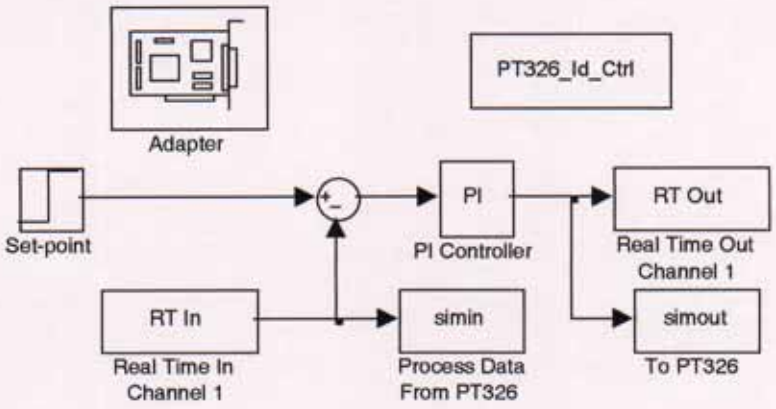

Figure (3.23) Closed loop system using real-time toolbox

The next task is to enter the proportional gain and integral gain controller parameters, using tuning rules, so that the system has an under-damped closed loop step response. The tuning rules require the model parameters be known in advance. For this reason, a representative result from the results of the experiments carried out in open loop is taken, yielding the following model parameter values and transfer function:

- Model gain, K_m, = 1.1,
- Model time constant, τ_m, = 0.6 seconds,
- Model dead time, d_m, = 0.26 seconds.

$$G_m(s) = \frac{1.1e^{-0.26s}}{1+0.6s} \tag{3.29}$$

These are considered as "starting values" of the process model parameters. The first tuning rule to be used is taken from page 16 of the "Handbook of PI and PID controller tuning rules", by O'Dwyer (2003). The process model is a FOPDT model shown in equation (1.1). The controller is the ideal PI controller and has the following transfer function:

$$G_c(s) = K_c\left(1+\frac{1}{T_i s}\right) \tag{3.19}$$

PI Tuning Rule 1: The tuning rule is the process reaction curve method of Ziegler and Nichols (1942).

$$K_c = \frac{0.9\times\tau_m}{K_m\times d_m} \tag{3.30}$$

$$K_c = \frac{0.9\times 0.6}{1.1\times 0.26}$$

$$K_c = \frac{0.54}{0.286}$$

$$K_c = 1.889$$

and

$$T_i = 3.33 d_m$$

$$T_i = 3.33\times 0.26 \tag{3.31}$$

$$T_i = 0.866$$

To convert integral time, T_i, to integral gain, K_i, the following formula is used:

$$K_i = \frac{K_c}{T_i}$$

$$K_i = \frac{1.889}{0.866} \tag{3.32}$$

$$K_i = 2.182$$

These settings give a quarter decay ratio as long as the following criteria is met:

$$\frac{d_m}{\tau_m} \le 1 \tag{3.33}$$

The parameters for proportional gain and integral gain are now entered into the SIMULINK file in figure (3.23) and the closed loop step response data is used to determine the "updated" FOPDT model parameters using the closed loop identification method. The step size is set to 0.2, which corresponds to a voltage reference of 1V. There is no clipping of the output signal at this value. The plot of the output response data is shown in figure (3.24).

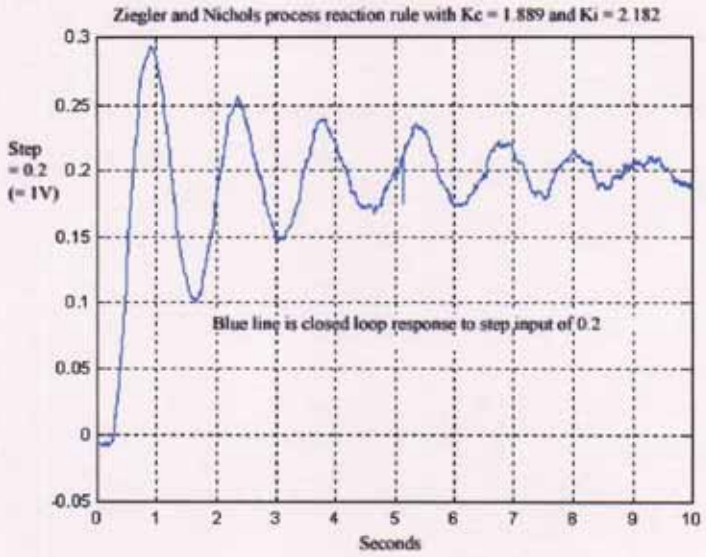

Figure (3.24) Closed loop step response – PT326

The algorithm proposed is now run to determine the "updated" parameters of the FOPDT process model. The program used to do this is entitled "CL_MandF_1" and is in Appendix 2 section 3, page A53, with the following second order approximation parameters:

- $K = 1.00$
- $d = 0.495$
- $\tau = 0.23$
- $\zeta = 0.084$

These parameters are obtained by applying equations (3.21) using the data in figure (3.24). The second order approximation of the system is then

$$G_{CL}(s) = \frac{C(s)}{R(s)} = \frac{1.00e^{-0.495s}}{0.0536s^2 + 0.039s + 1} \tag{3.34}$$

From the values of K, d, ζ and τ, the frequency response of the closed loop system, $G_{CL}(j\omega)$, is now determined using the MATLAB command **bode**, as shown in figure (3.25).

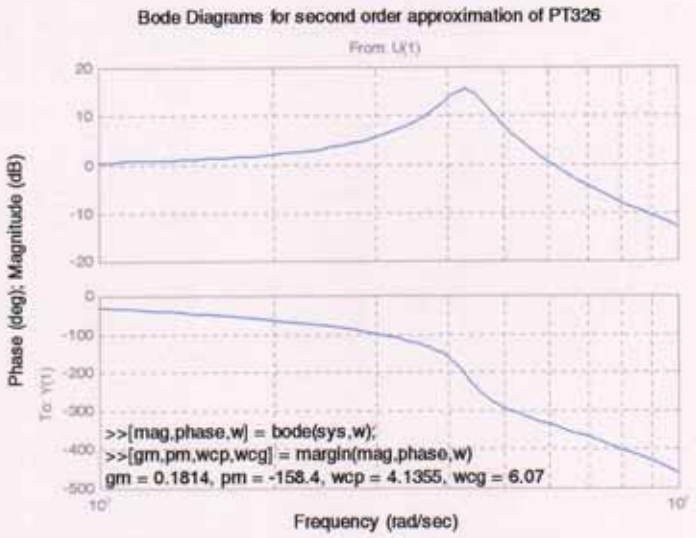

Figure (3.25) Bode plot of second order approximation of PT326

From the bode plot, the phase crossover frequency, ω_C, is found to be 4.1355 radians/second. The magnitude at this frequency, M, is (1/0.1814) = 5.5. The parameters of the "updated" FOPDT model are determined from equations (3.24), (3.25) and (3.26). The MATLAB commands are in Appendix 2 section 3, page A54, under the title "CL_MandF_2". This algorithm gives the following results:

- Model gain, K_m, = 0.86:
- Model time constant, τ_m, = 0.42 seconds:
- Model time delay, d_m, = 0.51 seconds.

In order to validate the model in the closed loop time-domain, a SIMULINK file is run, figure (3.26), using the above parameters and the result plotted in figure (3.27).

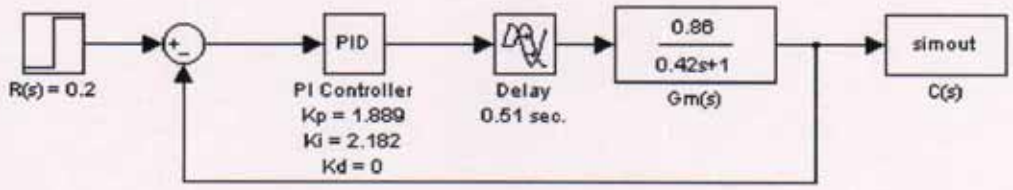

Figure (3.26) SIMULINK file with FOPDT parameters used to validate model

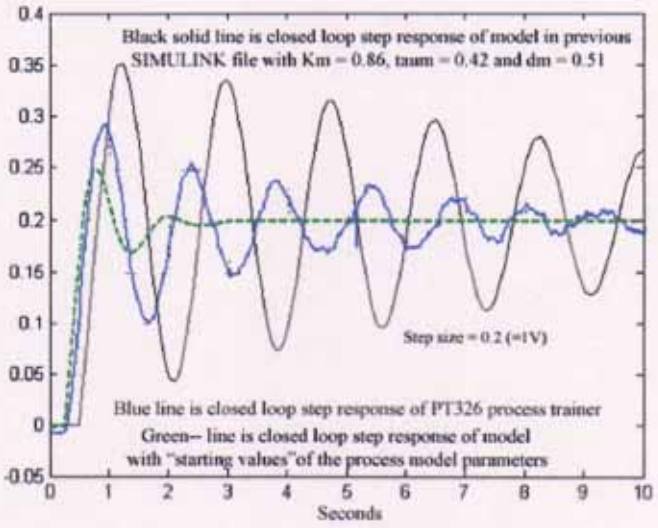

Figure (3.27) Closed loop step response of PT326, FOPDT model of PT326 in figure (3.26), and model with "starting values"

Validation: From the result in figure (3.27), it is clear that although the closed loop model fitting (Black solid line) is not perfect, it can be argued that it captures the dynamics of the process better than the fitting obtained in closed loop with the open loop model (Green -- line). This is to be expected as the validation test in figure (3.27) is a closed loop test.

PI Tuning Rule 2: To check the effect on the identification algorithm of using different PI controller test parameters, the "servo tuning" rules of the "Handbook of PI and PID controller tuning rules" by O'Dwyer (2003), page 29 are used. These rules are designed for minimum IAE by Rovira *et al.* (1969). The process model is

$$G_m(s) = \frac{1.1e^{-0.26s}}{1+0.6s}$$

The formula for the tuning rule is

$$K_c = \frac{0.758}{K_m}\left(\frac{\tau_m}{d_m}\right)^{0.861} \qquad (3.35)$$

$$T_i = \frac{\tau_m}{1.020 - 0.323\frac{d_m}{\tau_m}} \tag{3.36}$$

Comment: $0.1 \leq \frac{d_m}{\tau_m} \leq 1.0$

The PI controller parameters are calculated to be K_c = 1.416 and T_i = 0.68 using equations (3.35) and (3.36). Therefore K_i = 2.08. These values are inserted into the real-time SIMULINK file in figure (3.23) and a step input of 0.25 is applied to the system. The response data is plotted in figure (3.28).

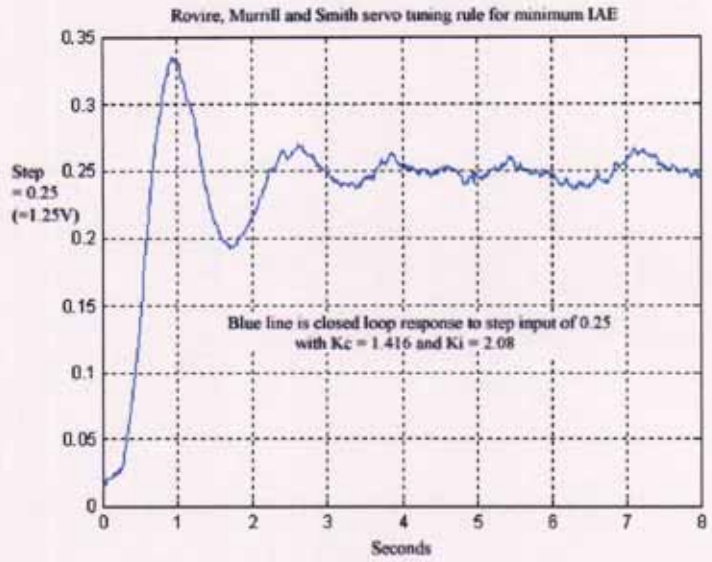

Figure (3.28) Closed loop step response – PT326

The algorithm proposed in Appendix 2 section 3, page A54, entitled "CL_MandF_3" is now run to determine the parameters of the SOPDT closed loop approximation shown in equation (3.20) with the following results:

- $K = 1.00$
- $d = 0.29$
- $\tau = 0.26$
- $\zeta = 0.23$

Thus,

$$G_{cl} = \frac{C(s)}{R(s)} = \frac{1.00e^{-0.29s}}{0.069s^2 + 0.1192s + 1} \tag{3.37}$$

From the values of K, d, τ and ζ, the frequency response of the closed loop system, $G_{cl}(j\omega)$, is now determined using the MATLAB command **bode**, as shown in figure (3.29).

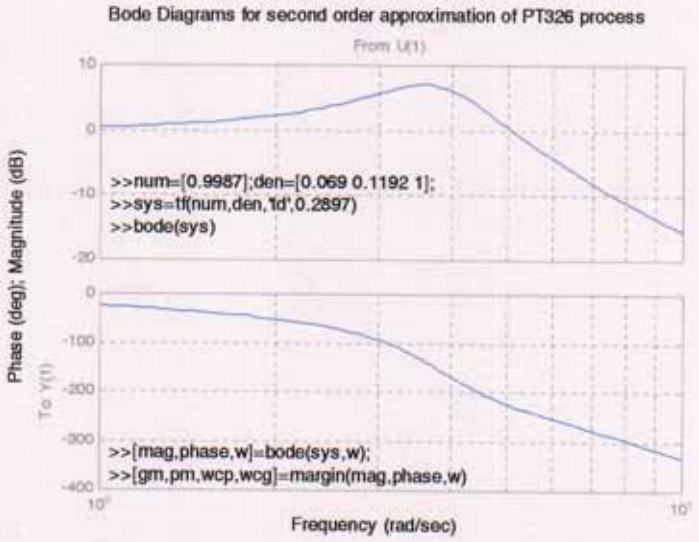

Figure (3.29) Bode plot of second order approximation of PT326

From the bode plot in figure (3.29), the phase crossover frequency, ω_C, is found to be 4.1543 radians/second. The magnitude at this frequency, M, is (1/0.5317) = 1.88. The parameters of the "updated" FOPDT model are determined from equations (3.24), (3.25) and (3.26). The algorithm to do this is entitled "CL_MandF_4" in Appendix 2 section 3, page A54. The first order plus dead time process model parameters are thus given as shown:

- Model gain, K_m, = 1.18:
- Model time constant, τ_m, = 0.61 seconds:
- Model time delay, d_m, = 0.48 seconds.

In order to validate the model, a SIMULINK file is run, figure (3.30), using the above parameters and the result plotted in figure (3.31).

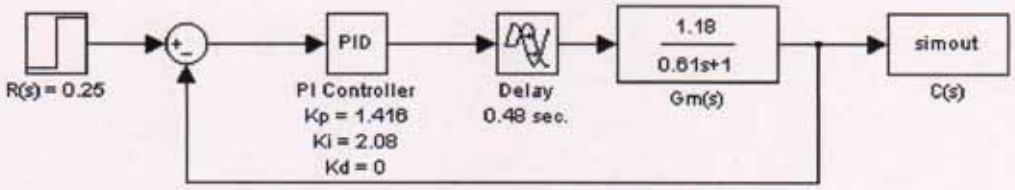

Figure (3.30) SIMULINK file with FOPDT parameters used to validate model

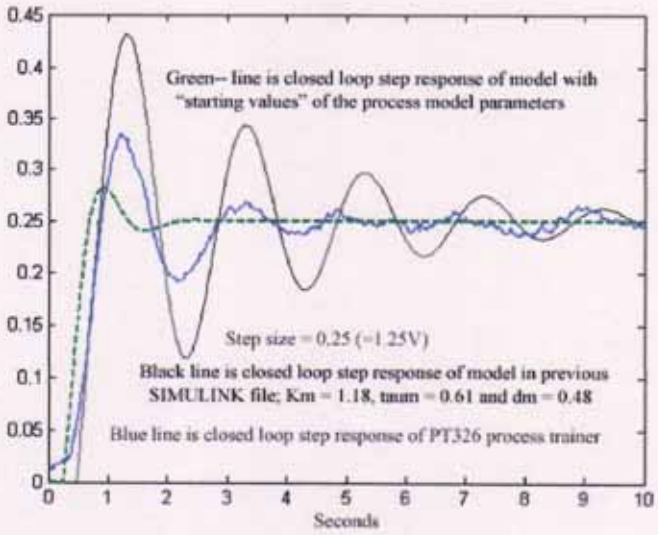

Figure (3.31) Closed loop step response of SIMULINK model of PT326, PT326 process trainer and model with "starting values"

Validation: Figure (3.31) indicates that the closed loop step response of the process model identified using the closed loop data and the identification method (Black solid line) can be argued to be a better representation of the closed loop step response of the process than the closed loop step response of the process model identified in open loop (Green -- line). It is interesting to note that, in both sets of results, the damping factor of the closed loop response of the system, whose process is identified using the

closed loop identification method, is less than that of the closed loop response of the process trainer (as shown in figures (3.27) and (3.31)). Similarly, the damping factor of the closed loop response of the system, whose process is identified using an open loop identification method, is greater than that of the closed loop response of the process trainer, for both sets of results.

Sensitivity Analysis - 1

From the results obtained from the aforementioned experiments, the process model gain, K_m, and the process model time constant, τ_m, estimates are quite close to the values obtained from the application of the open loop time-domain identification methods. However, the dead time, d_m, estimated by the closed loop identification technique differs considerably from that obtained from the average of the open loop techniques. It was considered possible that since all of the experiments are based on the calculation of five points on the step response of the closed loop process reaction curve (C_{ss}, C_{p1}, C_{p2}, t_{p1} and t_{p2}, as shown in figure (3.15)), the parameters identified may be sensitive to errors in the recording of this data. Each of these values are now changed by ±10% from the values obtained from figure (3.28), and the change in d_m with respect to the parameter is evaluated. As each one of the points is changed, the other four points are kept constant. In the first instance, C_{ss} is changed by +7% as a change of +10% gives complex values for ρ (in equation (3.21)). C_{p2} is changed by –7% for similar reasons.

