DEPARTMENT OF INSTRUMENTATION ENGINEERING

B.E. VIII-SEMESTER (E & I)

INSTRUMENTATION AND COMPUTER CONTROL LAB

LIST OF EXPERIMENTS

Cycle-I

1. Design of sampled data control system with Dead-beat controller using TUTSIM
2. Design of Dead-time compensator using smith predictor algorithm and simulation using SIMULINK
3. Process identification using Least Square Estimator algorithm (MATLAB).
4. Design and simulation of Kalman's Controller using TUTSIM
5. Design and realization of digital filter
6. Design of sampled data control system with Dhalin's controller and simulation using TUTSIM.
7. Study of LABVIEW software.

Cycle-II

8. Study of SCADA software
9. Design of Feed forward – Feedback controller and simulation using SIMULINK
10. a) Simulation of a temperature process using LabView.
    b) Data acquisition using LabView.
11. a) Design of inverse response compensator and simulation using SIMULINK
    b) Study of Bio signals
12. Study of PLC (GE Fanuc make)
13. PC based control of Air flow/pressure process using V-MAT data acquisition card.
14. Study of DCS

Expt. No. :

Date :

DESIGN OF SAMPLED DATA CONTROL SYSTEM WITH DEAD-BEAT CONTROLLER USING TUTSIM

AIM

To design a deadbeat controller for a given process and to find the closed loop response of the sampled data system to a unit step change in the set point incorporating deadbeat algorithm.

THEORY

The block diagram of Direct Digital Control Loop (DDC) is as shown in Figure 1. For a set point change the closed loop response is given by

$$\ddot{Y}(z) = \frac{HG_P(z)D(z)}{1 + HG_P(z)}\ddot{Y}_{SP}(z) \qquad (1)$$

Where $HG_P(z)$ =Pulse transfer function of the process with Zero Order Hold.

In DDC for a given step change in the set point the discrete-time response should be specified. Then if HG_P (z), $\ddot{Y}$ (z) and $\ddot{Y}_{sp}$(z) are known we can solve equation (1) with respect to unknown D(z).

The transfer function of the digital controller is given by

$$D(z) = \frac{1}{HG_P(z)} * \frac{\left(\frac{Y(z)}{Y_{SP}(z)}\right)}{1 - \left(\frac{Y(z)}{Y_{SP}(z)}\right)} \qquad (2)$$

DESIGN OF DEADBEAT CONTROLLER FOR A GIVEN PROCESS

The closed loop response for a deadbeat control algorithm to a unit step change in the input exhibits no error at all sampling instants after the first instant. If the response is to have zero error at all sampling instants after the first, its discrete-time behaviour resembles that of a unit step delayed by one sampling instant. Therefore,

$$Y(z) = \frac{z^{-1}}{1 - z^{-1}} \qquad (3)$$

$$Y_{SP}(Z)=\frac{1}{1-z^{-1}}; \qquad \frac{Y(z)}{Y_{sp}(z)}=z^{-1} \tag{4}$$

Substituting equations (3) and (4) in equation (2)

$$\mathrm{D(z)}=\frac{1}{\mathrm{HG_P(z)}}*\frac{\mathrm{z^{-1}}}{1-\mathrm{z^{-1}}} \tag{5}$$

PROBLEM

Design a dead beat controller for the process

$$G_P(s)=\frac{1}{(0.5s+1)(s+1)(s+1)(2s+1)} \text{ with Sampling Time T=5 seconds} \tag{6}$$

DESIGN

First approximate the given process into first order plus dead time model using process reaction curve method. The block diagram schematic is as shown in Figure 2 and the open loop response obtained is as shown in Figure 3.

The approximated first order time delay model is obtained from Figure 3 and is given by

$$G_P(s)=\frac{1e^{-1.889s}}{(2.943s+1)} \tag{7}$$

The pulse transfer function is given by

$$HG_P(z)=Z[H(s)\,G_P(s)]$$

where $H(s)$ = zero order hold transfer function and is given by

$$H(s)=\left(\frac{1-e^{-sT}}{s}\right)$$

$$HG_P(z)=Z\left[\left(\frac{1-e^{-sT}}{s}\right)\left(\frac{1*e^{-1.889s}}{(2.943s+1)}\right)\right]$$

$$=(1-Z^{-1})*Z\left[\frac{e^{-1.889s}}{S*(2.943S+1)}\right]$$

Sampling time T = 5 seconds

$$\lambda T = t_d$$
$$\lambda T = 1.889$$
$$\lambda = \frac{1.889}{5} = 0.3778$$
$$m = 1 - \lambda$$
$$m = 1 - 0.3778 = 0.6222$$

$$\therefore HG_P(z) = \left(1 - z^{-1}\right) * Zm\left[\frac{1}{s(2.943s+1)}\right]_{m=0.6222}$$

$$= \left(1 - z^{-1}\right) * Zm\left[\frac{1/2.943}{s\left(s + 1/2.943\right)}\right]_{m=0.6222}$$

$$= \left(1 - z^{-1}\right) * Zm\left[\frac{0.3398}{s(s + 0.3398)}\right]_{m=0.6222,\, a=0.3398}$$

$$HG_P(z) = \left(1 - z^{-1}\right) * Zm\left[\frac{0.3398}{s(s + 0.3398)}\right]_{m=0.6222,\, a=0.3398}$$

We Know

$$Zm\left[\frac{a}{S(S+a)}\right] = \left[\frac{1}{z-1} - \frac{e^{-amT}}{z - e^{-aT}}\right]$$

$$HG_P(z) = \left(1 - z^{-1}\right) * Zm\left[\frac{0.3398}{s(s + 0.3398)}\right]_{m=0.6222,\, a=0.3398}$$

$$= \left(1 - z^{-1}\right)\left[\frac{1}{z-1} - \frac{e^{-(0.3398)\ (0.6222)\ (5)}}{z - e^{-(0.3398)\ (5)}}\right]$$

$$= \left(1 - z^{-1}\right)\left[\frac{1}{z-1} - \frac{0.3475}{z - 0.1829}\right]$$

$$= \frac{z-1}{z}\left[\frac{1}{z-1} - \frac{0.3475}{z - 0.1829}\right]$$

$$= \frac{1}{z} - \frac{(0.3475z - 0.3475)}{(z^2 - 0.1829z)}$$

$$= \frac{z^2 - 0.1829z - 0.3475z^2 + 0.3475z}{z^3 - 0.1829z^2}$$

$$= \frac{0.6525z^2 + 0.1646z}{z^3 - 0.1829z^2}$$

$$= \frac{z(0.6525z + 0.1646)}{z(z^2 - 0.1829z)}$$

$$HG_P(z) = \frac{0.6525z + 0.1646}{z^2 - 0.1829z}$$

$$= \frac{z\left(0.6525 + 0.1646z^{-1}\right)}{z^{2}\left(1 - 0.1829z^{-1}\right)}$$

$$= \frac{0.6525 + 0.1646z^{-1}}{z - 0.1829}$$

$$= \frac{z^{-1}\left(0.6525 + 0.1646z^{-1}\right)}{\left(1 - 0.1829z^{-1}\right)}$$

$$D(z) = \frac{1}{HG_P(z)} * \frac{z^{-1}}{1 - z^{-1}}$$

$$D(z) = \frac{\left(1 - 0.1829z^{-1}\right)}{z^{-1}\left(0.6525 + 0.1646z^{-1}\right)} * \frac{z^{-1}}{1 - z^{-1}}$$

$$D(z) = \frac{\left(1 - 0.1829z^{-1}\right)}{\left(0.6525 - 0.6525z^{-1} + 0.1646z^{-1} - 0.1646z^{-2}\right)}$$

$$\frac{m(z)}{e(z)} = \frac{\left(1 - 0.1829z^{-1}\right)}{\left(0.6525 - 0.4879z^{-1} - 0.1646z^{-2}\right)}$$

$$0.6525m(z) - 0.4879Z^{-1}m(z) - 0.1646Z^{-2}m(z) = e(z) - 0.1829e(z)z^{-1}$$

Taking inverse Z-transform

$$m_n = 0.7477m_{n-1} + 0.2523m_{n-2} + 0.15326e_n - 0.2803e_{n-1}$$

EXPERIMENTAL PROCEDURE

(i) Give the step input of magnitude 'u' for the given higher order process as shown in Figure 2.