C_{ss} changed by +7%

$$\frac{\Delta d_m}{\Delta C_{ss}} = \frac{-0.31}{0.07} = -4.43 \tag{3.38}$$

C_{ss} changed by –10%

$$\frac{\Delta d_m}{\Delta C_{ss}} = \frac{-0.33}{-0.1} = 3.3 \tag{3.39}$$

C_{p1} + 10%

$$\frac{\Delta d_m}{\Delta C_{p1}} = \frac{-0.0457}{0.1} = -0.45 \tag{3.40}$$

$$C_{p1} - 10\%$$

$$\frac{\Delta d_m}{\Delta C_{p1}} = \frac{0.0602356}{-0.1} = -0.6023 \tag{3.41}$$

$$C_{p2} + 10\%$$

$$\frac{\Delta d_m}{\Delta C_{p2}} = \frac{0.094987}{0.1} = 0.9498 \tag{3.42}$$

$$C_{p2} - 7\%$$

$$\frac{\Delta d_m}{\Delta C_{p2}} = \frac{-0.371525}{-0.07} = 5.3 \tag{3.43}$$

$$t_{p1} + 10\%$$

$$\frac{\Delta d_m}{\Delta t_{p1}} = \frac{0.29339}{0.1} = 2.9339 \tag{3.44}$$

$$t_{p1} - 10\%$$

$$\frac{\Delta d_m}{\Delta t_{p1}} = \frac{0.025}{-0.1} = -0.25 \tag{3.45}$$

$$t_{p2} + 10\%$$

$$\frac{\Delta d_m}{\Delta t_{p2}} = \frac{0.0674}{0.1} = 0.674 \tag{3.46}$$

$$t_{p2} - 10\%$$

$$\frac{\Delta d_m}{\Delta t_{p2}} = \frac{-0.0788}{-0.1} = 0.788 \tag{3.47}$$

These results indicate that the estimation of the time delay parameter, d_m, in the FOPDT process model, does depend on the accuracy at which relevant data points are

measured. The ten programs to determine the above parameters are located in Appendix 2 section 3, pages A54 – A58, under the titles "PD_1" to "PD_10".

Sensitivity Analysis – 2

Because the results of the implementation tests have not been as accurate as was hoped, two more implementation tests are carried out on the PT326 Process Trainer, with PI controller values obtained from a variety of tuning rules, and the results evaluated. The SIMULINK file in figure (3.32) is used in these experiments.

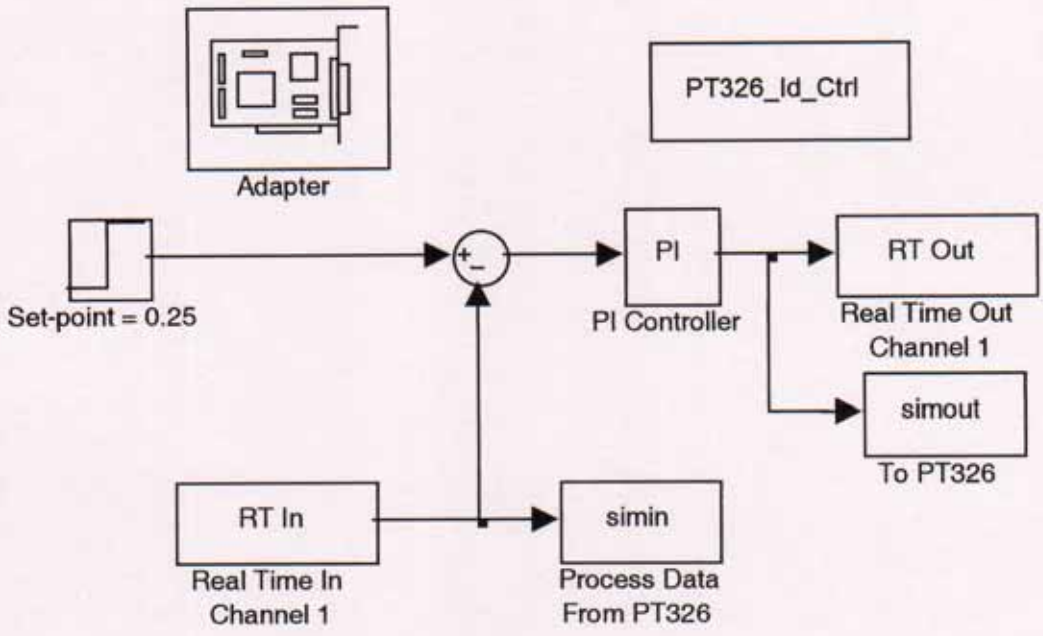

Figure (3.32) Real time data acquisition for Process Trainer, PT326

Two more tuning rules are used from the "Handbook of PI and PID controller tuning rules" by O'Dwyer (2003) to determine the controller parameters.

PI Tuning Rule 3: Page 31: Minimum ISE – Zhuang and Atherton (1993): K_c = 1.88, T_i = 0.96. The tuning rule is shown in equations (3.48) and (3.49):

$$K_c = \frac{0.980}{K_m}\left(\frac{\tau_m}{d_m}\right)^{0.892} \tag{3.48}$$

$$T_i = \frac{\tau_m}{0.690 - 0.155\dfrac{d_m}{\tau_m}} \tag{3.49}$$

Comment: $0.1 \le \dfrac{d_m}{\tau_m} \le 1.0$

The FOPDT process model is $G_m(s) = \dfrac{K_m e^{-d_m s}}{1+\tau_m s} = \dfrac{1.1e^{-0.26s}}{1+0.6s}$

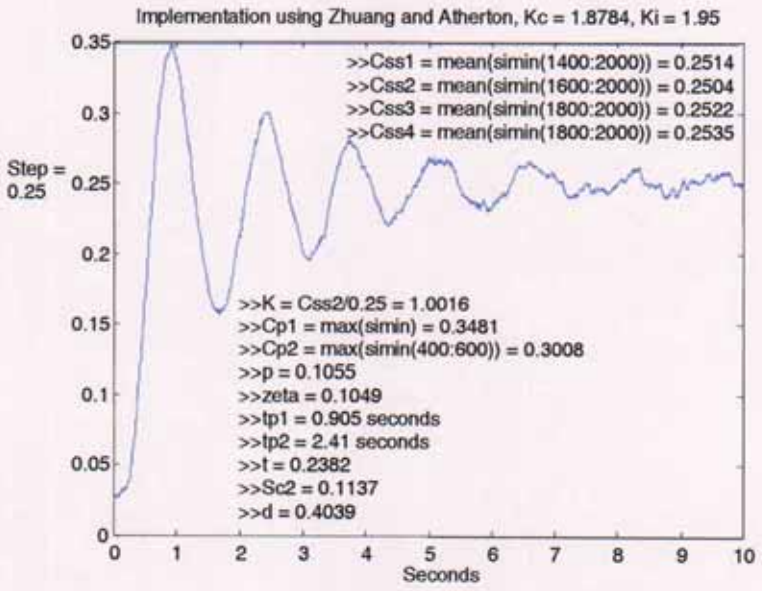

Figure (3.33) Closed loop step response of PT326

Using the values in figure (3.33) and the corresponding bode plot, the phase crossover frequency is found to be 4.15 radians/second and the magnitude, M, at this frequency equals 4.8. Using equations (3.24), (3.25) and (3.26), the three parameters for the FOPDT model are determined to be:

- Model gain, K_m, = 1.13
- Model time constant, τ_m, = 0.59 seconds
- Model time delay, d_m, = 0.51 seconds

These three FOPDT process model parameters are now inserted into a SIMULINK file and the closed loop step response compared to the closed loop process step response, for validation of the model. The resulting plot is shown in figure (3.34).

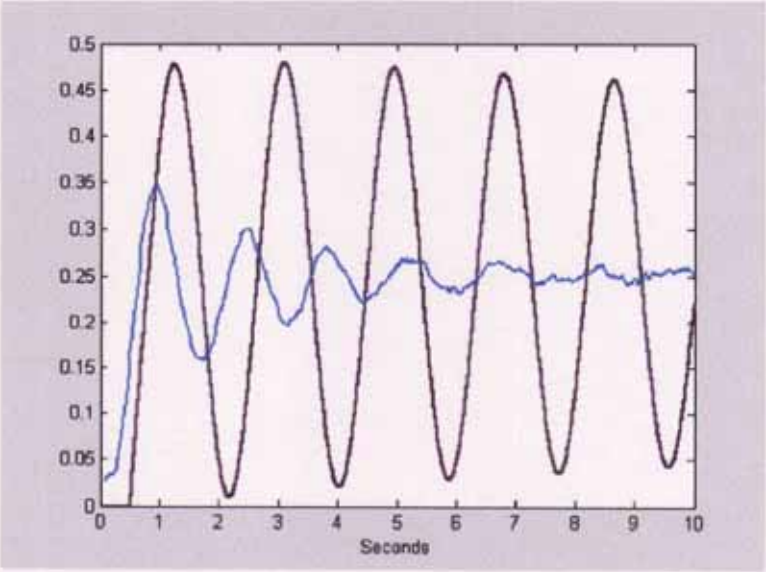

Figure (3.34) Closed loop step response: FOPDT process model (Purple) and process (Blue)

Validation: The result of plotting the process and model closed loop step response on the same plot in figure (3.34) shows that there is a large difference between the responses.

PI Tuning Rule 4: The next tuning rule used is on page 29 of the "Handbook of PI and PID controller tuning rules" by O'Dwyer (2003). The servo tuning rule is designed for minimum IAE by Smith and Corripio (1997). The tuning rule is shown in equations (3.50) and (3.51).

$$K_c = \frac{0.6\tau_m}{K_m d_m} \tag{3.50}$$

$$T_i = \tau_m \tag{3.51}$$

Comment: $0.1 \leq \frac{d_m}{\tau_m} \leq 1.5$

The FOPDT process model is $G_m(s) = \frac{K_m e^{-d_m s}}{1+\tau_m s} = \frac{1.1e^{-0.26s}}{1+0.6s}$

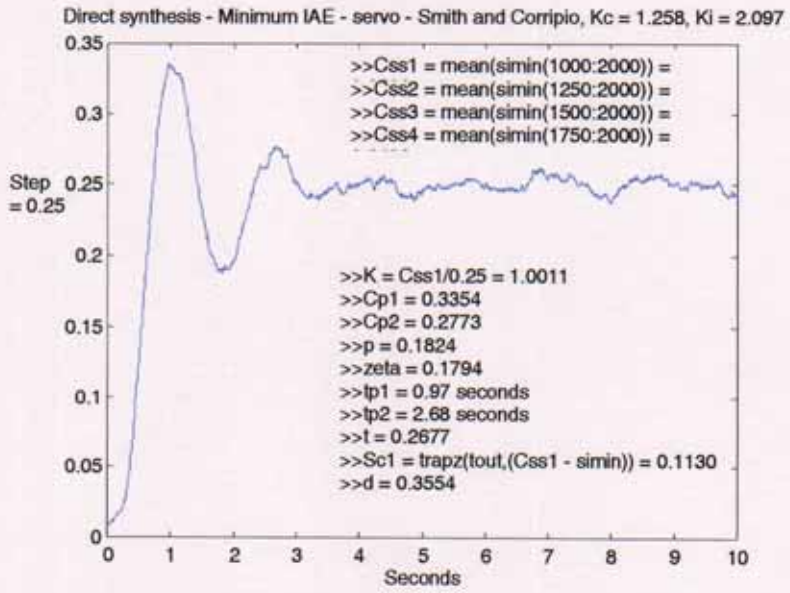

Figure (3.35) Closed loop step response of PT326

Using the values in figure (3.35) and the corresponding bode plot, the phase crossover frequency is found to be 3.87 radians/second and the magnitude, M, at this frequency equals 2.64. Using equations (3.24), (3.25) and (3.26), the three parameters for the FOPDT model are determined to be:

- Model gain, K_m, = 1.06
- Model time constant, τ_m, = 0.45 seconds
- Model time delay, d_m, = 0.50 seconds

These three FOPDT process model parameter values are inserted into a SIMULINK file and the model closed loop step response compared to the process closed loop step response. The resulting plot is shown in figure (3.36).

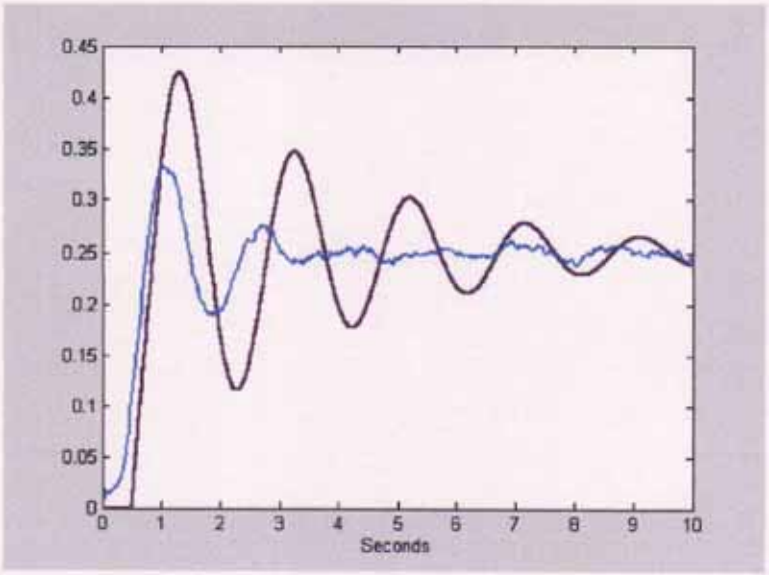

Figure (3.36) Closed loop step response: FOPDT process model (Purple) and process (Blue)

Validation: The result of plotting the process and FOPDT process model closed loop step response on the same plot in figure (3.36) shows that the model captures the dynamics of the process somewhat better than that shown in figure (3.34). Overall, it is clear from figures (3.27), (3.31), (3.34) and (3.36) that the choice of PI values has an impact of the goodness of fit of the closed loop model step response to the closed loop process step response.

The experiments in this section highlights a problem encountered when applying closed-loop identification techniques using tuning rules to determine controller parameters from process model parameters. The problem is as follows: What initial process model parameters are used to apply the tuning rules? The tuning rules require process model parameters to be known in advance, which is not always the case.

3.3.2 SOPDT Model

Some of the results obtained in section 3.3.1, when a FOPDT process model is identified, were not as satisfactory as expected. Therefore it is decided to investigate the identification of a SOPDT process model, with the aim of determining improved results. The closed loop identification method investigated is that proposed by Suganda *et al.* (1998) to identify a second-order-plus-dead-time process model (equation (1.2), introduced in chapter 1):

$$G_m(s) = \frac{K_m e^{-d_m s}}{\tau_m^2 s^2 + 2\tau_m \zeta_m s + 1} \tag{1.2}$$

In equation (1.2), K_m is the process model gain, d_m is the model time delay, τ_m is the model time parameter and ζ_m is the model damping coefficient. The PI controller is the ideal controller and has the following transfer function:

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s}\right) \tag{3.19}$$

The system is in closed-loop under PI control, see figure (3.37).

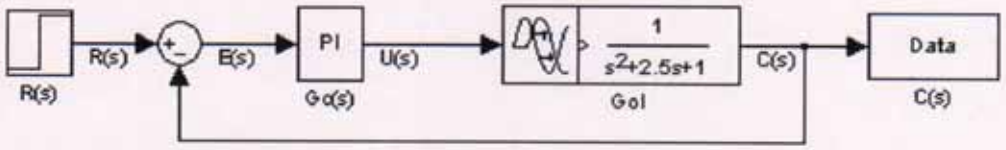

Figure (3.37) Block diagram of standard feedback control system

The identification method can be readily applied regardless of whether the response obtained during closed-loop identification is under-damped or over-damped. The drawbacks of the other methods investigated, i.e. a poorer model fit due to first order plus dead time process model assumed and/or the need for a step test under proportional control, are overcome. In this method, the same five characteristic points, as shown in figure (3.15), that are used in the method of Mamat & Fleming (1995), are also taken to determine the second-order-plus-dead-time model of the overall closed loop system.

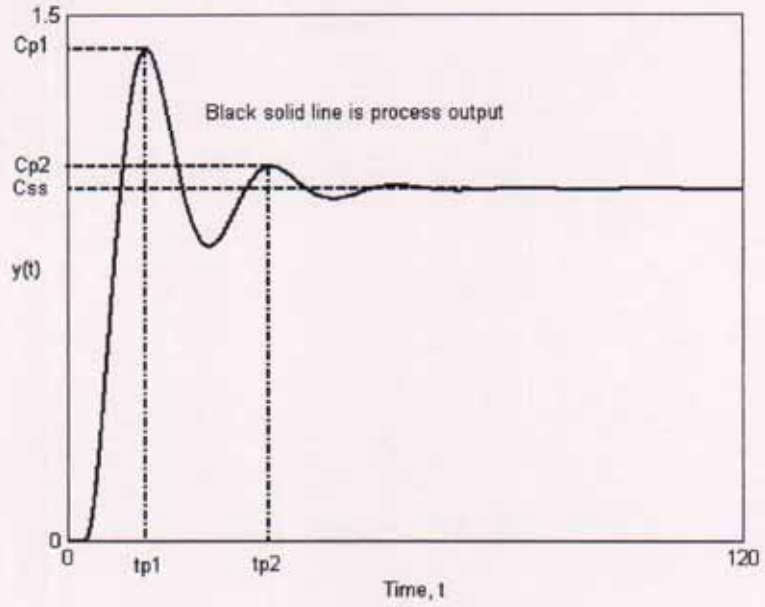

Figure (3.15) Typical under-damped closed-loop servo step response under PI control

The phase crossover frequency and the magnitude at this frequency are then determined (as done in the method of Mamat and Fleming (1995)); the four parameters for the second-order-plus-dead-time process model are subsequently calculated.

The closed loop transfer function for the system in figure (3.37) is then given by (Suganda *et al.* (1998)):

$$\frac{C(s)}{R(s)} = \frac{K_c K_m e^{-d_m s}(T_i s + 1)}{T_i s(\tau_m^2 s^2 + 2\tau_m \zeta_m s + 1) + K_c K_m e^{-d_m s}(T_i s + 1)} \qquad (3.52)$$

By using the Pade approximation for the time delay term, $e^{-d_m s}$, in the denominator of equation (3.52), the expression becomes a higher order transfer function with time delay, which can be approximated by a second-order-plus-dead-time model as shown in equation (3.20).

Note: The authors, Suganda *et al.* (1998) were asked if the Pade approximation for the time delay term, $e^{-d_m s}$, in the denominator of equation (3.52) was a first-order or

second-order Pade approximation. Dr. G.P. Rangaiah (e-mail: chegpr@nus.edu.sg) explained the following:

"Sentences below equation (3.52) [in their paper], including Pade approximation, provide qualitative reasoning for approximating equation (3.52) by a SOPDT model shown in equation (3.20). However, a specific (first or second order) Pade approximation was not used to derive equation (3.20), partly since some other approximation may also be required to obtain the SOPDT model in equation (3.20)."

$$G_{cl}(s) = \frac{C(s)}{R(s)} = \frac{K e^{-ds}}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{3.20}$$

If the closed loop step response is underdamped as shown in figure (3.15), the parameters for the SOPDT closed loop approximation in equation (3.20), K, τ, ζ and d can be calculated using equations (3.21) (defined by Mamat and Fleming (1995)).