(ii) Obtain the process reaction (S- shaped) curve as shown in Figure 3.

(iii) From the process reaction curve find the steady state value(B), t_1 and t_2 where t_1=time corresponds to 28.3% of B and t_2= time corresponds to 63.2% of B.

(iv) Approximate the first order plus dead time model using the given formulae as $G_P(s) = \frac{K_P e^{-tds}}{\tau s + 1}$ where K_P is the process gain, τ is the process time constant and t_d is the process dead time.

$$K = \frac{\text{Change in steady state value}}{\text{change in the input}} = \frac{B}{U}$$

$$\tau = 1.5 * \left(t_2 - t_1\right) \qquad (8)$$

$$t_d = t_2 - \tau$$

(v) Derive the pulse transfer function $HG_P(z)$.

(vi) Obtain the digital controller D(z) as given in the design.

(vii) Implement the transfer function of the dead beat controller by using Z-blocks in TUTSIM for actual and approximated process as shown in Figure 4 and Figure 5 respectively.

(viii) Analyze the response and comment on your results.

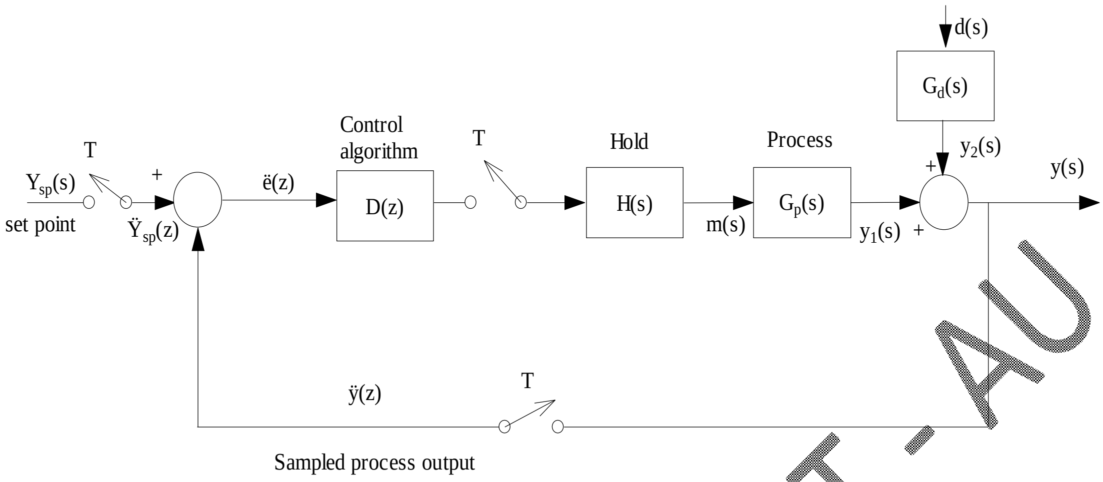


Figure1. Block Diagram of Direct Digital Control Loop

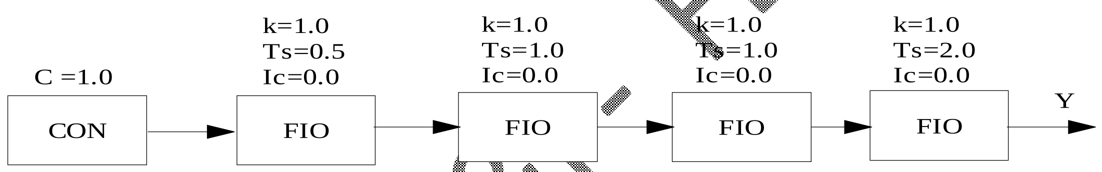


Figure2. Block Diagram to obtain Open Loop Step Response

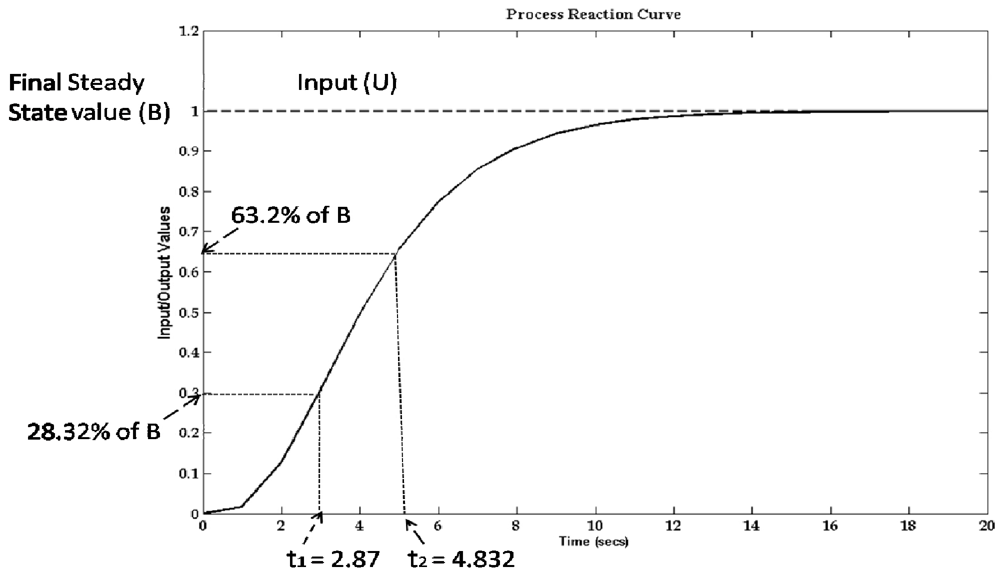


Figure3. Open Loop Step Response of the higher order process for the step input of magnitude "u"

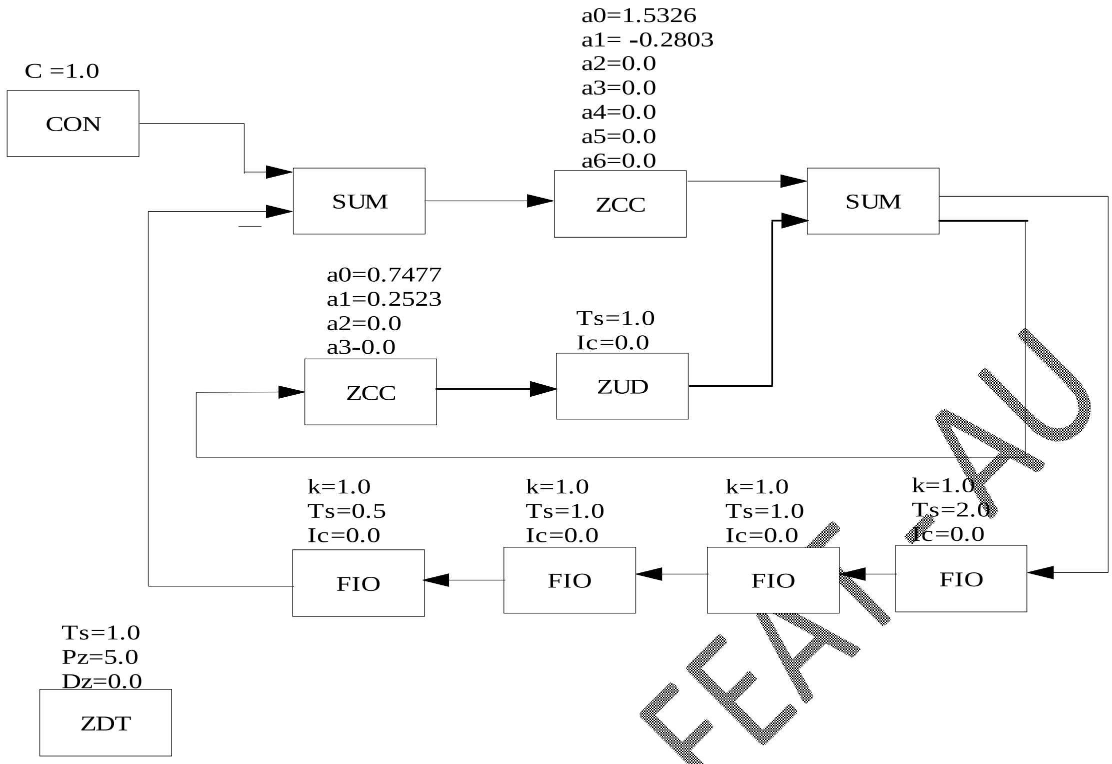


Figure4. TUTSIM Block Diagram of Deadbeat Controller for actual process

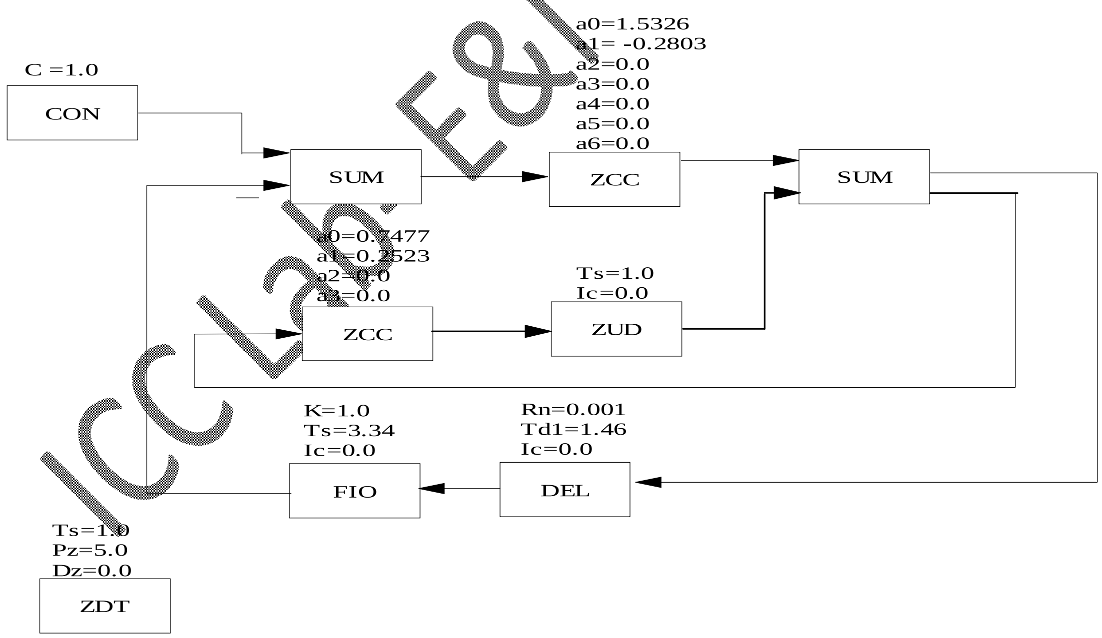


Figure5. TUTSIM Block Diagram of Deadbeat Controller for approximated process

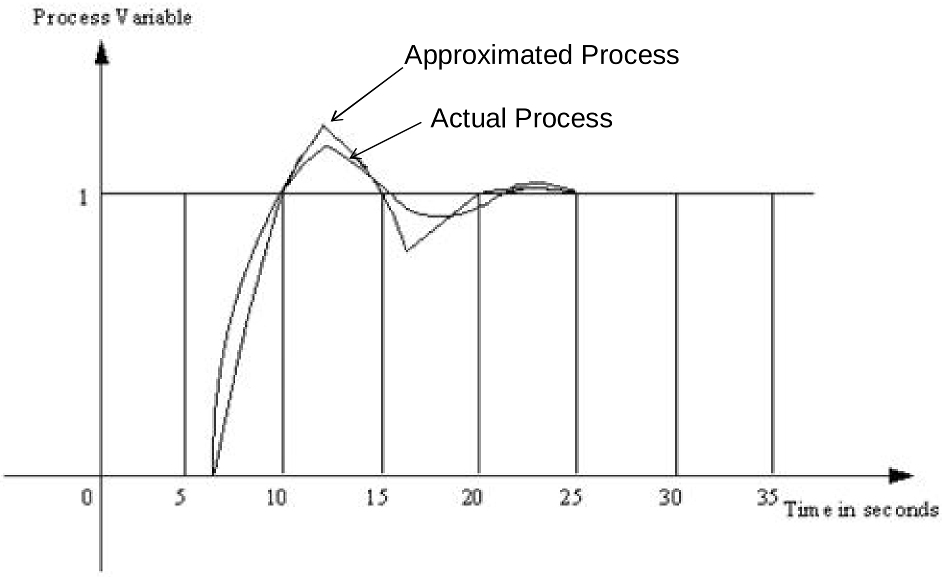


Figure6. Response of Deadbeat Controller for both given process and approximated process

RESULT

The closed loop response of a sampled data system to a unit step change in the set point has been obtained after designing a dead beat controller.

Expt. No. :

Date :

DESIGN OF DEAD-TIME COMPENSATOR USING SMITH PREDICTOR ALGORITHM AND SIMULATION USING SIMULINK

AIM

1. To design a dead-time compensator using Smith Predictor algorithm.
2. To obtain the closed loop response of the given system (with large dead time) with and without compensator.

THEORY

A conventional feedback controller would provide unsatisfactory closed loop response for a system having large dead-time, for the following reasons:

1. A disturbance entering the process will not be detected until after a significant period of time.
2. The control action that will be taken on the basis of the last measurement will be inadequate because it attempts to regulate a situation that originated before.
3. The control action will also take some time to make its effect felt by the process.

As a result of all the factors noted above, considerable dead time is a significant source of instability. Consider a simple feed back loop shown in Figure. 1(a) with the process transfer function,

$$G_p(s) = G(s)e^{-tds}$$

The open loop response to a change in set point for the process is given by

$$Y(s) = G_c(s)\ \{G(s)\ e^{-tds}\}\ Y_{sp}(s)$$

Here the response is delayed by t_d minutes.

In order to eliminate the undesired effects caused by dead time, the open loop feedback signal should carry the current information and not the delayed information such as

$$Y^*(s) = G_cG(s)\ Y_{sp}(s)$$

This is made possible, if $Y'(s)$ is added with $Y(s)$ as shown in Figure 1(b) and $Y'(s)$ is given by,