Once the closed loop parameters, K, τ, ζ and d, are estimated for equation (3.20), the magnitude and phase angle of the closed loop transfer function, $G_{cl}(s)$, can be computed at any frequency, ω (Suganda *et al.* (1998)):

$$M = |G_{cl}(j\omega)| = \frac{K}{\sqrt{\left(1-\tau^2\omega^2\right)^2 + \left(2\tau\zeta\omega\right)^2}} \tag{3.53}$$

and

$$\alpha = \angle G_{cl}(j\omega) = \tan^{-1}\left(\frac{-2\tau\zeta\omega}{1-\tau^2\omega^2}\right) - d\omega \tag{3.54}$$

At the closed loop cross-over frequency, ω_c, equation (3.54) reduces to

$$-\pi = \tan^{-1}\left(\frac{-2\tau\zeta\,\omega_c}{1-\tau^2\omega_c^2}\right) - d\,\omega_c \tag{3.55}$$

Thus, ω_c can be obtained from equation (3.55) and the corresponding M from equation (3.53). From figure (3.37), the relationship between the open- and closed-loop transfer functions, $G_{ol}(j\omega)$ and $G_{cl}(j\omega)$ respectively, can be seen as

$$G_{ol}(j\omega) = G_c(j\omega)G_m(j\omega) = \frac{G_{cl}(j\omega)}{1-G_{cl}(j\omega)} \tag{3.56}$$

from which it is evident that

$$|G_c(j\omega_c)G_m(j\omega_c)| = \left.\frac{M}{1+M}\right|_{\omega_c} \tag{3.57}$$

Also,

$$\angle G_c(j\omega_c)G_m(j\omega_c) = -\pi \tag{3.58}$$

From equations (3.57) and (3.58), the dynamic parameters d_m, τ_m and ζ_m, of the open loop model in equation (1.2) need to be determined. The static gain of the model, K_m, can be calculated using the final value of the controller output, $U(\infty)$, where both C_{ss} and $U(\infty)$ are in the form of deviation variables.

$$K_m = \frac{C_{ss}}{U(\infty)} \tag{3.59}$$

To obtain the time delay, d_m, the phase angle information is employed as follows. The open-loop phase angle, ϕ, is related to the closed-loop magnitude and phase data as

$$\phi = \tan^{-1}\left[\frac{\sin\alpha}{\cos\alpha - M}\right] \tag{3.60}$$

In equation (3.60), α is the phase angle of the closed loop transfer function and M is the magnitude of the closed loop transfer function at this frequency (Suganda *et al.* (1998)).

Since α and M are known for any given ω, equation (3.60) allows the corresponding ϕ, the open loop phase angle, to be computed. As ω increases, the angle contribution in ϕ due to G_c (PI controller) and the denominator dynamics of G_m tend to zero and $-\pi$ respectively. Under this condition

$$\phi = \angle G_c(j\omega)G_m(j\omega) = -\omega d_m - \pi \tag{3.61}$$

from which the only unknown, d_m, can be estimated assuming a sufficiently large ω. To determine the remaining two model parameters (τ_m and ζ_m), the amplitude ratio and phase angle of the model, equation (1.2), and the PI controller transfer function, equation (3.19), may be evaluated at ω_c, to yield (Suganda *et al.* (1998)):

$$\tau_m^2\left(\tau_m^2\omega_c^2 - 2 + 4\zeta_m^2\right) = \frac{(K_cK_m)^2\left(\omega_c^2T_i^2+1\right)(M+1)^2 - M^2\omega_c^2T_i^2}{M^2\omega_c^4T_i^2} \tag{3.62}$$

and

$$\tan^{-1}(\omega_c T_i)+\tan^{-1}\left(\frac{-2\tau_m\zeta_m\omega_c}{1-\tau_m^2\omega_c^2}\right)=d_m\omega_c-\frac{\pi}{2} \tag{3.63}$$

These two equations can be solved for the two unknowns, τ_m and ζ_m, sequentially (Suganda *et al.* (1998)). Equation (3.62), upon rearranging, gives

$$\zeta_m=\frac{1}{2\tau_m}\times\sqrt{\frac{(K_C K_m)^2(\omega_c^2 T_i^2+1)(M+1)^2-M^2\omega_c^2 T_i^2}{M^2\omega_c^4 T_i^2}-\tau_m^4\omega_c^2+2\tau_m^2} \tag{3.64}$$

which, when substituted in equation (3.63) yields a non-linear equation for τ_m, which may be solved. A comparison of equation (3.20), the closed loop second order approximation, and the closed loop transfer function before using the Pade approximation for the dead time term, $e^{-d_m s}$, in the denominator indicate that τ_m should be less than τ, and so bounds for τ_m are 0 and τ. The solution of the non-linear equation for τ_m, obtained by combining equations (3.63) and (3.64), can easily be implemented on a spreadsheet, using a reasonable initial estimate for τ_m such as $\tau/2$. After finding τ_m, ζ_m can be calculated from equation (3.64).

3.3.2.1 Simulation

To test the validity of the proposed method, a "known" process is simulated using the MATLAB/SIMULINK software and identification parameter results compared with the "correct" values. The process is the second order process with a time delay of 0.28 seconds, shown in figure (3.38). The PI controller parameters are also shown in figure (3.38). Note that $K_i = K_c/T_i$. These values give the desired under-damped closed loop system output.

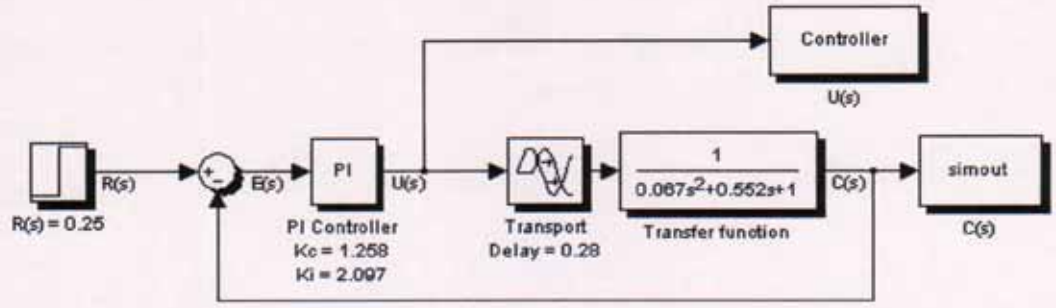

Figure (3.38) SIMULINK file for Suganda *et al.* (1998) method

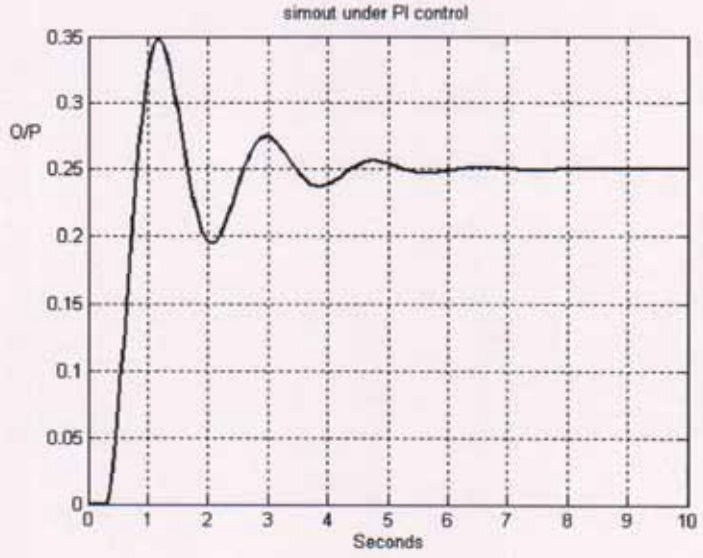

Figure (3.39) Closed loop step response

Using the response data from figure (3.39), and equation (3.21), in the algorithm entitled "Suganda_1" in Appendix 2 section 3, page A58, the following data is deduced:

- $K = 1$
- $\zeta = 0.21$
- $\tau = 0.28$
- $d = 0.91$

These values are inserted into equation (3.20). The closed loop frequency response plot is subsequently drawn (figure (3.40)). This allows the two pieces of critical information to be deduced, namely ω_c and M. The MATLAB commands to determine the phase crossover frequency, ω_c, and the magnitude at the crossover frequency, M, are in Appendix 2 section 3, page A58, entitled "Suganda_2".

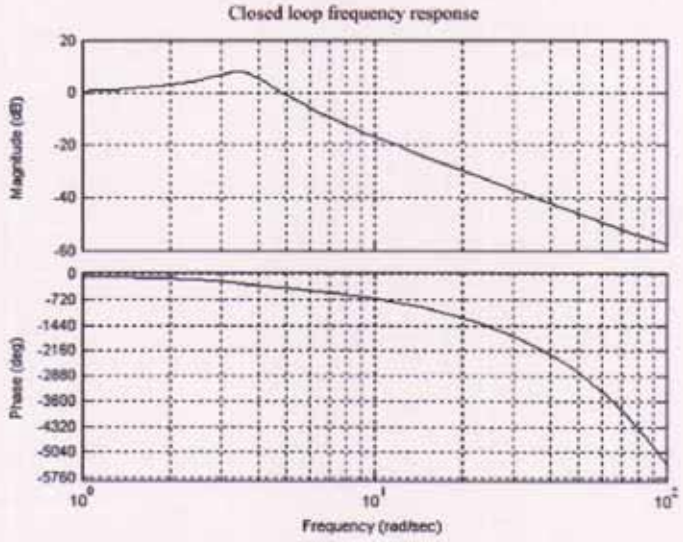

Figure (3.40) Closed loop frequency response (second order approximation of closed loop system)

The phase crossover frequency in radians/second, ω_c, is measured at the point where the system frequency response intersects with the $-180°$ point in figure (3.40). The magnitude of the system response, M, is then measured at this frequency. The phase crossover frequency, ω_c, is found to be **2.73 radians/second**. The magnitude of the response at this crossover frequency, M, is found to be **5.51 dBs**. This is equivalent to a gain of **1.89**.

A point is taken on the closed loop frequency response plot shown in figure (3.41) to determine the dead time of the process model, d_m. Note that d_m is best estimated

assuming a sufficiently large ω. This point, indicated by the Math1 symbol ■, is identified using the MATLAB software as shown.

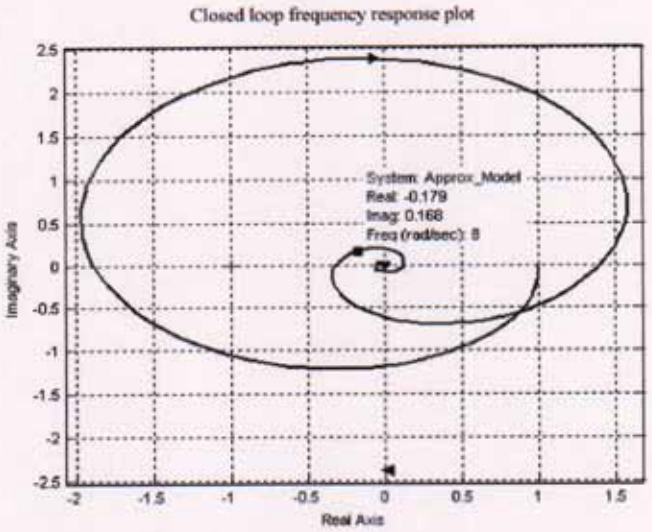

Figure (3.41) Closed loop frequency response plot of second order approximation

In figure (3.41) at the point indicated by the ■ symbol, M, the magnitude of the closed loop transfer function is calculated as 0.25. The phase angle of the closed loop transfer function, α, is calculated as –223.18° or –3.895 radians at this point. These two values are inserted into equation (3.60) to determine ϕ, the open loop phase angle. This results in ϕ being equal to –394.77° or –6.89 radians (allowing -2π radians for the correct quadrant). Equation (3.61) is now implemented to determine d_m, the model dead time estimate. The frequency of the response at the selected point in figure (3.41) is 8 radians/second (Note that phase angles must be converted to radians from degrees for consistency):

$$-6.98 = -(8)d_m - 3.14159 \quad \Rightarrow \quad d_m = 0.47\,\text{sec}.$$

The process model gain, K_m, is obtained using equation (3.59) as follows:

$$K_m = \frac{0.25}{0.25} = 1$$

To determine the remaining two model parameters (τ_m and ζ_m), equations (3.63) and (3.64) are utilized as described previously. In the solution of these equations, the following values, as determined in the procedure to this point, apply:

- $K_c = 1.26$
- $T_i = 0.60$
- $K_m = 1$
- $d_m = 0.47$
- $\omega_c = 2.73$
- $M = 1.89$
- $\tau = 0.28$

The bounds for τ_m are 0 to τ i.e. are 0 to 0.28 seconds. A number of values between these two extreme values are then inserted into equation (3.63) and the value of τ that most closely solves the equation is determined. This work is implemented using the Mathematica software (Wolfram, 1996); sample commands are given below.

$$\zeta_m = \frac{1}{2 \times \tau_m} \times \sqrt{\frac{(K_c \times K_m)^2 \times (\omega_c^2 \times T_i^2 + 1) \times (M+1)^2 - M^2 \times \omega_c^2 \times T_i^2}{M^2 \times \omega_c^4 \times T_i^2} - \tau_m^4 \times \omega_c^2 + 2 \times \tau_m^2}$$

$$\text{LHS} = \text{ArcTan}[(\omega_c \times T_i)] + \text{ArcTan}\left[\left(\frac{-2 \times \tau_m \times \zeta_m \times \omega_c}{1 - \tau_m^2 \times \omega_c^2}\right)\right]$$

With the model dead time, d_m, estimated, the right hand side of equation (3.63) is –0.29. When the τ_m model parameter is estimated as 0.26, the ζ_m model parameter in equation (3.64) is 1.37. Subsequently, the left hand side of equation (3.63) is –0.29. This calculation validates the estimated model transfer function as follows:

$$G_m(s) = \frac{1e^{-0.47s}}{0.068s^2 + 0.71s + 1} \tag{3.65}$$

The process model transfer function shown in equation (3.65) is now compared with the process transfer function shown in the SIMULINK file in figure (3.38), in an open loop frequency-domain validation test (figure (3.42)).

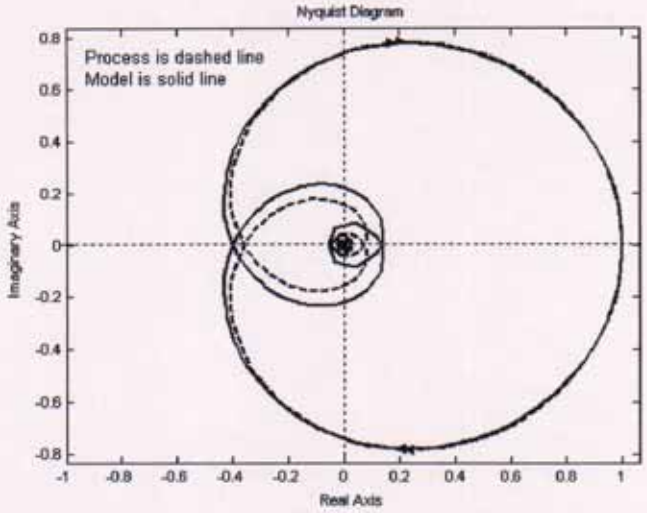

Figure (3.42) Nyquist plot of process and SOPDT model of process

The nyquist plot in figure (3.42) shows that the model structure and model parameters are valid for the simulated process under test. An open loop step test is now performed on the results as an alternative validation of the process model; the result is shown in figure (3.43).

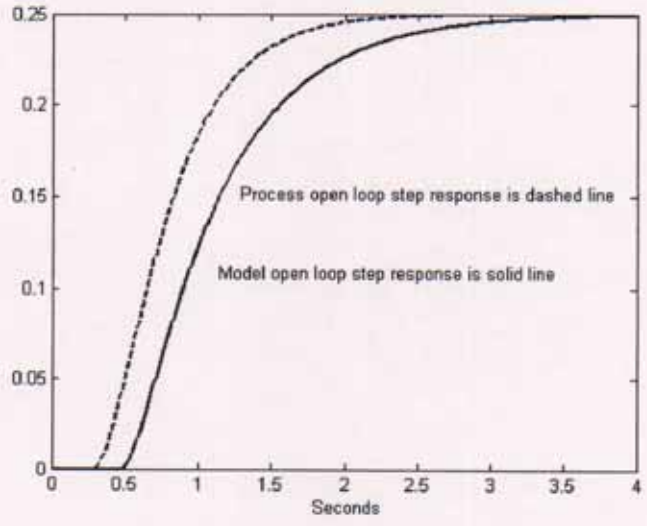

Figure (3.43) Comparison of SOPDT process and SOPDT process model open loop step response

The plot in figure (3.43) indicates a difference in the open loop step response of the process and model. The time delay estimation, d_m, seems to be the major source of error. The process model gain estimate, K_m is a very good estimate of the process gain, K_p. In a final validation test, the closed loop step response of the process and the model are compared.

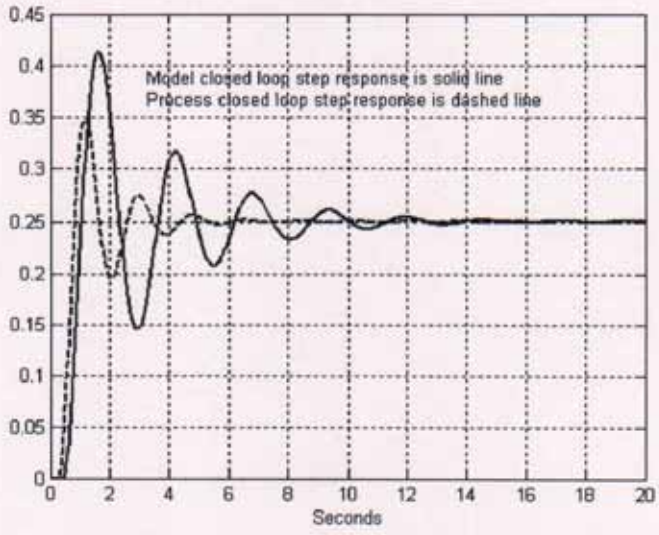

Figure (3.44) Closed loop step response of process and model

Validation: The plot in figure (3.44) shows some differences between the closed loop step response of the model and the closed loop step response of the process. The time-domain comparisons in figures (3.43) and (3.44) highlighted this more than the frequency-domain comparison in figure (3.42), showing the usefulness of different validation tests.