$$Y'(s) = \{(1-e^{-tds})\ G(s)\}\ G_cY_{sp}(s).$$

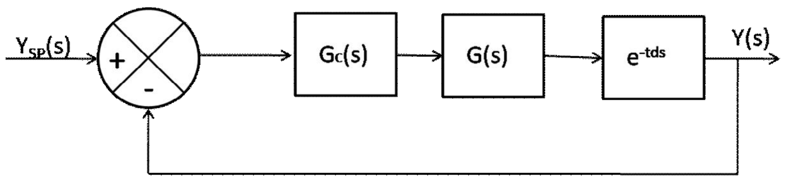


Figure 1(a). Block Diagram of Closed Loop System

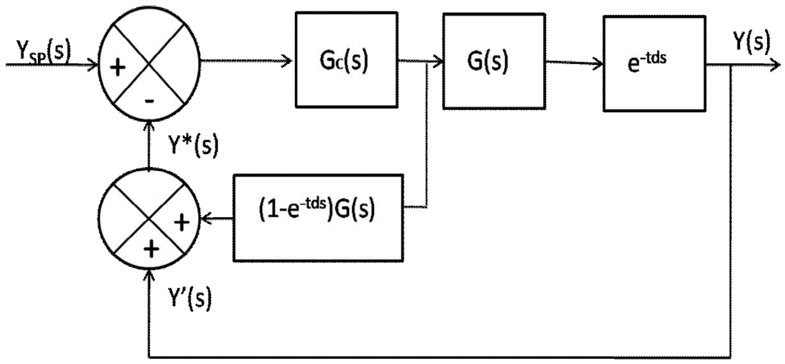


Figure 1(b). Block Diagram of System with Dead time and Compensation Scheme

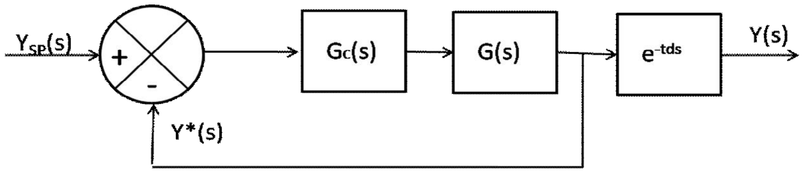


Figure 1(c). Block Diagram of System with Dead time Compensation

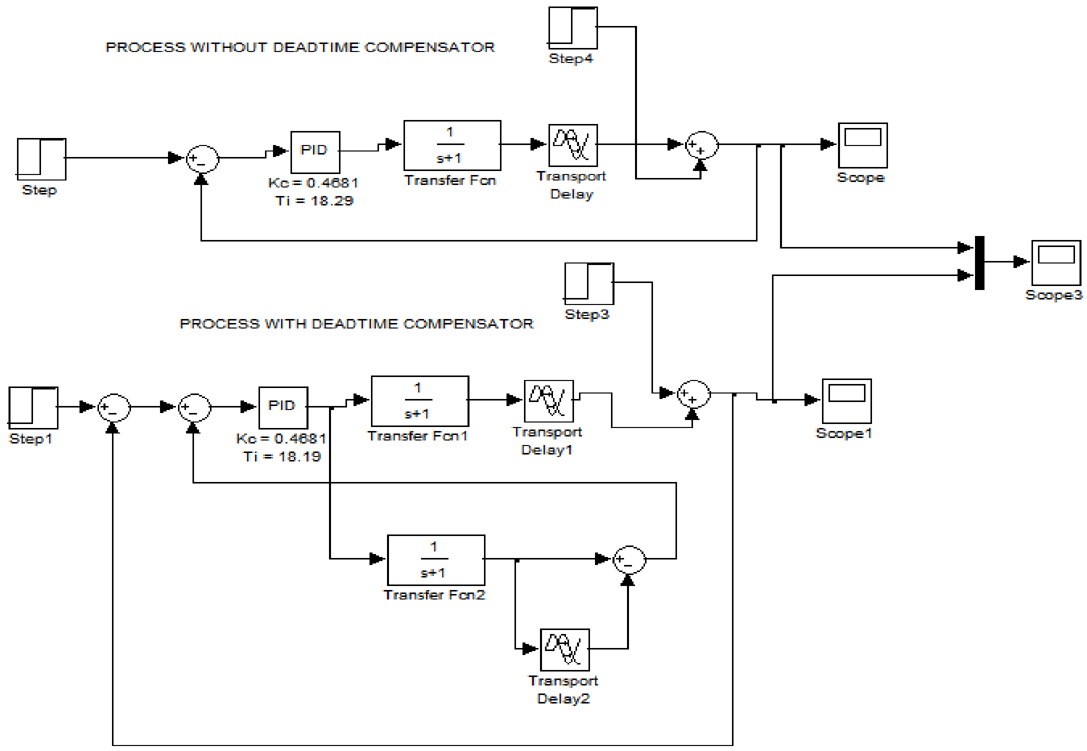


Figure 2. SIMULINK Diagram of System without and with Dead time Compensation

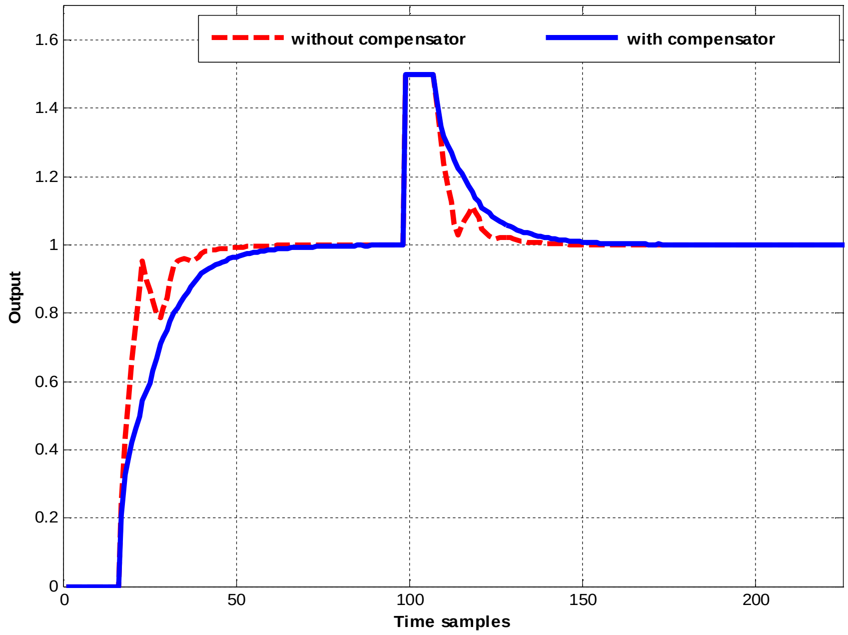


Figure 3. Servo-Regulatory Response of the process with and with out dead time compensator

Now,

$$Y^*(s) = Y(s) + Y'(s)$$

$$= [(1-e^{-tds})\, G(s)\}\, G_cY_{sp}(s)] + G(s)G_c(s)\, e^{-tds}$$

$$= G(s)G_c(s)Ysp(s) - e^{-tds}G(s)G_c(s)Ysp(s) + e^{-tds}\, G(s)G_c(s)Ysp(s)$$

Therefore, $Y^*(s) = G_cG(s)\, Y_{sp}(s)$. Here the feedback signal contains only the current information as shown in Figure 1(c).

The dead time compensator predicts the delayed effect that the manipulated variable will have on the process output. This prediction led to the term Smith predictor. Perfect compensation is possible only if the process model is perfectly known.

Dead time compensation is needed only when the ratio of dead time to time constant is greater than one. The controller should be conventional PID or PI controller.

EXAMPLE

Demonstrate the effect of dead time compensation for the process given by,

$$G(s) = \{ e^{-10s}/(s+1)\}$$

Compare the response with and without dead time compensator.

PROCEDURE

1. For the given process $G(s) = \{ e^{-10s}/(s+1)\}$ check the closed loop response by tuning PI controller values [$k_c = 0.4681$; $T_i = 18.29$]
2. Analyze both the servo and regulatory responses with and without dead time compensator and comment on your result.
3. Obtain the closed loop response of the given process without dead time compensator and analyze the robustness i.e by applying deviation in model parameters like gain, time constant and deadtime to show the effect of compensator under model mismatch condition. (Refer Figure 4 and Figure 5).

RESULT

A dead time compensator using smith predictor algorithm was designed and the process was simulated using Simulink and the closed loop response incorporating dead time compensator and without dead time compensator was compared.

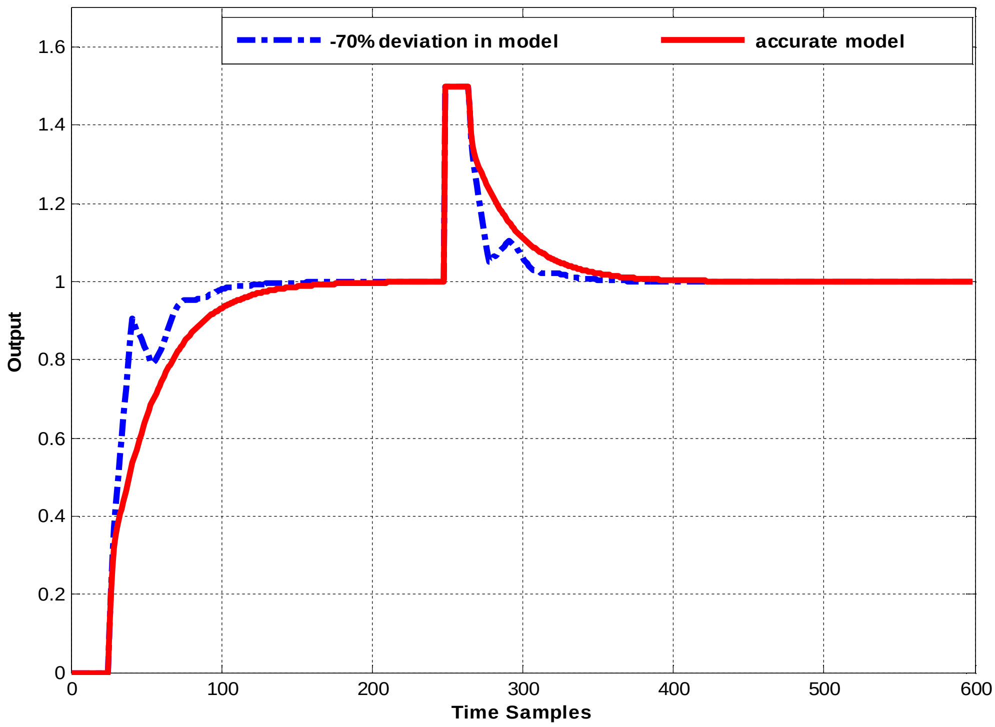


Figure 4. Servo-Regulatory Response of the process with dead time compensator and -70% model mismatch

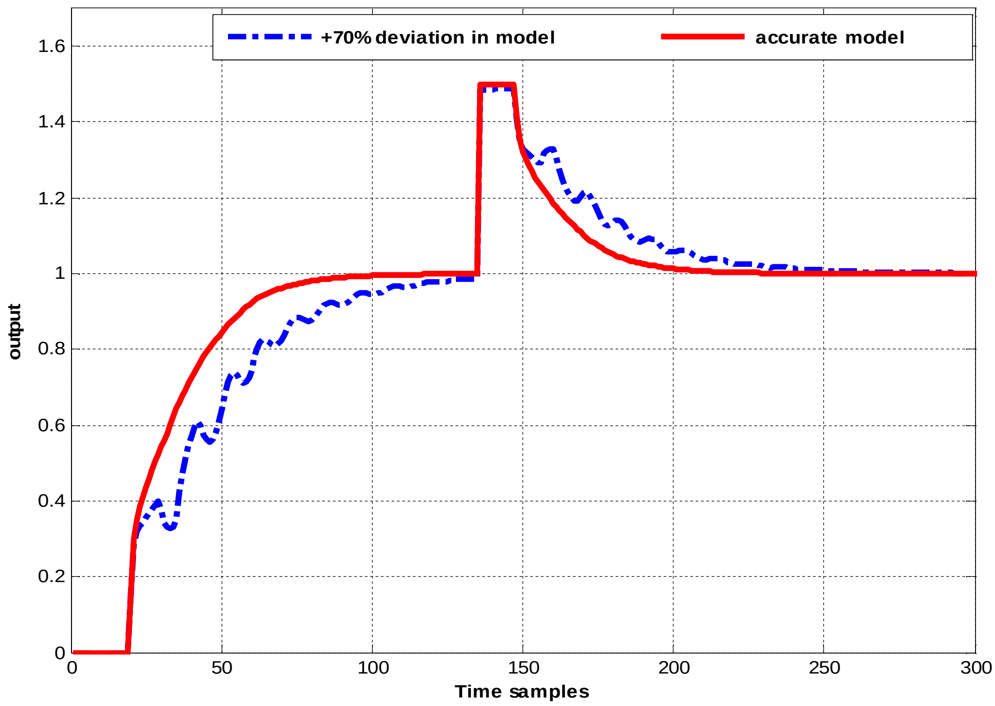


Figure 5. Servo-Regulatory Response of the process with dead time compensator and +70% model mismatch

Expt. No. :

Date :

PROCESS IDENTIFICATION USING LEAST SQUARE ESTIMATION ALGORITHM (MATLAB)

AIM

To identify the parameters of a linear discrete model for a given process using least square estimation algorithm.

INTRODUCTION

System identification is the art and exercise of identifying the various process variable or the underlying physical phenomena in a process (more often from a process data). The resulting expression is known as model.

Thus the model is a set of differential (or difference) and algebraic equations that can describe the process behavior.

The need for process model arises in many control applications for the use of timing methods. Process model are also needed in developing feed forward control algorithm, self-tuning and internal-model algorithm.

Process identification provides several forms that are useful in process control are

1. Process reaction curve (obtained by step input)
2. Frequency response diagram (obtained by sinusoidal input)
3. Pulse response (obtained by pulse input)

The n^{th} order linear discrete model is given by

$$\bar{y}_n = b_1 u_{n-1} + b_2 u_{n-2} \ldots\ldots + b_{nb} u_{n-nb} - a_1 y_{n-1} \; a \; a_2 y_{n-2} - \ldots\ldots - a_{na} y_{n-na} \qquad (1)$$

Where $\bar{y}_n \rightarrow$ predicted value of the current output of the process.

$b_1, b_2, \ldots b_{nb}, a_1, a_2, a_{na} \rightarrow$ unknown parameters

$y_{n-1}, y_{n-2}, \ldots y_{n-na} \rightarrow$ previous values of output

$u_n, u_{n-1}, u_{n-2}, \ldots u_{n-nb} \rightarrow$ present and past values of input

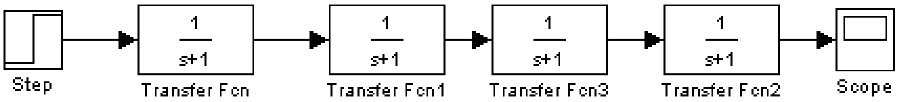


Figure1.Block diagram to get open loop response

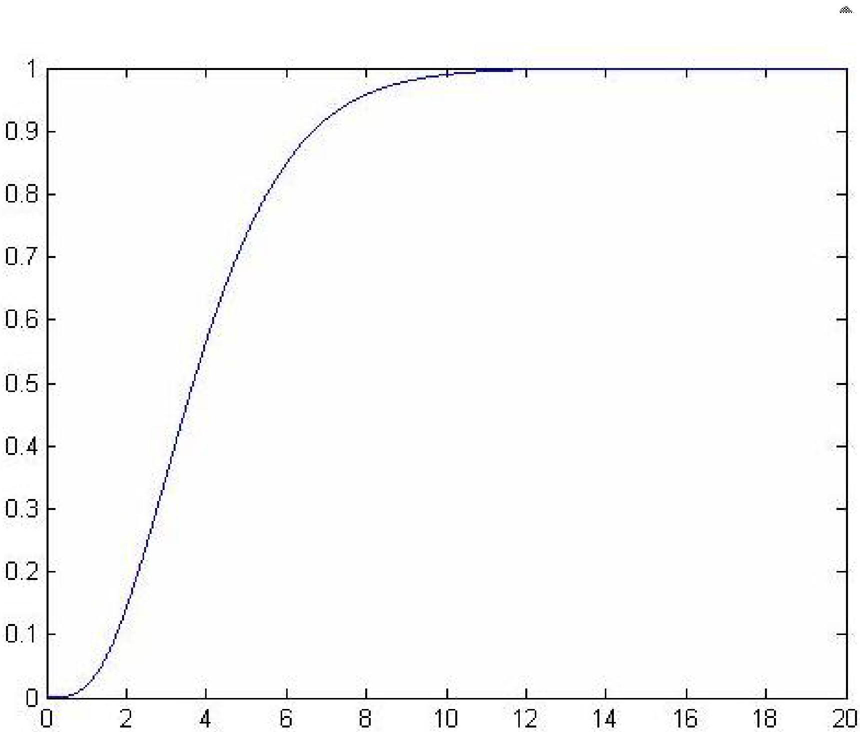


Figure 2 process response for $\frac{1}{(s+1)^4}$

From the response,

t_1 (23.6% of the final value)=2.71secs

t_2 (63.2% of the final value)=4.857secs

$\tau=1.5*(t_2-t_1)=1.5*(4.857-2.71)=3.8$secs

$t_d=t_2-\tau=4.85-3.8=1.28$secs

The approximated model is $\frac{1.04}{3.8s+1}e^{-1.28S}$

Equ. (1) is usually written as

$$\overline{Y}(z) = \frac{B(z\,\text{-}1)}{A\,(z\,\text{-}1)} z^{-nk}\ u(z) \tag{2}$$

Now we define the parameter vector containing unknown parameters.

$$\theta = [b_1, b_2, \ldots b_{na}, a_1, a_2, \ldots a_{na}]^T \tag{3}$$

There are (n_b+n_a) unknown parameters.