3.3.2.2 Implementation

The identification techniques proposed by Suganda *et al.* (1998) are now implemented on the PT326 process trainer. The SIMULINK file used to determine the gain parameter of the second-order-plus-dead-time process model is shown in figure (3.45); the closed loop step response of the system is shown in figure (3.47).

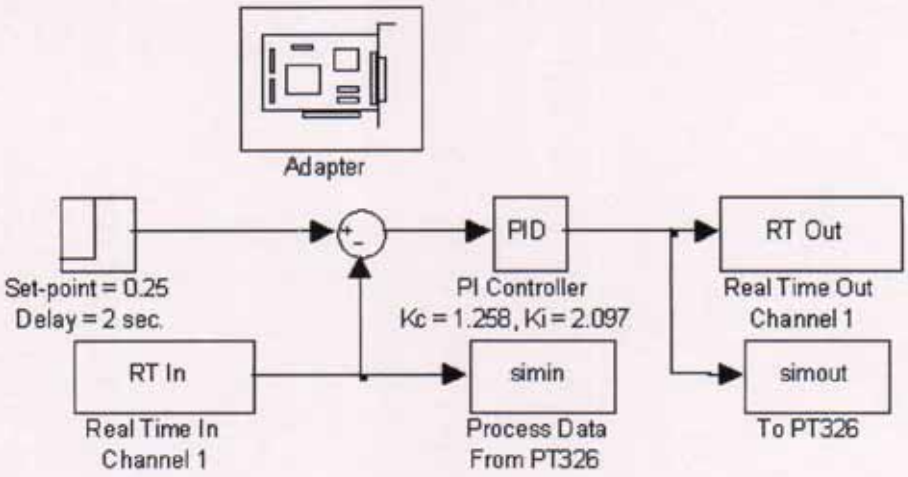

Figure (3.45) SIMULINK file used to output data to PT326 process trainer

The output from the controller after the step is applied is shown in figure (3.46).

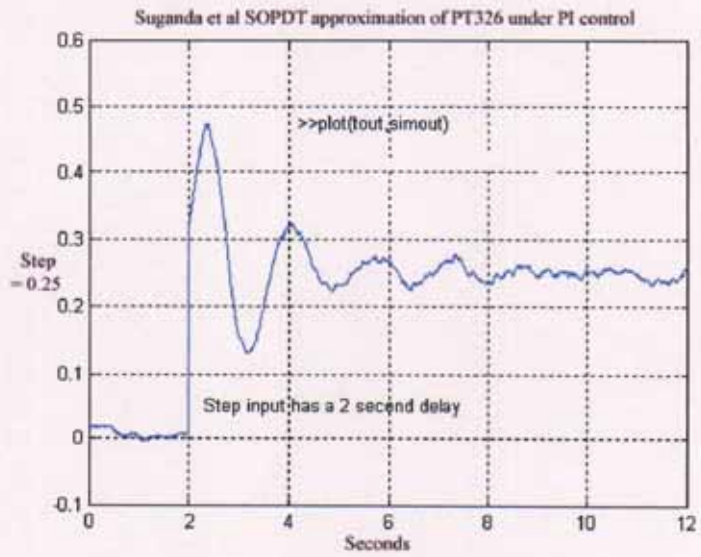

Figure (3.46) Controller output from SIMULINK file shown in figure (3.45)

>>U(∞) = mean(simout(1600:2400)) $U(\infty) = 0.25$

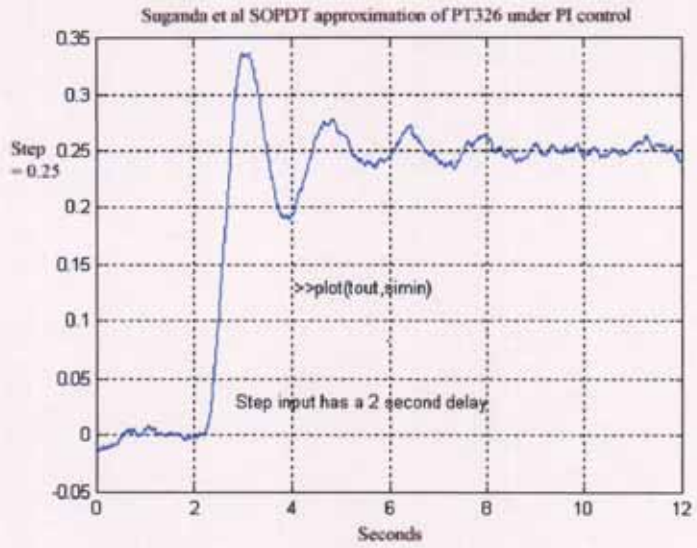

Figure (3.47) Closed loop step response of PT326 under PI control

>>Css = mean(simin(1700:2400)) $Css = 0.25$

The static gain of the model, K_m, can be calculated using the final value of the controller output, $U(\infty)$ (Suganda *et al.* (1998))

$$K_m = \frac{C_{ss}}{U(\infty)} = \frac{0.25}{0.25} = 1.00 \qquad (3.66)$$

The second-order-plus-dead-time approximation model of the closed loop system, equation (3.20), using the Mamat and Fleming (1995) technique (equation (3.21)), is now determined. The parameters determined are $K = 1.00$, $d = 0.36$, $\tau = 0.27$ and $\zeta = 0.18$. The phase crossover frequency, ω_c, and the magnitude at the phase crossover frequency, M, are subsequently calculated as follows: $\omega_c = 3.87$ radians/second, $M = 2.64$. Details are provided in Appendix 2 section 3, page A58, entitled "Suganda_2". A closed loop frequency response plot is drawn to determine the open loop phase angle.

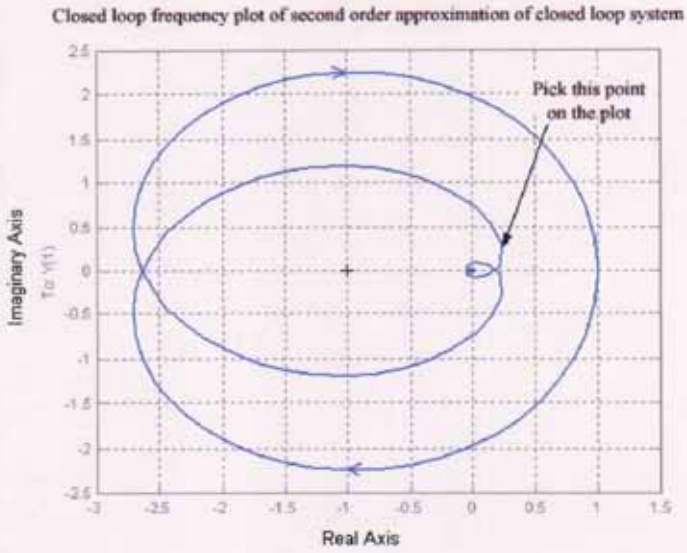

Figure (3.48) Closed loop frequency plot of second order approximation of closed loop system

The arrow in figure (3.48) shows the point chosen to ensure that a sufficiently large ω is used to determine the time delay, d_m, of the model. At this point, Imaginary = 0.269, Real = 0.224 and Frequency = 7 radians/second. For equation (3.60), α, the phase angle of the closed loop transfer function is –309.78° or –5.4067 radians at a frequency of 7 radians/second. The magnitude at this frequency is 0.35. Putting these values into equation (3.60), it can be determined that ϕ, the open loop phase angle = -290.7° or –5.073 radians at a frequency of 7 radians/second. Equation (3.61) is now implemented to determine the dead time of the second-order-plus-dead-time model, d_m:

$$\phi = -\omega d_m - \pi$$
$$-5.073 = -7 d_m - \pi$$
$$-5.073 + 3.14159 = -7 d_m$$
$$-1.931449 = -7 d_m$$
$$d_m = \frac{-1.931449}{-7}$$
$$d_m = 0.28\,\text{sec}$$

In the estimation of the remaining two model parameters, τ_m and ζ_m, equations (3.63) and (3.64) are used. The following values, as determined in the procedure to this point, apply in the solution of these equations:

$K_c = 1.26$; $K_m = 1.00$; $T_i = 0.6$ seconds; $\omega_c = 3.87$ radians/second; $d_m = 0.28$seconds; $M = 2.64$; $\tau = 0.27$.

As $\tau = 0.27$, the bounds for τ_m are 0 and 0.27. Seven different values for τ_m were estimated between its bounds and the corresponding ζ_m calculated. This value is then inserted into equation (3.63) to check if the estimate is correct. The step responses of the models determined at these seven sets of parameters are illustrated in figure (3.49).

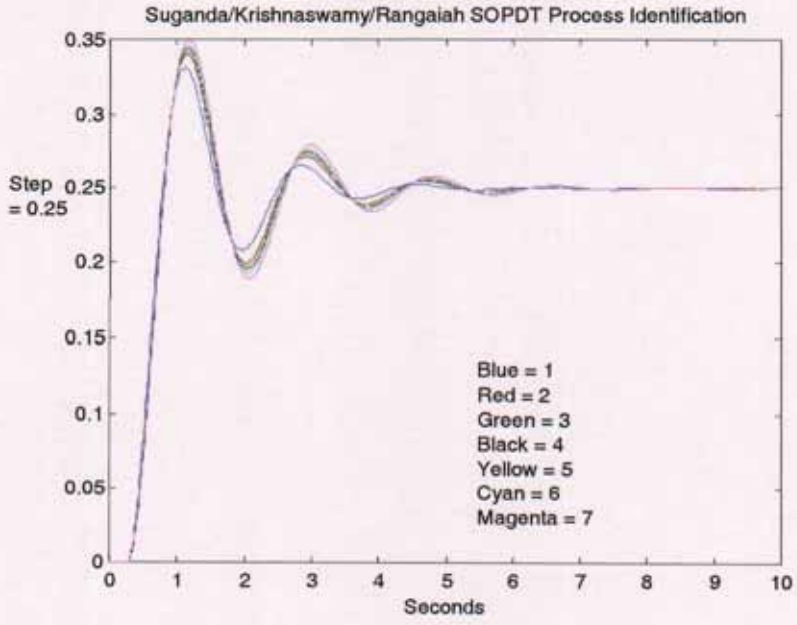

Figure (3.49) Step response of models determined at seven sets of (ζ_m, τ_m) parameters

Data for figure (3.49) are as follows:

- Blue (1), $\tau_m = 0.23$, $\zeta_m = 1.19433$.
- Red (2), $\tau_m = 0.25$, $\zeta_m = 1.10348$.
- Green (3), $\tau_m = 0.253$, $\zeta_m = 1.09068$.
- Black (4), $\tau_m = 0.258$, $\zeta_m = 1.0697$.
- Yellow (5), $\tau_m = 0.26$, $\zeta_m = 1.0615$.
- Cyan (6), $\tau_m = 0.2627$, $\zeta_m = 1.0505$.
- Magenta (7), $\tau_m = 0.27$, $\zeta_m = 1.0212$.

The most accurate the solution of equation (3.63) is $\tau_m = 0.258$, with a corresponding ζ_m value = 1.0697.

The four estimated model parameters are thus:

- Model gain, K_m, = 1.00
- Model time delay, d_m, = 0.28 seconds
- Model time parameter, τ_m, = 0.26 seconds
- Model damping coefficient, ζ_m, = 1.07

The four estimated process model parameters are now inserted into a SIMULINK file, shown in figure (3.50), and the result plotted to validate the model in a closed loop time-domain validation test.

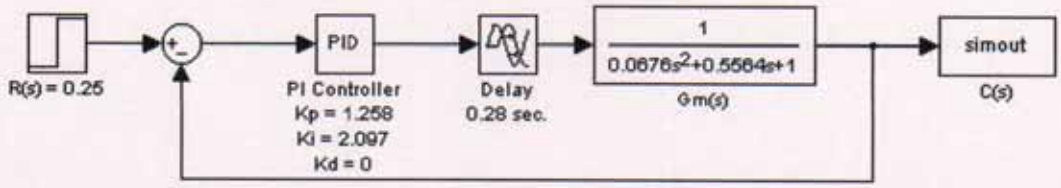

Figure (3.50) SIMULINK file used to validate model

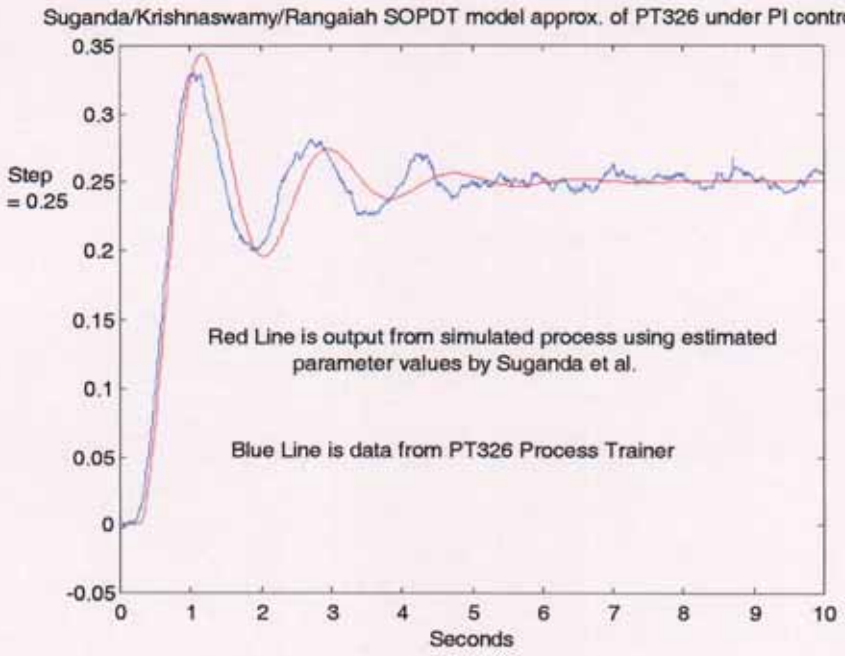

Figure (3.51) Comparison of PT326 closed loop step response and SOPDT model closed loop step response

The plot in figure (3.51) shows that the dynamics of the model, using the closed loop step response, captures the dynamics of the process very accurately. In a conference paper by Kealy and O'Dwyer (2002b) it is seen that the Suganda *et al.* (1998) method of identifying a model for a real process is one of the best methods of all those investigated. The paper by Kealy and O'Dwyer (2002b) can be viewed in Appendix 2 section 6.2, page A179.

In an alternative validation test, the open loop frequency response of the process and model are compared (figure (3.52)).

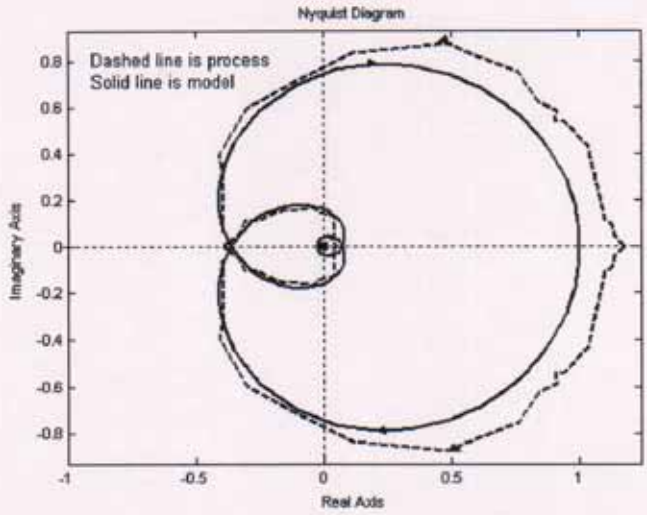

Figure (3.52) Open loop frequency plot comparison of SOPDT process and SOPDT process model

The plot in figure (3.52) show that the model matches the process at higher frequencies, indicating the estimate of the delay d_m, in particular, is accurate; this is confirmed by examining figure (3.51). There is a small error in the process model gain estimate, K_m, as the estimated value should be higher than that calculated.

The model obtained from the Suganda *et al.* (1998) identification technique is now compared with the model obtained from the Mamat and Fleming (1995) identification technique, in a validation test, by plotting the closed loop step response of the two models on the same plot as the closed loop step response of the process, demonstrated in figure (3.53).

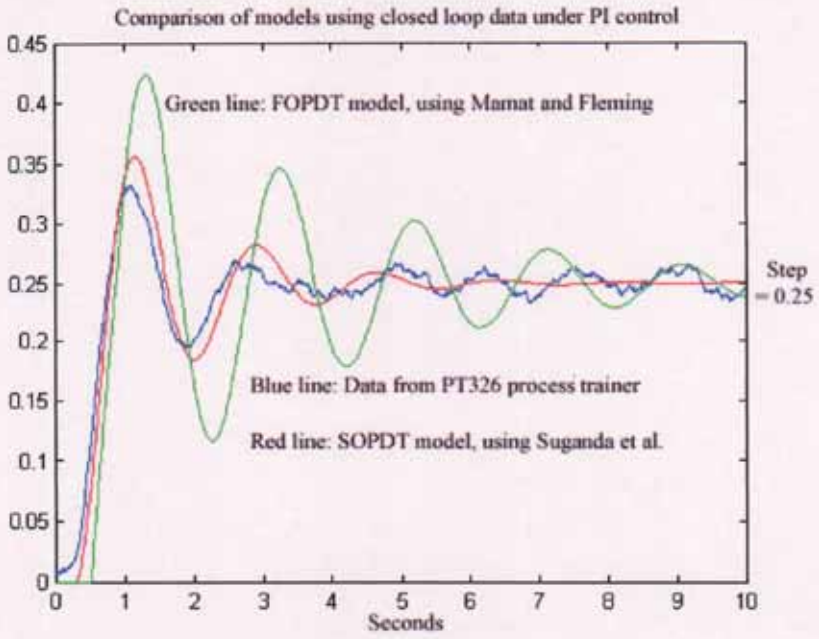

Figure (3.53) Comparison of SOPDT and FOPDT models

For each of the methods shown in figure (3.53), the process is in closed loop under PI control. Proportional gain, K_c, is set to 1.26 and the Integral time, T_i, is set to 0.6 seconds. The results demonstrate that the second-order-plus-dead-time model, as proposed by Suganda *et al.* (1998) is a better fit, in this example, to the actual process than the first-order-plus-dead-time model proposed by Mamat and Fleming (1995). This is confirmed in Appendix 2 section 3 where the PT326 process trainer and time-domain models step and frequency responses are compared. The reports are entitled "Comparisons_TD" and "Comparisons_FD". "Comparisons_TD" can be viewed in Appendix 2 section 3, pages A58 – A60, and "Comparisons_FD" on pages A61 – A69. The FOPDT model is shown in figure (TR_Id_17) on page A68, and the SOPDT model is shown in figure (TR_Id_19) on page A69.