Let us define NC as this total, so the vector has NC rows.

Let N=na, M=nb.

Now suppose we have NP data points that give values of the output yn for known values of y_{n-1}, y_{n-2}, $\ldots y_{n-N}$, u_n, u_{n-1}, $\ldots$ u_{n-M}.

The data could be grouped as follows.

	y_n values	y_{n-1} Values L.	y_{n-N} values	u_1 Values	u_{n-1} Values	L u_{n-M} Values
1	$y_{1,n}$	$y_{1,n-1}$	$y_{1,n-N}$	$u_{1,n}$	L	$u_{1,n-M}$
2	$y_{2,n}$	$y_{2,n-1}$	$y_{2,n-1}$	$u_{z,n-1}$	L	$u_{2,n-M}$
M	M	M	M	M		M
NP	$y_{NP,n}$	$y_{NP,n-1}$	$y_{NP,n-N}$	$u_{NP,n-1}$	L	$u_{NP,n-M}$

Our objective is to minimize the sum of the squares of the differences between actual measured data points $(y_{i,\,r})$ and those predicted by our model equation $(\overline{y}_{i,n})$.

$$J = \sum_{i=1}^{NP} (y_{i,n} - \overline{y}_{i,n})^2 \tag{4}$$

This is a least square problem that is solved by taking derivatives of J with respect to each of the unknown parameters (the NC elements of the θ vector) and setting these partial derivatives equal to zero. This gives NC equations i.e., NC unknowns. The solution is compactly written in matrix form.

$$\theta = [A^TA]^{-1}A^TY \tag{5}$$

where matrix A (with NP rows and NC columns) and the Y vector (with NP rows) are defined to represent the data points.

$$A \begin{bmatrix} y_{1,n-1} & y_{1,n-2} & \cdots & y_{1,n\text{-}N} & u_{1,n} & \cdots & u_{1,n-M} \\ y_{2,n-1} & y_{2,n-2} & \cdots & y_{2,n\text{-}N} & u_{2,n} & \cdots & u_{2,n-M} \\ \vdots & \vdots & & \vdots & \vdots & & \\ y_{NP,n\text{-}1} & y_{NPP,n\text{-}2} & \cdots & y_{NP,n-N} & u_{NP,n} & \cdots & u_{NP,n-M} \end{bmatrix} \tag{6}$$

$$Y = \begin{bmatrix} y_{1,n} & y_{2,n} & y_{3,n} & \cdots & y_{NP,n} \end{bmatrix}^T \tag{7}$$

In this experiment the actual process is

$$G_P(s) = \frac{1}{(s+1)^4} \tag{8}$$

The first order process with dead time can exactly model the above process as

$$G_P(s) = \frac{1.04e^{-1.28s}}{3.8s+1}$$

Let the first order model can be written in the form,

$$\frac{y(s)}{u(s)} = G_P(S) = \frac{K_P e^{-t_d s}}{\tau s + 1}$$

Let $\quad G_o(s) = \dfrac{1 - e^{-sT}}{s}$

Where T $\rightarrow$ sampling period

The pulse transfer function of this process with zero order hold is

$$Z\,[G_o(s)\,G_o(s)] = Z\left[\frac{1-e^{-sT}}{s} \times \frac{kpe^{-tds}}{\tau s+1}\right]$$

Let td = nk* T

$$nk = \frac{td}{T}$$

$$Z\left[G_o(s)G_p(s)\right] = Z\left[\frac{1-e^{-ST}}{s} \times \frac{kpe^{-(nkTs)}}{\tau s+1}\right]$$

$$= z^{-nk}\,(1-z^{-1})\; Z\left[\frac{kp}{s(\tau s+1)}\right]$$

$$= z^{-nk}\,(1-z^{-1})\; Z\left[\frac{kp/\tau}{s(s+1/\tau)}\right]$$

$$\frac{kp/\tau}{s(s+1/\tau)} = \frac{A}{s} + \frac{B}{(s+1/\tau)}$$

$$A = \frac{kp/\tau}{s(s+1/\tau)} * s|_{s \to 0} = \frac{kp/\tau}{1/\tau} = kp$$

$$B = \frac{kp/\tau}{s\,(s+1/\tau)} * (s+1/\tau)\,|_{s \to -1/\tau}$$

$$= \frac{kp/\tau}{-1/\tau}$$

$$B = -\ Kp$$

$$Z[G_o(s)\,G_P(s)] = z^{-nk}\,(1-z^{-1})\,Z\left[\frac{kp}{s} - \frac{kp}{(s+1/\tau)}\right]$$

$$= (1-z^{-1})\,z^{-nk}\,kp\;Z\left[\frac{1}{s} - \frac{1}{s+1/\tau}\right]$$

$$= (1-z^{-1})\,z^{-nk}\,kp\left[\frac{z}{z-1} - \frac{z}{z-e^{-T/\tau}}\right]$$

$$= \frac{z-1}{z} = z^{-nk}\,kp\left[\frac{z^2 - z\,e^{-T/\tau} - z^2 + z}{(z-1)\,(z-e^{-T/\tau})}\right]$$

$$= z^{-nk}\,kp\left[\frac{1-e^{-T/\tau}}{z-e^{-T/\tau}}\right]$$

Let $b = e^{-T/\tau}$

$\log b = -\ T/\tau \quad \tau = -\dfrac{T}{\log b}$

$$Z\,[G_o(s)\;G_P(s)] = z^{-nk}\,kp\left[\frac{1-b}{z-b}\right]$$

$$Z\,[G_o(s)\,G_P(s)] = z^{-nk}\,\frac{kp\,(1-b)}{z-b}$$

$$HGp(z) = \frac{y(z)}{u(z)} = \frac{kp(1-b)}{z^{nk}\,(z-b)}$$

Multiply numerator and denominator by z^{-1}

$$\frac{y(z)}{u(z)} = \frac{kp(1-b)\,z^{-1}}{z^{nk}\,(z-b)\,z^{-1}}$$

$$HGp(z) = \frac{kp(1-b)\,z^{-1}}{z^{nk}\,(1-b\,z^{-1})}$$

Let $b_1 = Kp*(1-b)$

$a_1 = -b$

$$HGp(z) = \frac{z^{-nk}\,b_1\,z^{-1}}{1+a_1\,z^{-1}}$$

Assume $b_0 = 0$, we get the final value as,

$$\frac{y(z)}{u(z)} = HGp(z) = z^{-nk}\,\frac{(b_0 + b_1\,z^{-1})}{1+a_1\,z^{-1}}$$

The discrete transfer function has three parameters that need to be identified; nk, b_1 and a_1.

PROCEDURE

1. Obtain the open loop response of actual process Gp (S)(fig.1,fig.2).
2. From the open loop response, process parameters are estimated using Two point method.
3. Load the data files from the simulink environment to matlab editor and then execute the program to estimate the values of a, b, nk.
4. Tabulate Kp, τ, t_d and $J_{min}$for various valuesof nk.
5. Select the Kp, τ and t_d from table with the low J_{min} criterion.
6. Compare the identified model with the actual process using (fig.3) simulink. Check the accuracy of the estimated model. .