3.4 Conclusions

In industrial applications, closed loop identification methods to determine a process model are often more desirable than open loop identification methods, as the process

can generally be left in production while the tests are being carried out. However, care must be taken to ensure there is minimal disturbance in the controlled variable while also ensuring that the process is sufficiently disturbed to obtain the required data. The algorithms to determine the closed loop models are quite complicated to implement. The three closed loop identification methods described worked reasonably well, both in simulation and implementation. The controller parameters influenced the accuracy of the models identified. The Suganda *et al.* (1998) method identified a reasonably accurate model for a simulated process but a very accurate model for the PT326 process trainer.

As stated at the end of section 3.3.1.2, a problem may arise because the tuning rules require the model parameters to be known in advance. As the accuracy of the initial values of the process model may be questionable, an iterative parameter estimation step may need to be implemented. This step should (1) start off with an initial model, (2) apply the tuning rules based on the initial model, (3) carry out closed-loop identification test using the controller parameters obtained by applying the tuning rules based on the initial model. The results of the updated model parameters in (3) are subsequently used to tune the controller to carry out a further identification test to determine a more accurate process model. This activity is repeated until the model is deemed to be an accurate representation of the process.

Chapter 4 : Closed Loop Relay Based Identification of a Process Model

4.1 Introduction

An important technique of identifying a model for a process is based on an "ultimate cycle" type of experimental design. Probably the most successful part of this approach, introduced by Ziegler and Nichols (1943), is *not* the PI or PID tuning rule. Rather, it is the identification procedure: a way to find important process information, the ultimate gain (K_u) and ultimate frequency (ω_u). This is often done by trial-and-error (Seborg *et al.*, 1989; Luyben and Luyben, 1997). A typical approach can be summarised as follows (Seborg *et al.*, 1989):

1. Set the controller gain (K_c) at a low value, perhaps 0.2.
2. Put the controller in automatic mode.
3. Make a small change in the setpoint or load variable and observe the response. If the gain is low the response will be sluggish.
4. Increase the gain by a factor of two and make another setpoint or load change.
5. Repeat step 4 until the loop becomes oscillatory and continuous cycling is observed. The gain at which this occurs is the ultimate gain (K_u) and the period of oscillation is the ultimate period ($P_u = 2\pi/\omega_u$).

This is a simple and reliable approach to obtain K_u and ω_u. The disadvantages are also obvious: it is time consuming, the process is driven unstable and there are no limitations on the amplitude of the 'limit cycle'. An alternative is the relay feedback test proposed by Astrom and Hagglund (1984). This test may be used to find one or more points on the frequency response of a process. The method involves the introduction of a relay element in parallel with the controller. The relay is switched in when process parameter estimation or controller tuning is required. A continuous cycling of the controlled variable is generated from the relay feedback experiment

and the approximations for K_u and ω_u, labelled $\hat{K}_u$ and $\hat{\omega}_u$, can be extracted directly from the experiment. The information obtained from the relay feedback experiment is exactly the same as that from the continuous cycling method. However, an important difference is that the sustained oscillation is generated in a *controlled* manner (i.e., the magnitude of oscillation can be controlled) in a relay feedback test. Moreover, in virtually all cases, this is a very efficient method, i.e. a one-shot solution, to generate a sustained oscillation. Applications of Astrom-Hagglund autotuners are found throughout process industries using single station controllers or a distributed control system. Table (4.1) shows the trend a decade ago where major vendors provide autotuners in their products (Hang *et al.*, 1993). Process model identification methods for these products include those based on open or closed loop step tests (step), responses based on a ramp input (ramp), relay feedback autotuners (relay) and responses based on a pseudo-random binary signal (PRBS) input (Yu, 1999).

Manufacturer	Identification method
Bailey Controls	Step
Control Techniques	Ramp
Fisher Controls	Relay
Foxboro	Step
Fuji	Step
Hartmann & Braun	Step
Honeywell	Step
Satt Control	Relay
Siemens	Step
Toshiba	PRBS
Turnbull Control Systems	Step
Yokogawa	Step

Table (4.1) Autotuners from different vendors (Yu, 1999)

The success of the relay feedback autotuner is due to the fact that the identification and tuning mechanism is so *simple* that operators understand how it works.

Moreover, it also works well with slow and highly non-linear processes (Luyben, 1987). Over the past decade, extensive research has been carried out on relay feedback autotuners. Refinements on the accuracy of results and improvements on the experimental design have been made. Discussions about potential problems, extensions to multivariable systems and incorporation of gain scheduling were also reported. It is a widely held view that the relay feedback based autotuners now can provide the necessary tools to improve control performance in a reliable way (Yu, 1999).

This chapter discusses process model identification methods using an ideal relay, a relay with bias and a relay with hysteresis, in turn, in a relay autotuner.

4.2 Ideal Relay Feedback Identification Method

As mentioned, Astrom and Hagglund (1984) suggested the relay feedback test to generate sustained oscillation as an alternative to the conventional continuous cycling technique. It is very effective in approximately determining the ultimate gain and ultimate frequency. The distinct advantages of the relay feedback test are (Yu, 1999):

1. It identifies process information around the important frequency, the ultimate frequency (the frequency where the phase angle is $-\pi$).
2. It is a closed loop test; therefore the process will not drift away from the nominal operating point.
3. For processes with a long time constant, it is a more time-efficient method than conventional step or pulse testing. The experimental time roughly equals $2 \sim 4$ times the ultimate period.

Experimental Design: Consider a relay feedback system where $G(s)$ is the process transfer function, y is the controlled output, y^{set} is the setpoint, e is the error and u is the manipulated input [figure (4.1)]:

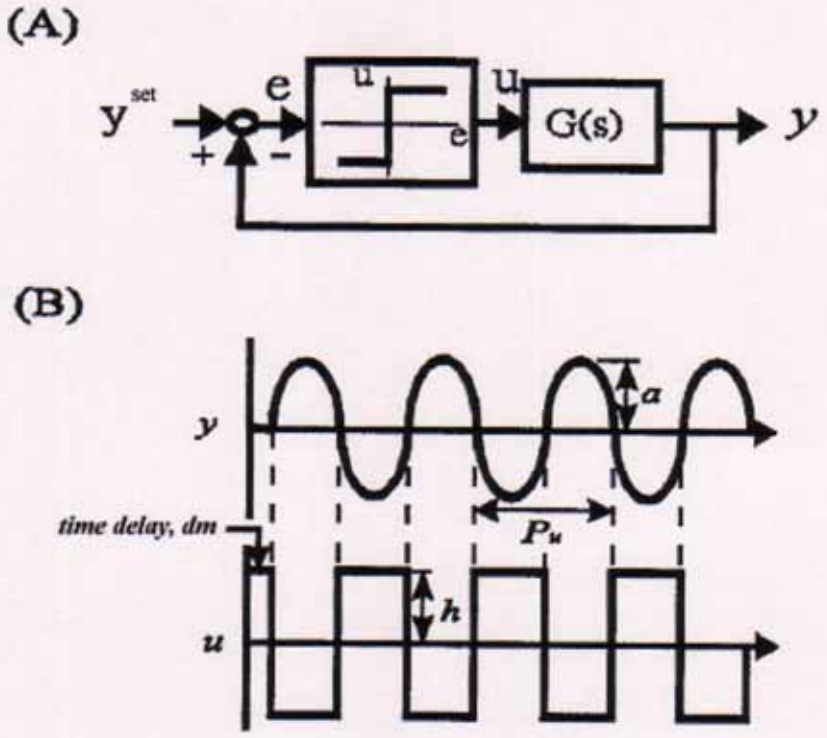

Figure (4.1) (A) Block diagram for a relay feedback system and (B) Relay feedback test for a process with positive steady state gain (Yu, 1999)

A relay of magnitude h is inserted in the feedback loop. Initially, the input u is increased by h. As the output y starts to increase (after a time delay d_m), the relay switches to the opposite position, $u = -h$. Since the phase lag is $-\pi$, a limit cycle with a period $\hat{P}_u$ results (figure (4.1)). The period of the limit cycle is approximately the ultimate period. Therefore, the approximate ultimate frequency from this relay feedback experiment is:

$$\hat{\omega}_u = \frac{2\pi}{\hat{P}_u} \tag{4.1}$$

From the Fourier series expansion, the amplitude a can be considered to be the result of the primary harmonic of the relay output. Therefore, the ultimate gain can be approximated as (Ogata, 1970; Astrom and Hagglund, 1984):

$$\hat{K}_u = \frac{4h}{\pi a} \tag{4.2}$$

$\hat{K}_u$ and $\hat{\omega}_u$ can be used directly to find controller settings.

The relay feedback test can be carried out manually (without any auto-tuner). The procedure requires the following steps:

1. Bring the system to steady state.
2. Make a small increase in the manipulated input. The magnitude of change depends on the process sensitivities and allowable deviations in the controlled output. Typical increases in manipulated input values are between 3 ~ 10%.
3. As soon as the output crosses the set point, the manipulated input is switched to the opposite direction (e.g. -5% change from the original value).
4. Repeat step 2 until sustained oscillation is observed (figure (4.1)).
5. Read off ultimate period $\hat{P}_u$ from the limit cycle and compute $\hat{K}_u$ from equation (4.2).

This procedure is relatively simple and efficient. Physically, it implies you move the manipulated input *against* the process (Yu, 1999).

4.3 Approximate Process Transfer Function Determination

After the relay feedback experiment, the estimated ultimate gain ($\hat{K}_u$), and estimated ultimate frequency ($\hat{\omega}_u$) can be used directly to calculate controller parameters. Alternatively, it is possible to back-calculate the approximated process transfer functions. The other data useful in finding the transfer function are the time delay (d_m) and/or the steady-state gain (K_m). The time delay d_m can be easily determined from the initial part of the relay feedback test as shown in figure (4.1).

4.3.1 FOPDT Model; Simulation; "Simple" Approach:

As an indicator of the accuracy of this method, a relay-based test is carried out using the MATLAB/SIMULINK software and the results compared with the "known" parameters.

The SIMULINK file in figure (4.2) is used for this part of the experiment.

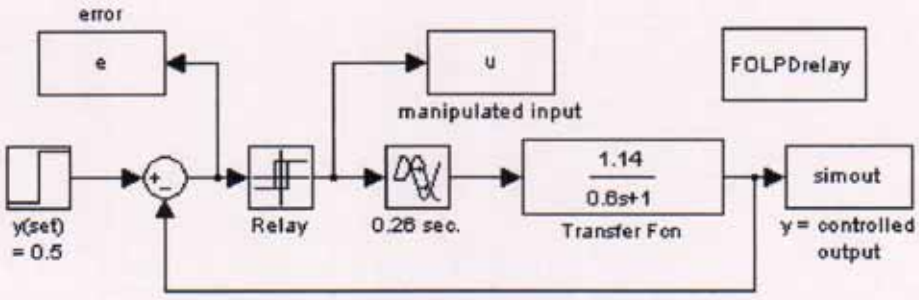

Figure (4.2) SIMULINK file used in relay based experiment

The **Relay** settings in figure (4.2) are shown in figure (4.3).

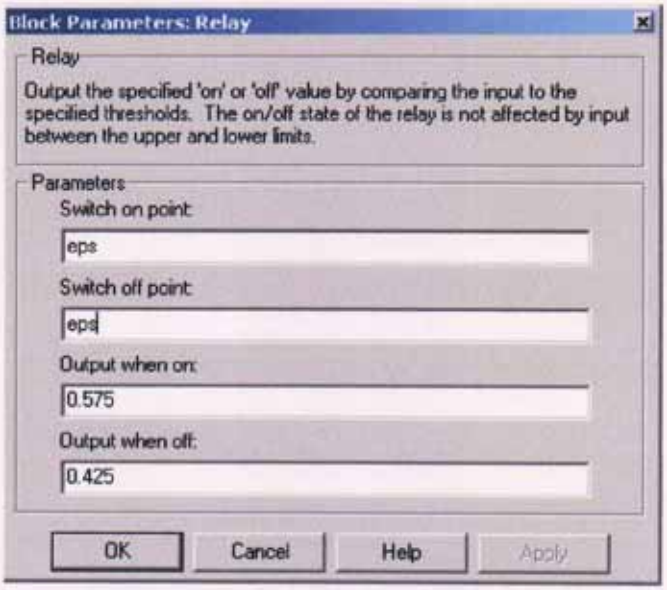

Figure (4.3) Ideal Relay parameter settings

The SIMULINK file shown in figure (4.2) is run for six seconds and the results are plotted and shown in figure (4.4). The switch on and switch off points are the same, i.e. at the zero line. This means that the relay is the ideal relay, no hysterisis. The relay output when "On" is 0.575 and the output when "Off" is 0.425. This gives the small change in manipulated input required to obtain the relevant information.

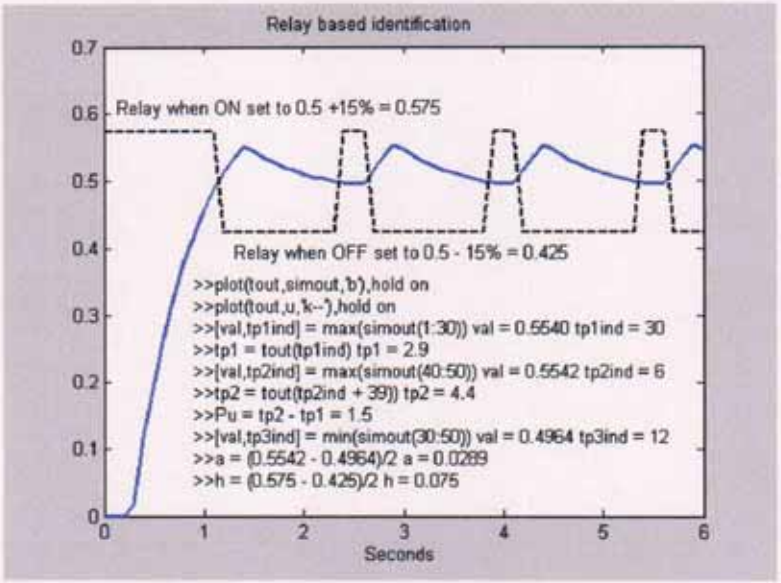

Figure (4.4) Plot of controlled O/P (Blue solid line) and manipulated I/P (Black --)

From the plot in figure (4.4), the ultimate period estimate, $\hat{P}_u$, is 1.5 seconds. The ultimate frequency estimate, $\hat{\omega}_u$, from equation (4.1) = $(2\pi)/\hat{P}_u$ = 4.189 radians/second.

The ultimate gain estimate $\hat{K}_u$ from equation (4.2) = $(4h)/(\pi a)$ = 3.3.

The time delay, d_m, read off from figure (4.4) is 0.29 seconds. Using the "**simple" approach**, the parameter values for the **FOPDT** model shown in equation (1.1) are determined as follows. The time constant, τ_m, is found using equation (4.3). (Note the

large values obtained if $d_m \times \hat{\omega}_u$ is close to $\pi/2$). In equation (4.3), d_m must be known *a priori* (Yu, 1999).

$$\tau_m = \frac{\tan(\pi - d_m \hat{\omega}_u)}{\hat{\omega}_u} \tag{4.3}$$

Inserting the values, τ_m is found to be –0.642 seconds. Clearly, the simple approach does not allow the estimation of appropriate FOPDT model parameters in this case. A variation of this method, which also fails to give adequate model parameters, is inserted into Appendix 2 section 4, page A71, entitled "FOPDT_SA_1".

4.3.2 FOPDT Model; Implementation; "Simple" Approach:

The file in figure (4.5) is used in implementing the relay-based techniques on the PT326 process trainer. The "Manual Switch" is in the position shown. The "Step-size" is set to 0.05 seconds in the Simulation/Parameters settings. This gives a sampling rate of 20 samples/second that allows the signal to be reconstructed accurately. The set-point is set to 0.25. This ensures that the process variable stays within the linear range of the system. The initial tests are carried out using an **ideal On-Off relay** i.e. no hysteresis, no bias. The relay output when on is set to 0.5 and when off is set to –0.5. These values ensure that the process variable cycles continuously as shown in figure (4.7).

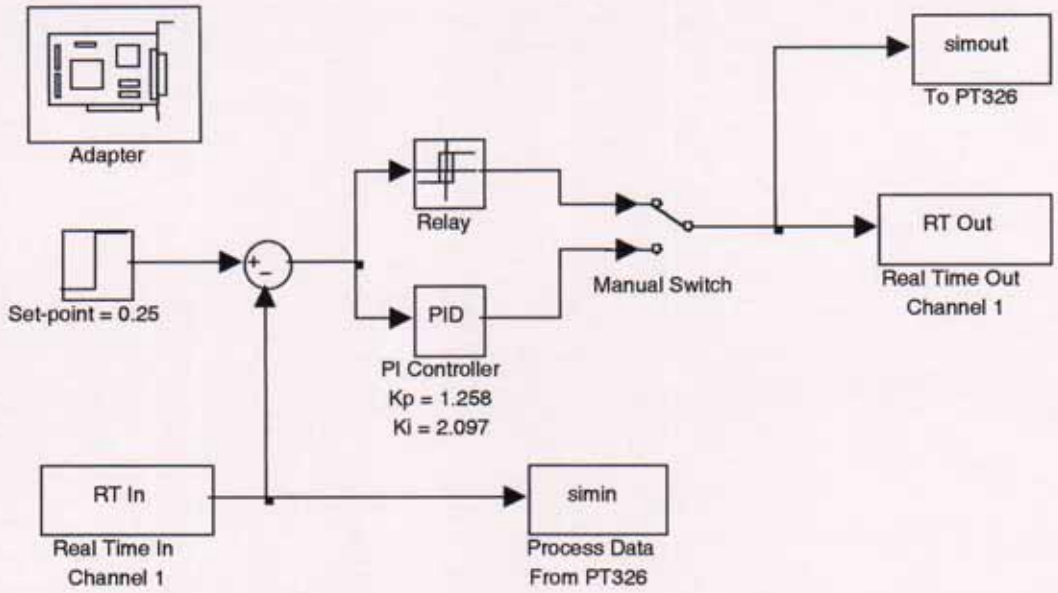

Figure (4.5) MATLAB/SIMULINK/HUMUSOFT file used for Relay-Based Identification

The "Relay" characteristics are set as 0.5 when "ON" and –0.5 when "OFF" with no hysteresis, i.e. both switch-on and switch-off points are at zero (eps), see figure (4.6).

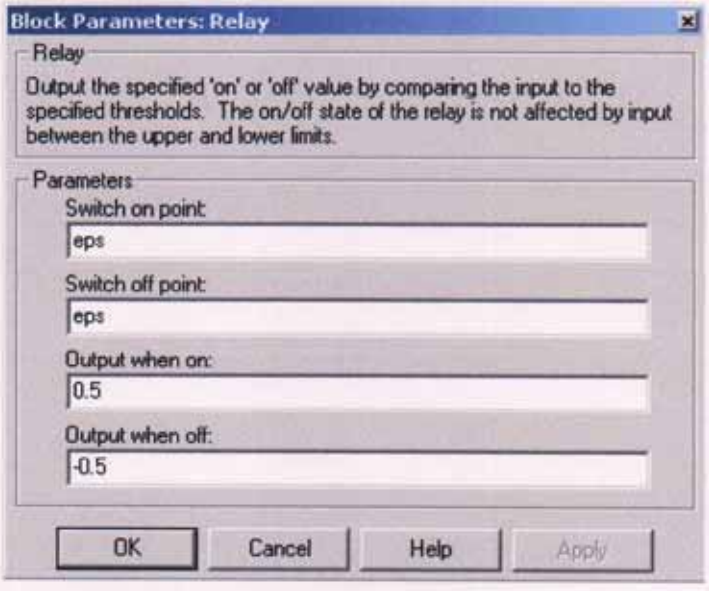

Figure (4.6) Relay settings

The program is run for 10 seconds and the resulting data is shown in figure (4.7).

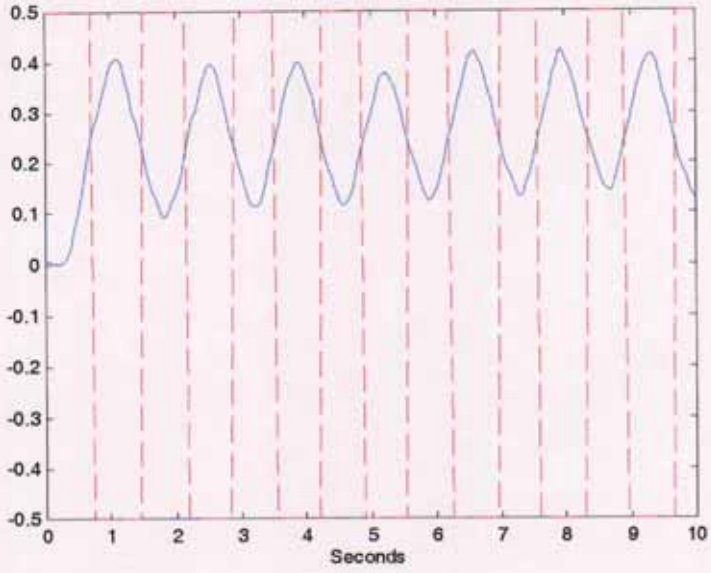

Figure (4.7) Process output (Blue) and Relay output (Red--)

The MATLAB commands used to generate the plot in figure (4.7) and determine the approximate ultimate frequency, ultimate gain and the parameters of the FOPDT model for this data, using the "simple" approach is shown in the program entitled "Relay_1" in Appendix 2 section 4, page A73. This results in the parameter estimates of the FOPDT model using the ideal On/Off relay method (simple approach) as follows:

- Model gain, K_m, = 0.78
- Model time constant, τ_m, = 0.72 seconds
- Model time delay, d_m, = 0.40 seconds

The actual process parameters, as shown in figure (4.2), are:

- Process gain, K_p, = 1.14
- Process time constant, τ_p, = 0.6 seconds
- Process time delay, d_p, = 0.26 seconds

Validation: As a test of the accuracy of the estimated parameters, an open loop step test and a frequency response plot is carried out on the real process and the FOPDT model of the process.

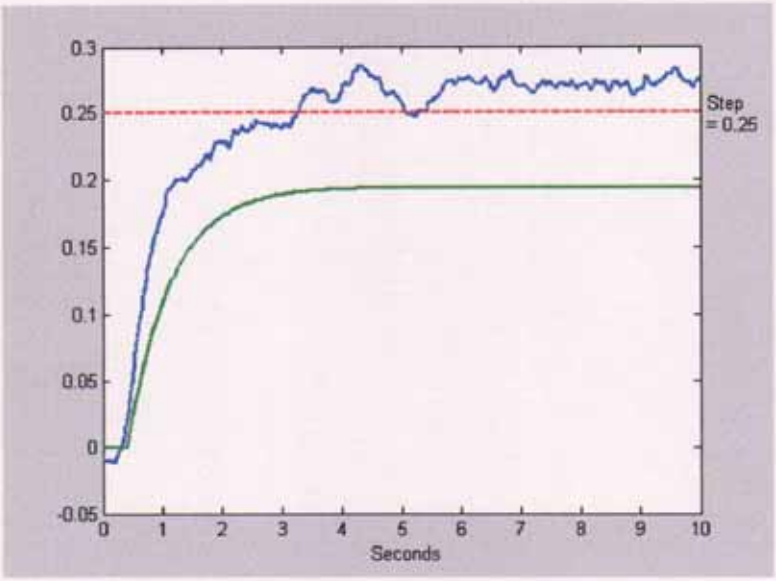

Figure (4.8) Open loop step response of PT326 (Blue) and FOPDT model of PT326 (Green) using the ideal On/Off relay identification technique (simple approach)

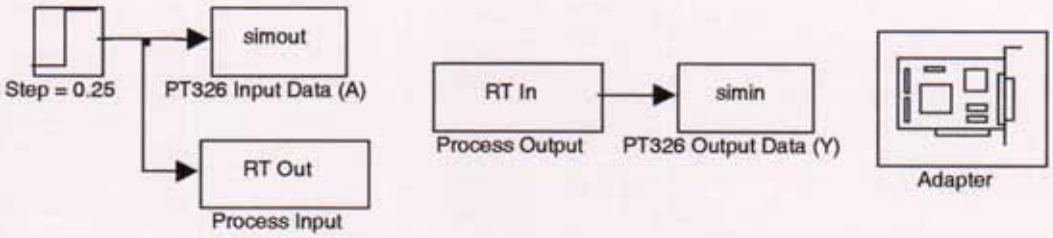

Figure (4.9) MATLAB/SIMULINK/HUMUSOFT file used to generate PT326 process trainer step response in figure (4.8)

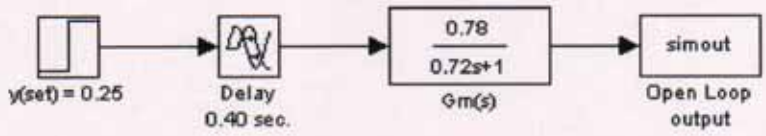

Figure (4.10) MATLAB/SIMULINK model of PT326 used to generate model open loop step response in figure (4.8)

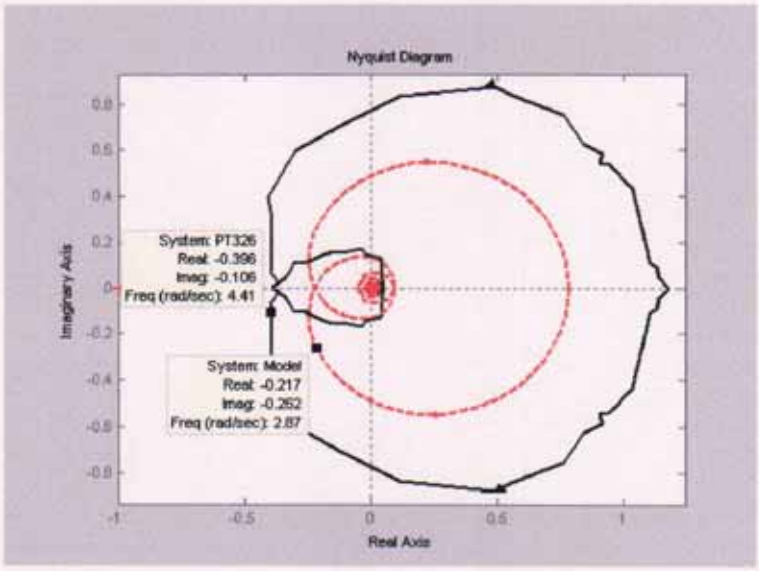

Figure (4.11) Nyquist plot of Process Trainer PT326 (Black line) and model of PT326 (Red-- line) using ideal On/Off relay identification (simple approach)

The labels on the plot in figure (4.11) have no significance except to highlight the two different systems.

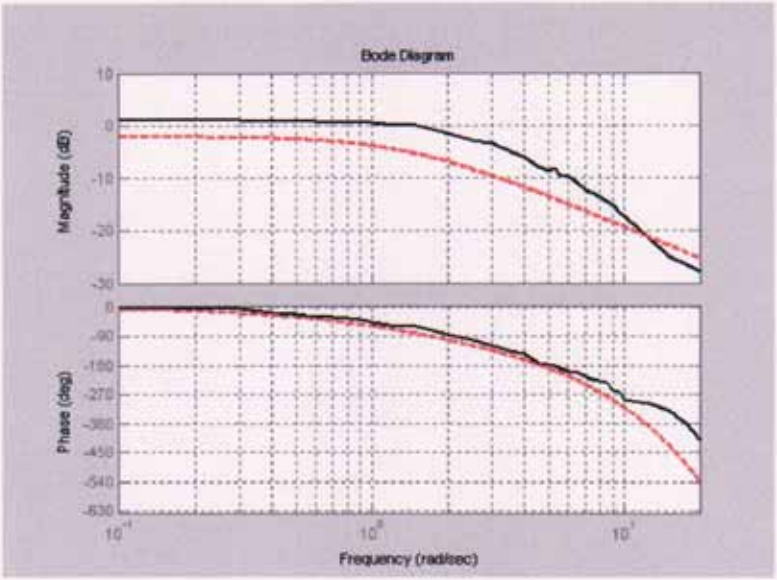

Figure (4.12) Bode plot of Process Trainer PT326 (Black) and FOPDT model of PT326 (Red--) using ideal On/Off system identification technique (simple approach)

The plots in figures (4.8), (4.11) and (4.12) show that the process model gain estimate, K_m, is not an accurate estimation of the process gain, K_p. The model time delay estimate, d_m, is a reasonably accurate estimate of the process time delay.

4.3.3 SOPDT Model; Simulation; "Simple" Approach:

It was decided to attempt to determine the parameters of the second order plus dead time model (equation (1.3)) using the "simple method". It is hoped that this model would give more accurate results than the FOPDT model experiments. The process whose parameters are being estimated is:

- Process gain, K_p, = 1.14
- Process time constant, τ_p, = 0.6 seconds
- Process time delay, d_p, = 0.26 seconds

Equations (4.4) and (4.5) are used for this purpose (Yu, 1999). The values of K_m and d_m must be known in advance.

$$-\pi = -\omega_u d_m - \tan^{-1}(\omega_u \tau_1) - \tan^{-1}(\omega_u \tau_2) \qquad (4.4)$$

$$\frac{1}{\hat{K}_u} = \frac{K_m}{\sqrt{(1+(\omega_u \tau_1)^2)(1+(\omega_u \tau_2)^2)}} \tag{4.5}$$

Previously determined values for d_m and K_m are used in order to solve for the two time constants τ_1 and τ_2 (d_m = 0.29 seconds, K_m = 0.87). The value of K_m = 0.87 is based on the assumption of using the absolute value of τ_m in the estimation of a FOPDT process model (discussed in Appendix 2 section 4, page A71, in file "FOPDT_SA_1"). There may, of course, be a problem with this assumption. Alternatively, the model gain, K_m, could be obtained using an open- or closed-loop test and this value used in the subsequent calculations. Yu (1999) suggest that if K_m is not available (or inaccurate), we can perform a second relay feedback test (Li *et al.*, 1991) or use a biased relay to find additional information. However, the method is explained using the value of K_m = 0.87.

Firstly, equation (4.5) is manipulated so that τ_1 can be written in terms of τ_2. Secondly, this solution of τ_1 is inserted into equation (4.4) so that the only unknown in equation (4.4) is τ_2. Thirdly, a solution to equation (4.4) is then determined for τ_2. The same steps are carried out with τ_1 as the dependent variable and the results recorded. The software package *Mathematica* by Stephen Wolfram (1996) is used to find solutions and manipulate the equations. The result of these manipulations are:

$$\tau_1 = \sqrt{\frac{\left(\frac{8.24}{1+17.55\tau_2^2}\right)-1}{17.55}} \tag{4.6}$$

and

$$\tau_2 = \sqrt{\frac{\left(\frac{8.24}{1+17.55\tau_1^2}\right)-1}{17.55}} \tag{4.7}$$

The Mathematica files entitled "τ1_Result" and "τ2_Result" in Appendix 2 section 4, page A73, determine the values as follows:

- τ_1 = -0.64 seconds
- τ_2 = -0.64 seconds

The negative values of the time constants obtained show the difficulty in applying the method. One problem appears to be that the time constant values are sensitive to the process model gain estimate, K_m. As a demonstration of this, different values for the process model gain, K_m, are inserted into the Mathematica (Wolfram, 1996) file and the solutions recorded. The other calculated values are kept constant, i.e. $d_m = 0.29$, $\hat{K}_u = 3.3$ and $\hat{\omega}_u = 4.189$. A representative Mathematica command used is as follows:

```
Solve[Tan[-1.926782654] == Tan[-(ArcTan[(4.189) x (Sqrt[((23.3444/(1+17.547721 x τ2^2)) - 1)/17.547721])] - (ArcTan[(4.189) x (τ2)]))], τ2]
```

Table (4.2) demonstrates the results.

Km	Solution 1	Solution 2	Solution 3
0.90	-0.67	0.33-0.075i	0.33+0.075i
1.00	-0.75	0.25	0.49
1.05	-0.79	0.23	0.54
1.10	-0.83	0.22	0.60
1.15	-0.87	0.21	0.64
1.20	-0.91	0.20	0.69
1.25	-0.95	0.19	0.73
1.40	-1.07	0.17	0.86

Table (4.2) Solutions for τ_1 and τ_2 using different values for process model gain, K_m

From the results in Table (4.2), the two relevant solutions for τ_1 and τ_2 are the two positive solutions, Solution 2 and Solution 3.

The **simple** method for approximating parameters of the transfer function, as shown previously, has inaccuracies. The "**improved" algorithm** is now investigated.

4.3.4 FOPDT Model; Simulation; "Improved" Algorithm:

Equation (4.8) is proposed by Yu (1999) to determine the time constant, τ_m, for the FOPDT system shown in equation (1.1).

The process whose parameters are being estimated, shown in figure (4.2), is:

- Process gain, K_p, = 1.14
- Process time constant, τ_p, = 0.6 seconds
- Process time delay, d_p, = 0.26 seconds

Note that for these calculations, d_m = 0.29 seconds and $\hat{\omega}_u$ = 4.189 radians/second as determined earlier for the simulated process shown in figure (4.2).

$$\tau_m = \frac{\pi}{\omega_u \log_e\left(2\exp\left(\frac{d_m}{\tau_m}\right)-1\right)} \tag{4.8}$$

By cross-multiplying in equation (4.8), equation (4.9) is generated and plotted using Mathematica. The Mathematica commands are in Appendix 2 section 4, page A73, entitled "Relay_2".

$$\pi = (\tau_m)(\omega_u)\left(\log_e\left(2Exp\left(\frac{d_m}{\tau_m}\right)-1\right)\right) \tag{4.9}$$

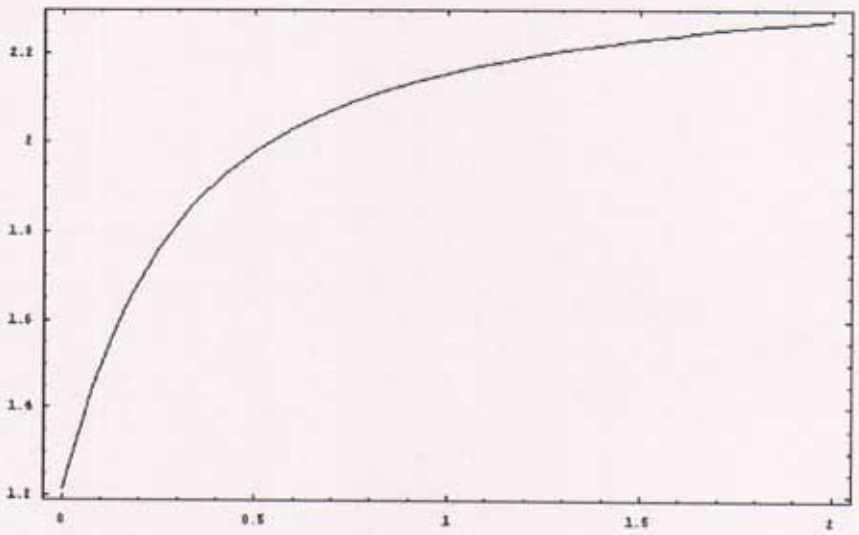

Figure (4.13) Plot of right-hand-side of equation (4.9)

No solution was found for τ_m so further investigation is needed. In another test, the ultimate frequency is changed to 5.25 radians/seconds and d_m, the time delay term is changed to 0.26 seconds. The Mathematica file entitled "Relay_3" in Appendix 2 section 4, page A74, attempts to solve the equation. However, the "FindRoot_1" file in Appendix 2 section 4, page A74, failed to converge and so a solution was not realised with the new parameter values, either.

In another test to find the solution to equation (4.8), the values $d_m = 0.4$ seconds and $\hat{\omega}_u = 4.6542$ radians/second are inserted into a Mathematica file named "Relay_4" (in Appendix 2 section 4, page A74) and the results are recorded. Equation (4.8) is now solved to yield $\tau_m = 0.89$ seconds. Clearly, the improved algorithm does not work satisfactorily.

Note: It is noted by Yu (1999) that the "Improved Algorithm" Ultimate Gain/Ultimate Period identification method works well for first order plus dead time systems with a long time constant, i.e., small d_m/τ_m value, less than approximately 0.28. In the case of the model parameter values used in this experiment, the ratio is approximately 0.26/0.6 = 0.43. This is a possible source of error with the parameter calculations.