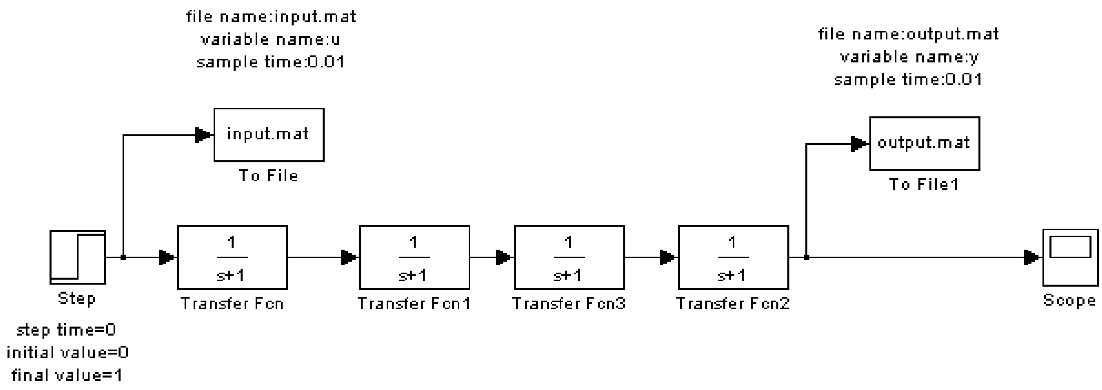


Figure 3.Simulink diagram for open loop response

MATLAB PROGRAM

```
loadoutput.mat;
loadinput.mat;
t=0:0.01:20;
y1=y(2,:);
plot(t,y1)
z=[y(2,:)' u(2,:)'];
na=1;nb=1;nk=125;
para=arx(z,[na,nb,nk],[],0.01);
[a,b]=th2tf(para);
k1=b(1,2);
k1=(-1.0)*k1;
k2=1-k1;
k3=a(1,126);
kp=k3/k2
ts=para(1,2)
deadtime=para(1,9)*para(1,2)
tau=(ts/log(k1))*(-1.0)
jmin=para(1,1)
```

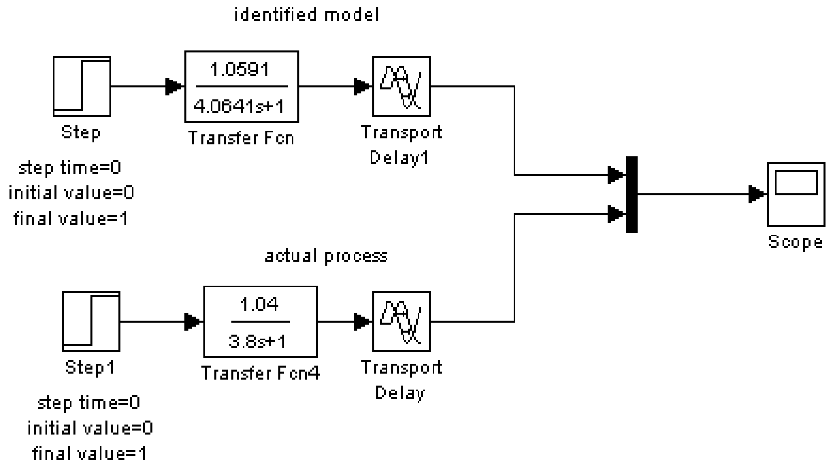


Figure 4 .Simulink diagram to verify the identified model with the original process

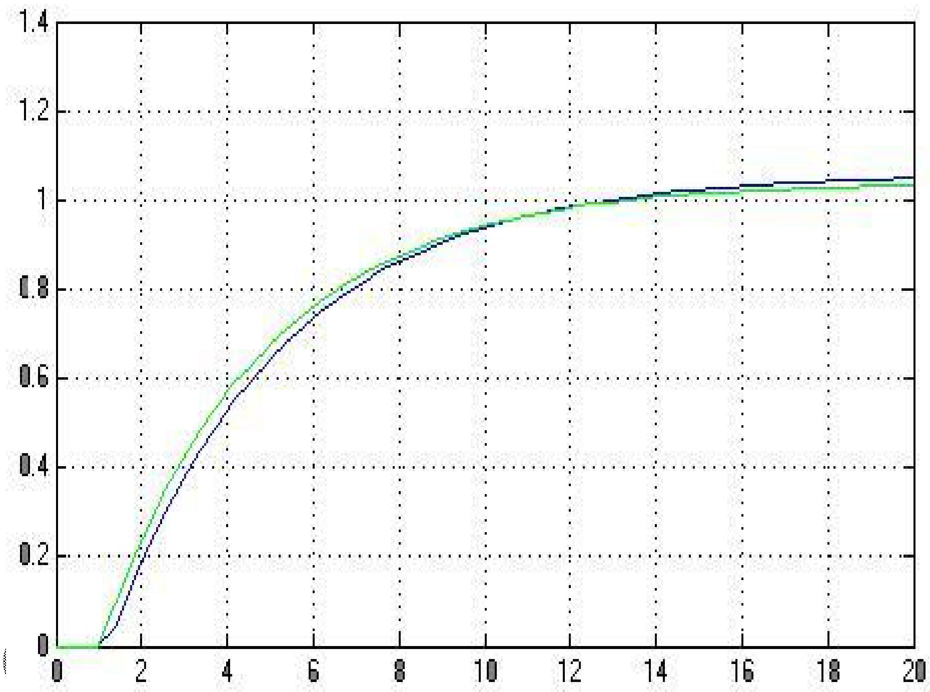


Figure 5 Estimated and Approximated model output

TABULATION:

nk	Kp	t_d	τ	J_{min}

RESULT

Thus the parameter of a linear discrete model for a given process has been identified using least square estimation approach.

The identified model is $\frac{1.0591}{4.0641s+1} e^{-1.2S}$

The actual process is $\frac{1.04}{3.8\,s+1} e^{-1.28.S}$

Expt. No. :

Date :

DESIGN AND SIMULATION OF KALMAN'S CONTROLLER USING TUTSIM

AIM

To design a Kalman's controller for a given process and to find the closed loop response of the sampled data system to a unit step change in the set point incorporating Kalman's algorithm.

THEORY OF KALMAN CONTROLLER

To synthesis a digital control algorithm by Kalman's approach, we place restrictions on C and M, instead of usual C/R. Thus the derivation of the Kalman's algorithm assumes a specific C (z). That is the system response to a unit step input reaches the final value in two sampling time instants and remains at the final value thereafter as shown in Figure 1. The expression for C (z) is

$$C(z) = c_1 z^{-1} + z^{-2} + z^{-3} + \ldots \tag{1}$$

In order to accomplish this manipulated variable will assume two intermediate values and then assumes its final value thereafter as shown in Figure 1.

$$M(z) = m_0 + m_1 z^{-1} + m_f z^{-2} + m_f z^{-3} + \ldots \tag{2}$$

where no restrictions need be placed on the value of c_1 and m_f equals the reciprocal of the process steady state gain. The number of intermediate values of M equals the order of the process.

If the control algorithm is to be used for a unit step change in set point, then

$$R(z) = \frac{1}{1 - z^{-1}}$$

$$\begin{aligned}\frac{C(z)}{R(z)} &= (1 - z^{-1})(c_1 z^{-1} + z^{-2} + z^{-3} + \ldots) \\ &= C_1 z^{-1} + (1 - C_1) z^{-2} \\ &= P_1 z^{-1} + P_2 z^{-2} \\ &= P(z)\end{aligned} \tag{3}$$

and $\dfrac{M(z)}{R(z)} = (1 - z^{-1})(m_0 + m_1 z^{-1} + m_f z^{-2} + m_f z^{-3} + \ldots.)$