4.3.5 SOPDT Model; Simulation; "Improved" Algorithm:

To determine the two unequal time constants for the second order system plus dead time process model shown in equation (1.3), equation (4.10) is used (Yu, 1999):

$$\tau_1\left(\frac{2\exp\left(-\frac{m\pi}{\tau_1\omega_u}\right)}{1+\exp\left(-\frac{\pi}{\tau_1\omega_u}\right)}\right)-\tau_1=\tau_2\left(\frac{2\exp\left(-\frac{m\pi}{\tau_2\omega_u}\right)}{1+\exp\left(-\frac{\pi}{\tau_2\omega_u}\right)}\right)-\tau_2 \qquad (4.10)$$

where

$$m=1-\frac{d_m\times\omega_u}{\pi} \qquad (4.11)$$

The process whose parameters are being estimated is the process in figure (4.2) as shown: $G_p(s) = \frac{1.14e^{-0.26s}}{0.6s+1}$. The values for equation (4.11) are known for this case, therefore m = 0.61 (d_m = 0.29 seconds and $\hat{\omega}_u$ = 4.189 radians/second for the simulated process in figure (4.2)). Equation (4.10) is a non-linear equation. By using Mathematica, each side of the equation can be plotted using values of τ_1 (and τ_2) between 0 and 1.2 seconds. From the result shown in figure (4.14), it is seen that there are many solutions to equation (4.10).

The Mathematica commands for plotting the left-hand-side of equation (4.10) is entitled "Relay_5" and is in Appendix 2 section 4, page A75.

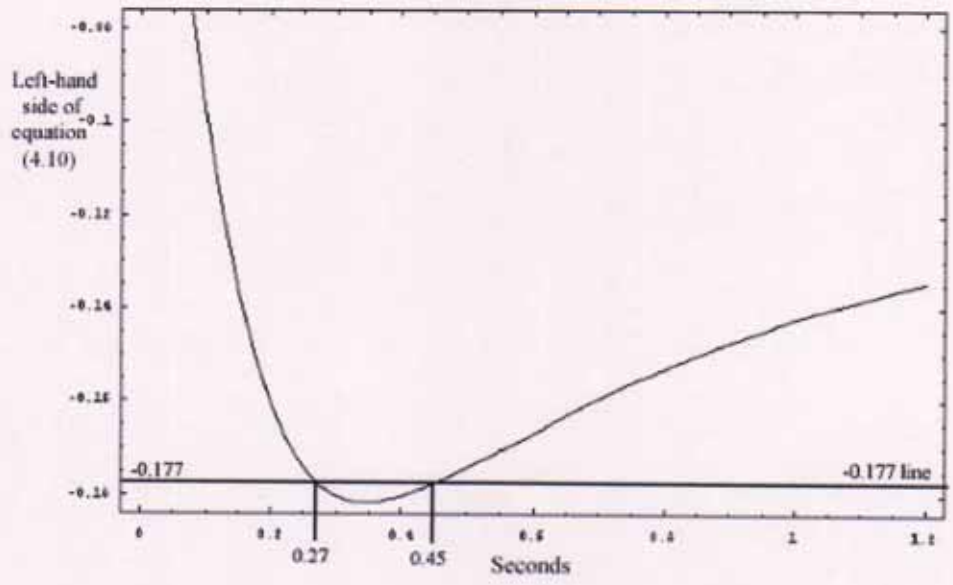

Figure (4.14) Mathematica plot of left-hand-side of equation (4.10)

As a check on the plot in figure (4.14), a line is drawn where the left-hand side of equation (4.10) equals –0.177 and the points where the line intersects the graph are the two time constants. The reason for this is explored below. These intersections correspond to τ_1 = 0.45 seconds and τ_2 = 0.27 seconds. The values are inserted into equation (4.10) and the results recorded as follows:

$$LHS(\text{equation}(4.10)) = (0.45)\times\left(\frac{2\times Exp\left(-\frac{1.93}{1.89}\right)}{1+Exp\left(-\frac{\pi}{1.89}\right)}\right)-(0.45) = -0.178 \qquad (4.12)$$

$$RHS(\text{equation}(4.10)) = (0.27)\times\left(\frac{2\times Exp\left(-\frac{1.93}{1.13}\right)}{1+Exp\left(-\frac{\pi}{1.13}\right)}\right)-(0.27) = -0.178 \qquad (4.13)$$

Equations (4.12) and (4.13) indicates that the values of τ_1 and τ_2 determined are the correct values. Examining equation (4.5) (repeated below),

$$\frac{1}{\hat{K}_u} = \frac{K_m}{\sqrt{(1+\left(\hat{\omega}_u\tau_1\right)^2)(1+\left(\hat{\omega}_u\tau_2\right)^2)}}$$

the values $\tau_1 = 0.45$, $\tau_2 = 0.27$, $\hat{\omega}_u = 4.189$ rads/sec and $\hat{K}_u = 3.3$ are inserted into equation (4.5) ($\hat{K}_u$ and $\hat{\omega}_u$ are determined in section 4.3.1). This gives a value for $K_m = 0.98$. The four parameters of the SOPDT model in equation (1.3) are then as follows:

- Model gain, K_m, = 0.98
- Model time constant 1, τ_1, = 0.45 seconds
- Model time constant 2, τ_2, = 0.27 seconds
- Model time delay, d_m, = 0.29 seconds

Validation: The model is now validated by comparing the process open loop step response with the model open loop step response, shown in figure (4.15).

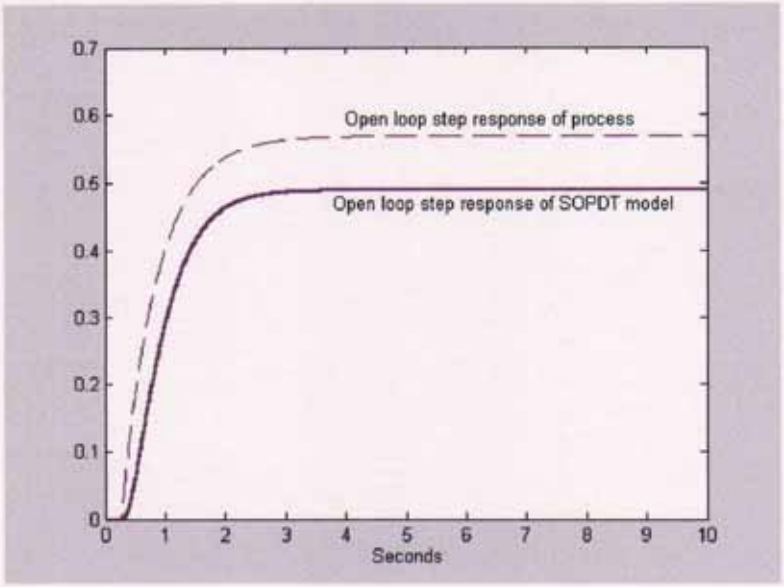

Figure (4.15) Open loop step response of process and SOPDT model of process using the Improved Algorithm relay based method

Figure (4.15) shows that the improved algorithm relay based method of system identification gives a reasonable estimate of the delay, in particular. The process model gain estimate, K_m, is not accurate; this reinforces the argument for estimating the gain in a second test e.g. an open loop step test.

4.4 Improved Relay Feedback

Luyben (1987) pioneered the use of relay feedback tests for system identification. The ultimate gain and ultimate frequency estimates from the relay feedback tests are used to fit a typical transfer function (e.g., first-, second- or third order plus time delay process model). However, as shown in the earlier part of this chapter, this can lead to significant errors in the process model parameter estimates for typical transfer functions in a control system, arising from the approximations of the ultimate gain and ultimate frequency determined. The error comes from the linear approximation (describing function analysis) to a non-linear element. The square wave output from the relay is approximated with the principle harmonic from the Fourier Transform (Atherton, 1982; Chang *et al.*, 1992) and the ultimate gain estimate is computed

accordingly. Several attempts have been made to overcome the inaccuracy. Li *et al.* (1991) use two relay tests to improve the estimation of $\hat{K}_u$ and $\hat{\omega}_u$. Chang *et al.* (1992) employ a discrete time approach to give a better estimation of $\hat{\omega}_u$. In these suggestions, an ideal relay is employed in the experiments and modifications are made *afterwards*. Since the source of the errors comes from a sine-wave approximation of a square-wave oscillation, a straightforward approach to overcome the inaccuracy is to re-design the experiment (instead of taking remedial action afterward) i.e. to produce a more sinewave like controlled signal using a different type of relay (Yu, 1999).

4.4.1 Relay With Bias

If an unbiased relay autotuner is used, then the resulting process input and output signals have zero DC components and the process gain estimation is thus difficult. To overcome this problem, the biased relay as shown in figure (4.16) is introduced. The resulting oscillation waveforms of the process output are shown in figure (4.17); DC components have been created in the waveforms (Wang *et al.*, 1999). For a relay with bias (and no hysteresis), the parameter ε in figure (4.16) is set to zero.

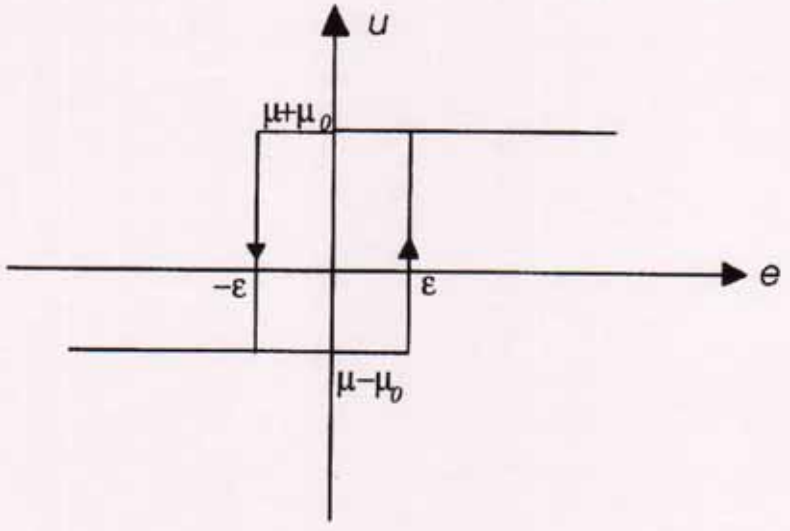

Figure (4.16) Biased Relay (Wang *et al.*, 1999)

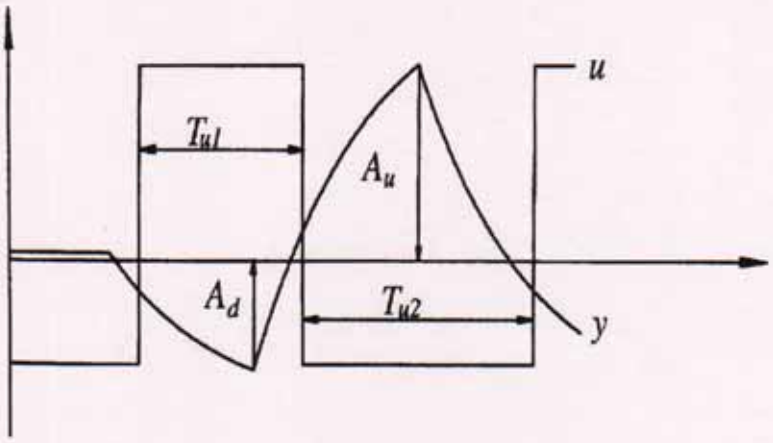

Figure (4.17) Oscillatory Waveforms under a Biased Relay Feedback (Wang *et al.*, 1999)

The steady state gain of the first-order plus dead-time process model (K_m in equation (1.1)) can be computed (Ramirez, 1985) via the following formula:

$$K_m = G_p(0) = \frac{\int_0^{T_{u1}+T_{u2}} y(t)dt}{\int_0^{T_{u1}+T_{u2}} u(t)dt} \tag{4.14}$$

With K_m known, the normalised dead time of the process, $\theta = d_m/\tau_m$, is obtained from equation (4.15) or equation (4.16) (Wang *et al.* (1999)):

$$\theta = \ln \frac{(\mu_0 + \mu)K_m - \varepsilon}{(\mu_0 + \mu)K_m - A_u} \tag{4.15}$$

or

$$\theta = \ln \frac{(\mu_0 - \mu)K_m - \varepsilon}{(\mu_0 - \mu)K_m + A_u} \tag{4.16}$$

It then follows that (Wang *et al.* (1999))

$$\tau_m = T_{u1}\left(\ln \frac{2\mu K_m e^{\theta} + \mu_0 K_m - \mu K_m + \varepsilon}{\mu K_m + \mu_0 K_m - \varepsilon} \right)^{-1} \tag{4.17}$$

or

$$\tau_m = T_{u1}\left(\ln \frac{2\mu K_m e^{\theta} - \mu_0 K_m - \mu K_m + \varepsilon}{\mu K_m - \mu_0 K_m - \varepsilon} \right)^{-1} \tag{4.18}$$

The dead time is thus

$$d_m = \tau_m \theta \tag{4.19}$$

4.4.1.1 FOPDT Model Identification; Implementation:

The biased relay experiment is implemented on the PT326 process trainer. The process input *u(t)* and output *y(t)* are recorded, and the periods and the amplitudes of the oscillations are measured. The following step-by-step procedure is then followed:

1. Compute K_m from equation (4.14).
2. Compute θ from equation (4.15) or equation (4.16).
3. Compute τ_m from equation (4.17) or equation (4.18).
4. Compute d_m from equation (4.19).

The file in figure (4.5) is again used for identification i.e. the parameters of the PT326 process trainer are identified but now the relay characteristics are as shown in figure (4.18). The output when "ON" is set to 0.3 and the output when "OFF" is set to 0.2 to ensure that the relay is biased. The parameter ε is set to zero (eps), i.e. no hysteresis.

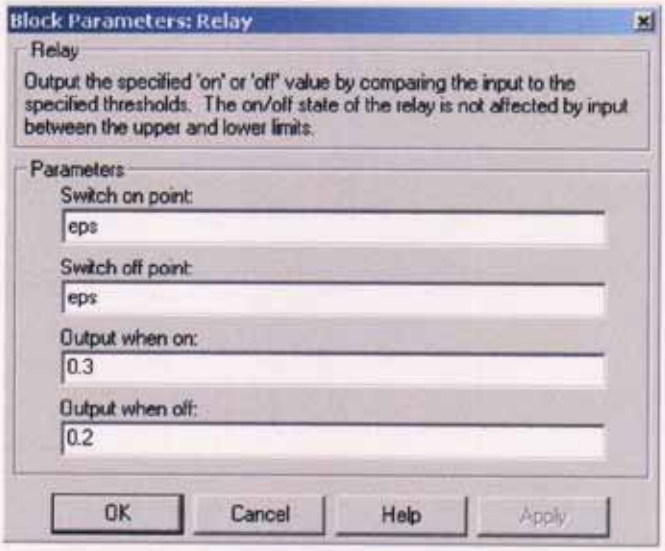

Figure (4.18) Biased Relay Settings

The resulting relay output and closed loop system output are shown in figure (4.19).

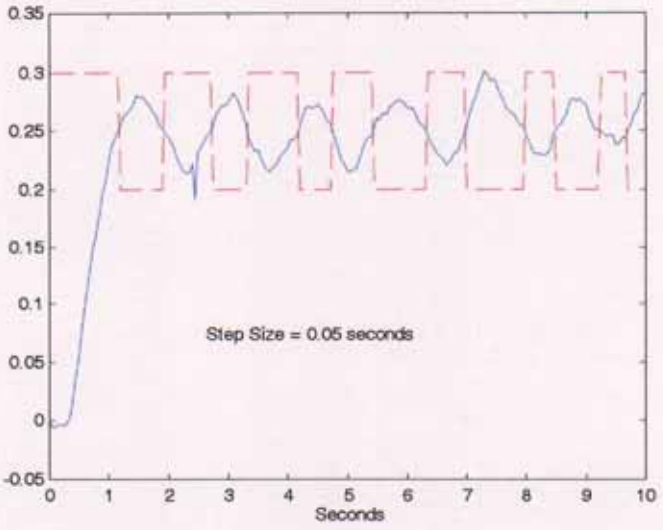

Figure (4.19) Relay based experiment using biased relay: outputs

The MATLAB commands to determine K_m, A_u and A_d (See figure (4.17)) are shown in the program "Relay_6" in Appendix 2 section 4, page A75, with the following results:

- $K_m = 0.68$
- $A_u = 0.032$
- $A_d = 0.035$

To determine μ and μ_0 in figure (4.16) using the biased relay settings in figure (4.18), the following simultaneous equations, (4.20) and (4.21), are solved.

$$\mu + \mu_0 = 0.3 \tag{4.20}$$

$$\mu - \mu_0 = 0.2 \tag{4.21}$$

Therefore, $\mu = 0.25$ and $\mu_0 = 0.05$. From figure (4.19), the value of T_{u1} in figure (4.17) is 0.8. The normalised dead time of the process, d_m/τ_m, is computed from equation (4.15) or (4.16) as 0.17. The time constant, τ_m, is found by using equation (4.17) or (4.18) as 2.97 seconds. The time delay, d_m, is determined from equation (4.19) as 0.51 seconds. This results in the parameter estimates of the FOPDT process model, for the PT326 process trainer, using the described methods as follows:

- Model gain, K_m, = 0.68
- Model time constant, τ_m, = 2.97 seconds
- Model time delay, d_m, = 0.51 seconds

The FOPDT model parameters are inserted into a SIMULINK file and the open loop step response compared with the open loop step response of the PT326 process trainer for validation purposes; the responses are shown in figure (4.20).

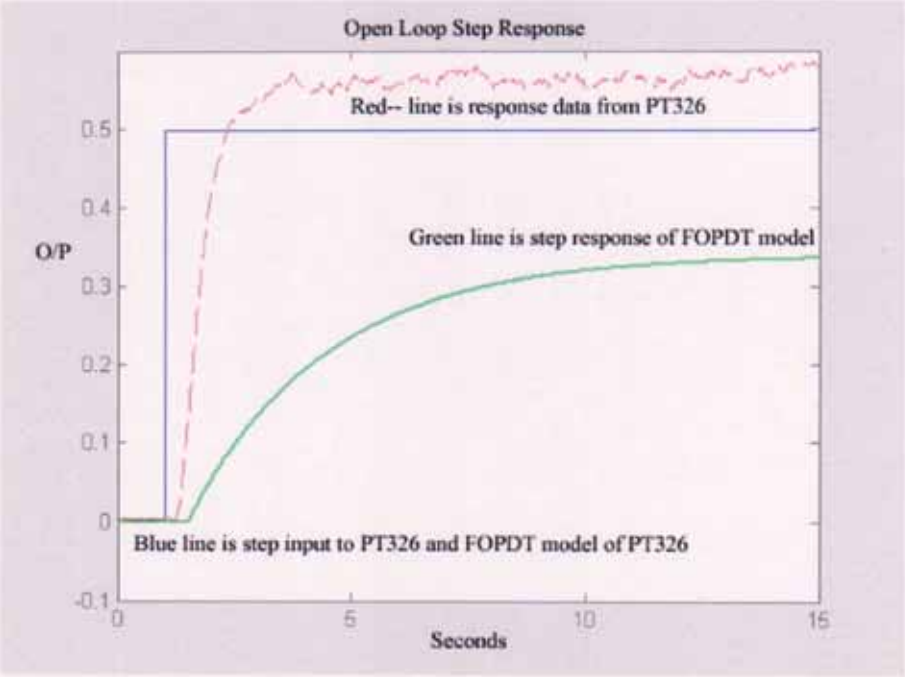

Figure (4.20) Open loop step response of PT326 and FOPDT model (Green line)

Clearly, the process model identified is unsatisfactory.

4.4.2 Relay With Hysteresis

It may be advantageous to use a relay with hysteresis as shown in figure (4.21), so that the resulting system is less sensitive to measurement noise; a further advantage is that the frequency response of $G_p(j\omega)$ at phase angles other than –180° can be obtained.

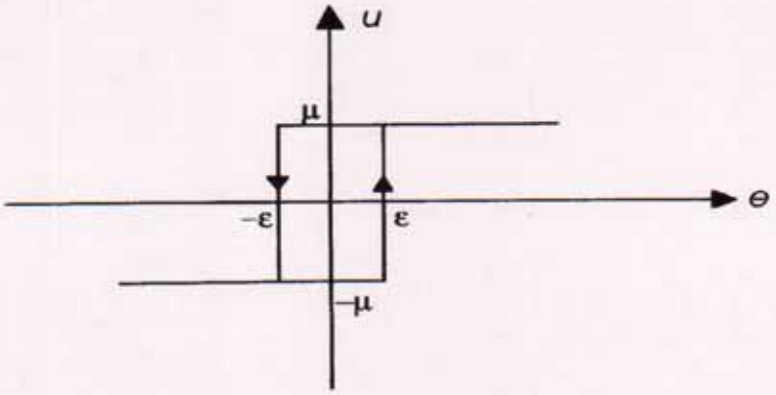

Figure (4.21) Relay with hysteresis (Wang *et al.*, 1999)

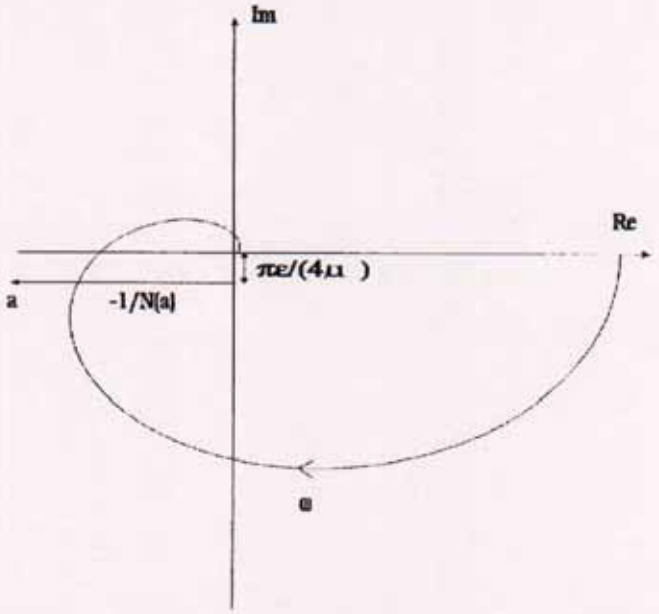

Figure (4.22) Negative inverse describing function of the hysteretic relay (Wang *et al.*, 1999)

The inverse negative describing function of this relay is given by equation (4.22):

$$-\frac{1}{N(a)} = -\frac{\pi}{4\mu}\left(\sqrt{a^2-\varepsilon^2} + j\varepsilon\right) \tag{4.22}$$

In this case, the oscillation corresponds to the point where the negative inverse describing function of the relay crosses the Nyquist curve of the process as shown in

figure (4.22). With hysteresis, there is an additional parameter ε, which can, however, be set automatically based on a pre-determination of the measurement noise level. The width of the hysteresis should be bigger than the noise band (Yu (1999)).

The describing function analysis shows that the introduction of a relay with hysteresis simply helps us to find the frequency response of $G_p(j\omega)$ with phase angles between –90° and –180°. It then becomes obvious that we are able to find the frequency response of $G_p(j\omega)$ between these phase angles by changing ε. In other words, instead of adjusting the frequency to find $G_p(j\omega)$ between phase angles of -90° and -180°, one can find $G_p(j\omega)$ by changing ε.

4.4.2.1 FOPDT Model Identification; Implementation:

A test is carried out to determine the parameters of the FOPDT process model. The process whose parameters are being estimated is the PT326 process trainer. The same file in figure (4.5) is used for identification, with the relay parameters set as shown in figure (4.23). The relay output characteristics are set as 1.3 when "ON" and –0.7 when "OFF". Hysteresis is added to the relay characteristic by setting the hysteresis element, i.e. ε, to 0.1: the relay switch-on point is set to 0.1 and the relay switch-off point is set to –0.1.

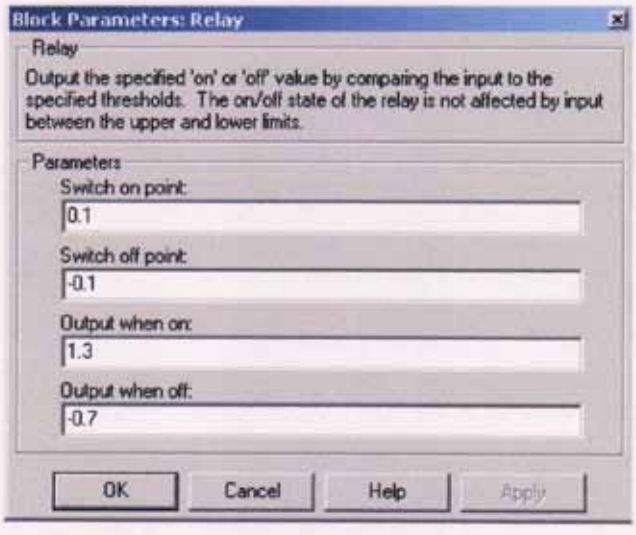

Figure (4.23) Relay with hysteresis settings for test

The program is run for 10 seconds and the resulting relay output and closed loop system output is shown in figure (4.24).

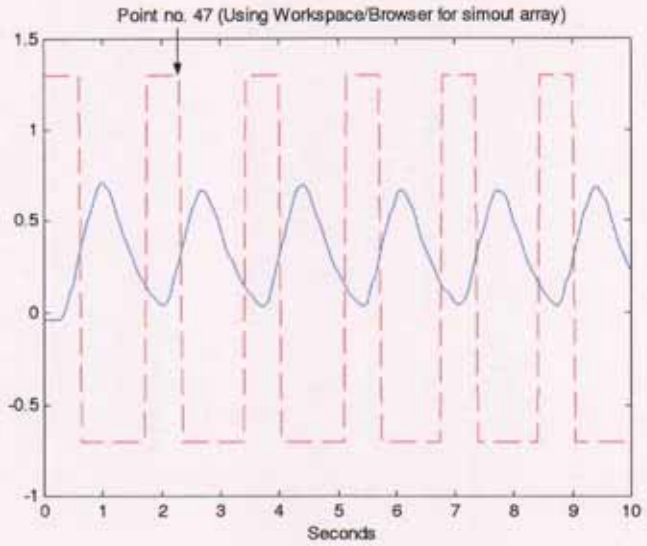

Figure (4.24) Plot of closed loop system output (blue) and relay output (red--)

The MATLAB commands to determine the necessary information from figure (4.24) are given in "Relay_7" in Appendix 2 section 4, page A75. The program shows how the areas are calculated to determine the model gain, K_m, from equation (4.14). The program also calculates the A_u value in figure (4.17) as 0.71. The values obtained are also used to determine the approximate ultimate frequency (equation (4.1)) and the approximate ultimate gain (equation (4.2)) from the data shown in figure (4.24).

The results of the program are:

- Model gain, K_m, = 0.78.
- Ultimate Period, $\hat{P}_u$, = 1.7 seconds
- Ultimate Frequency, $\hat{\omega}_u$, = 3.7 rads/sec
- Ultimate Gain, $\hat{K}_u$, = 5.20

The steady state gain estimate of the process, K_m, using equation (4.14) is 0.78. This appears to be a much smaller value than that calculated in a number of previous experiments carried out on the PT326 process trainer, and so indicates an error. Therefore K_m is determined from a closed loop step response method (Mamat and Fleming, (1995)). To activate this, the "Manual Switch" in figure (4.5) is toggled to the PI controller position and the steady state gain value calculated from the response data according to the technique in Chapter 3, section 3.3.1. The steady state gain, K_m, is determined from equation (3.24) as 1.14.

To determine μ and μ_0, the following simultaneous equations are solved.

$$\mu + \mu_0 = 1.3 \tag{4.23}$$

$$\mu - \mu_0 = -0.7 \tag{4.24}$$

Therefore, $\mu = 0.3$ and $\mu_0 = 1$. A_u is 0.71 and ε is 0.1. The normalised dead time of the process, d_m/τ_m, is computed from equation (4.15) or (4.16) as 0.89. The time constant, τ_m, is found by using equation (4.17) or (4.18) (with T_{u1} equal to 0.55 seconds) as 1.03 seconds. The time delay is determined from equation (4.19) as 0.91 seconds. This results in the parameter estimates of the FOPDT model using the relay with hysteresis, and the closed loop step response method described above to estimate the process model gain, K_m, as follows:

- Model gain, K_m, = 1.14
- Model time constant, τ_m, = 1.03 seconds
- Model time delay, d_m, = 0.91 seconds

Validation: As a test of the accuracy of the estimated parameters using the relay with hysteresis method, an open loop step response and an open loop frequency response plot of the PT326 process trainer and the FOPDT model of the process is recorded.

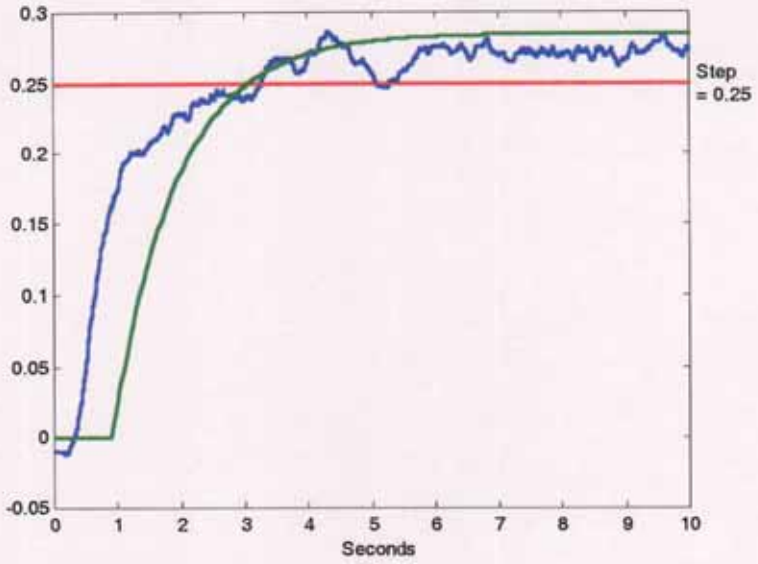

Figure (4.25) Open loop step response of both PT326 (Blue) and FOPDT model of PT326 (Green) using biased relay identification methods

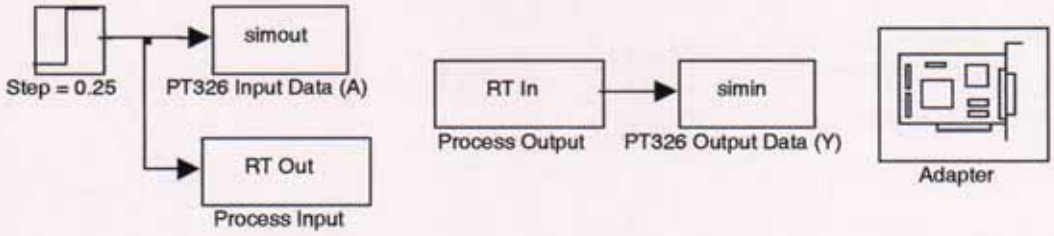

Figure (4.26) MATLAB/SIMULINK/HUMUSOFT file to output step to PT326 process trainer to generate plot in figure (4.25)

Figure (4.27) MATLAB/SIMULINK file to output step to FOPDT model of PT326 process trainer for figure (4.25)

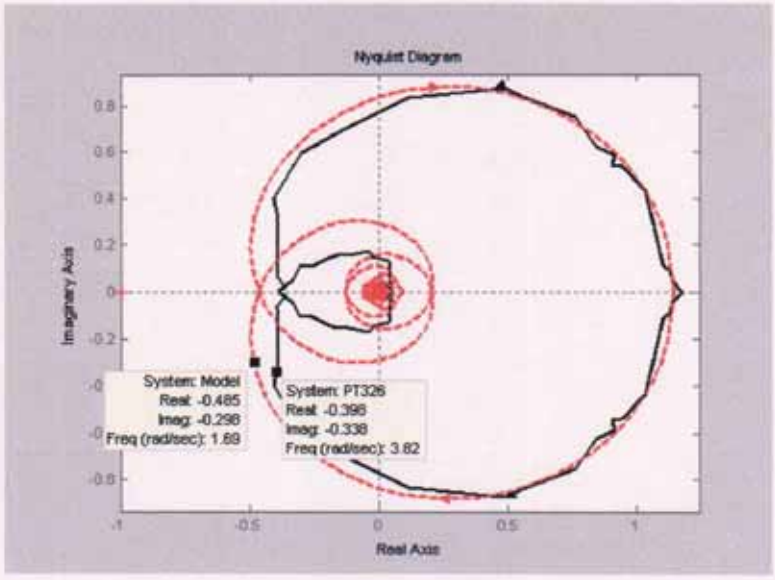

Figure (4.28) Nyquist plot of PT326 (Black) and FOPDT model (Red--)

The labels on the systems in figure (4.28) have no relevance except to identify the process and the process model more easily.

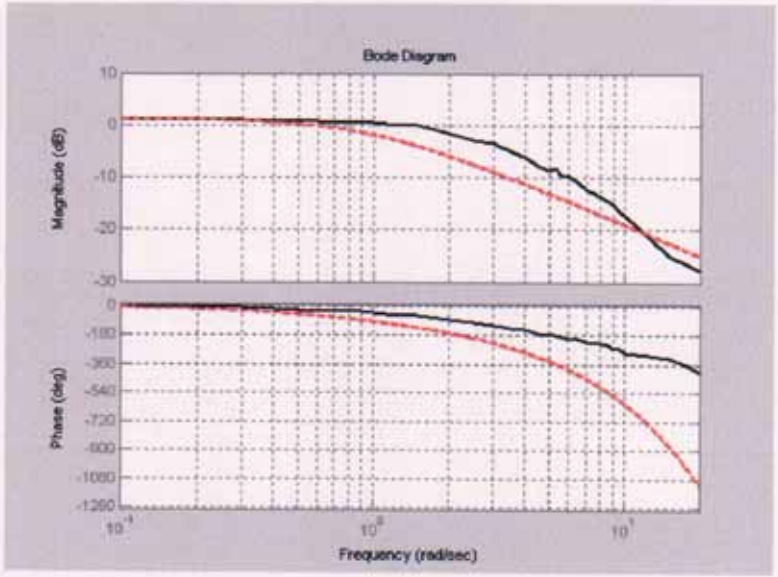

Figure (4.29) Bode plot of PT326 (Black) and FOPDT model (Red--)

The MATLAB commands used to generate figures (4.28) and (4.29) are shown in Appendix 2 section 4, page A75, file name "Relay_8".

The results of the time-domain and the frequency-domain comparisons of the FOPDT model with the actual process in figures (4.25), (4.28) and (4.29), using the relay with hysteresis identification method, show that the models obtained are not as satisfactory as the models obtained from, for example, open loop methods. It is worth mentioning again that the evaluation method for all the identified models is intuitive.

The techniques used in the estimation of model parameter values using the relay based identification methods need to be investigated further. A second-order-plus-dead-time process model is not determined for the PT326 process trainer due to difficulty in implementing the algorithms. Further work in this area is possible.

4.5 Conclusions

The relay based methods of process model identification are the least accurate of the identification techniques examined. The ideal relay method does not always allow real process model parameters to be determined for a simulated or real process. The algorithms are quite difficult to implement. The improved relay feedback methods, whereby bias and/or hysteresis are added to the ideal relay works better than the ideal relay based method, in identifying a process model for the PT326 process trainer.

The relay based algorithms are the most difficult of all the algorithms examined for process model identification. Issues like size of DC bias, and amount of hysteresis introduced are issues to be decided when applying the relay based methods.

The primary information deduced from these relay-based experiments is $\hat{K}_u$ and $\hat{\omega}_u$. This information is very useful in the auto-tuning of PI/PID controllers.

Chapter 5 : Control of Time-Delayed Processes

5.1 Introduction

The PID controller in many cases gives satisfactory performance. It can often be used on processes that are difficult to control provided that optimal performance is not required. There are, however, situations when it is possible to obtain better performance using other types of controllers. Typical examples are processes with long relative time delays and processes with oscillatory responses in open loop. This chapter deals with the control, using PI/PID controllers, of time-delayed processes in simulation and implementation. The proportional-integral-derivative (PID) controller is by far the most commonly used controller in industry (Astrom and Hagglund, 1995). Most feedback loops are controlled by this algorithm or minor variations of it. It is implemented in many different forms. It can be used as a stand-alone controller or as part of a network of distributed systems for process control. Many thousands of instrument and control engineers worldwide are using such controllers in their daily work. The PID algorithm can be approached from many different directions. It can be viewed as a device that can be operated with a few rules of thumb, but it can also be approached analytically (Astrom and Hagglund, 1995). In a PID controller, the control action is generated as a sum of three terms. The control law is thus described as

$$u(t) = u_p(t) + u_i(t) + u_d(t)$$

In the equation shown, u_p is the proportional part, u_i is the integral part and u_d is the derivative part. In proportional control, the output of the controller is proportional to the control error for small errors. The main function of integral action is to make sure that the process output agrees with the set-point in steady state. With proportional control, there is normally a control error in steady state. With integral action, a positive error will always lead to an increasing control signal, and a negative error will give a decreasing control signal. Thus, a controller with integral action will