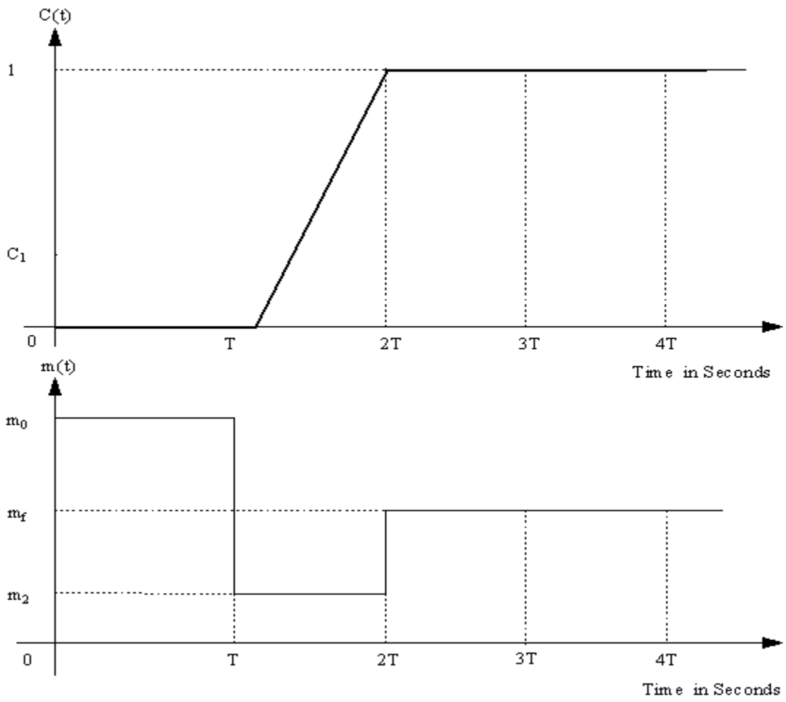


Figure 1. Desired response characteristics used to design Kalman's algorithm

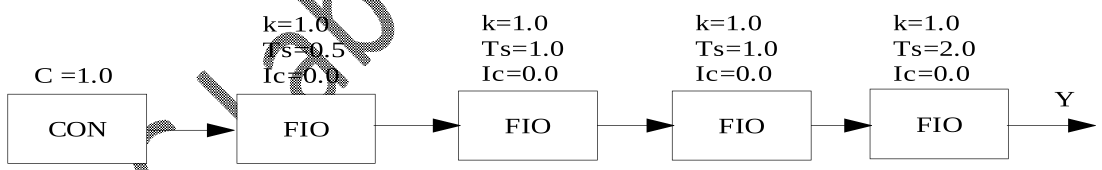


Figure 2. Block Diagram to obtain Open Loop Response

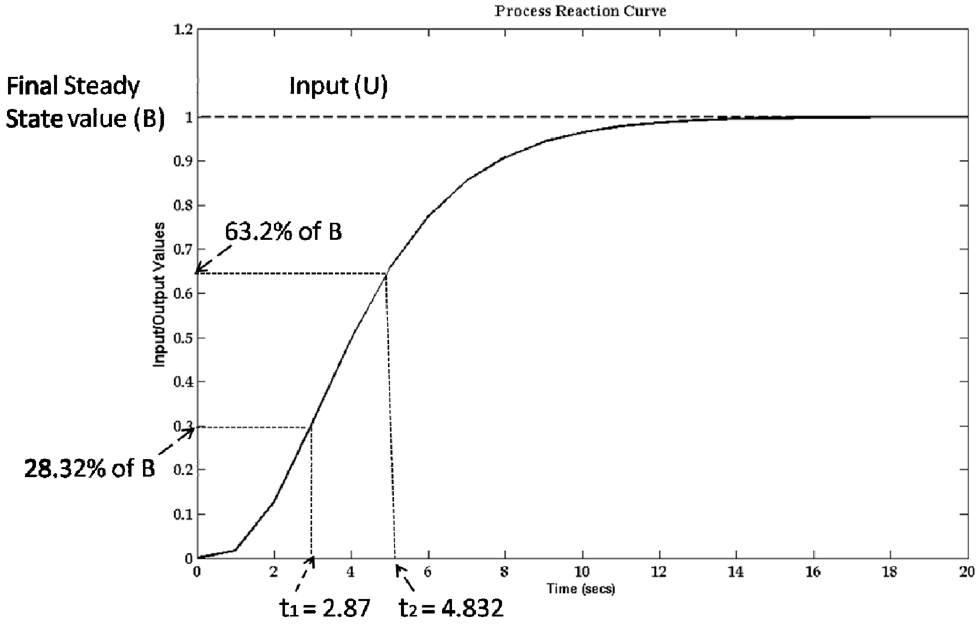


Figure 3. Open Loop Step Response of the higher order process for the step input of magnitude 'u'

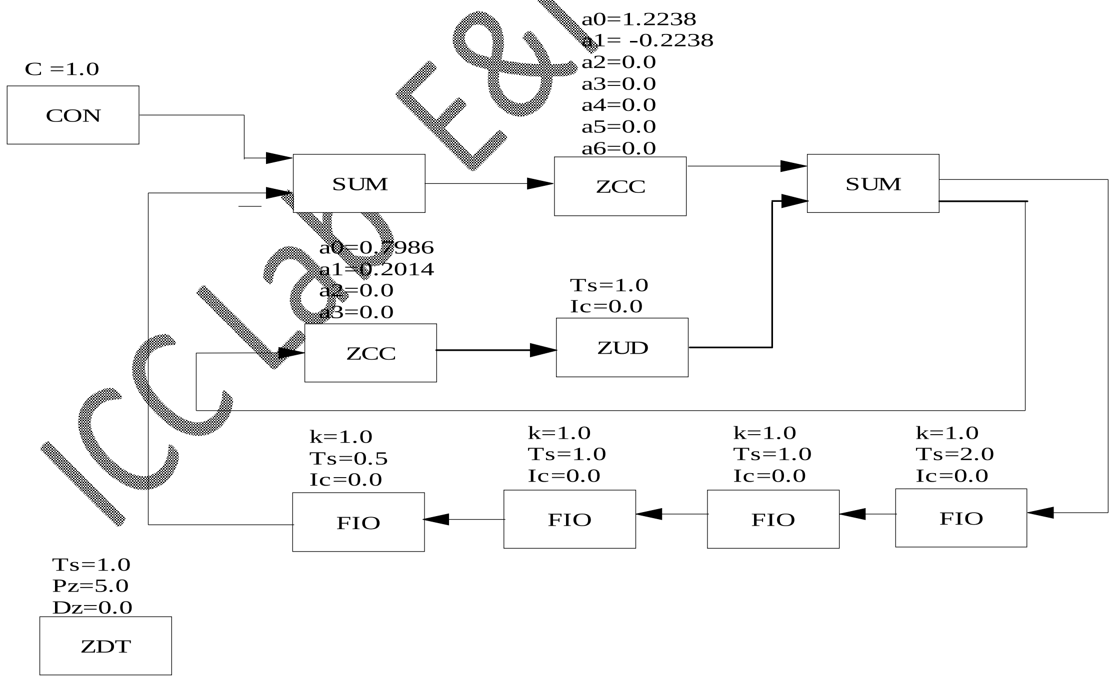


Figure 4. TUTSIM Block Diagram of Kalman's Controller for Actual Process

$$= m_0 + (m_1 - m_0)z^{-1} + (m_f - m_1)z^{-2}$$
$$= q_0 + q_1 z^{-1} + q_2 z^{-2}$$
$$= Q(z)$$

$$HG_P(z) = \frac{C(z)}{M(z)} = \frac{P(z)}{Q(z)} \qquad (4)$$

Thus the coefficient in H $G_P(z)$ must equal those in P(z) and Q(z)

$$\sum_{i=1}^{2} P_i = P_1 + P_2 = 1 \text{ and} \qquad (5)$$

$$\sum_{i=0}^{2} q_i = q_0 + q_1 + q_2 = \frac{1}{K_P} \qquad (6)$$

These relationships do not generally hold good for the pulse transfer functions, but dividing by sum of the numerator coefficients will ensure that both equations (5) and (6) hold good. Since P(z) and Q(z) are known, the control algorithm D(z) can be derived as

$$D(z) = \frac{\frac{C(z)}{R(z)}}{1 - \left[\frac{C(z)}{R(z)}\right]} \frac{1}{HG_P(z)}$$

$$D(z) = \frac{P(z)}{1 - P(z)} \frac{Q(z)}{P(z)} = \frac{Q(z)}{1 - P(z)} \qquad (7)$$

The equation holds good even if the process has a dead time of N sampling instants. In such case the specification of C(z) must include the appropriate number of zeros.

PROBLEM

Design a Kalman's controller for the process given by

$$G_P(s) = \frac{1}{(0.5s+1)(s+1)(s+1)(2s+1)} \qquad (8)$$

Sampling Time T=5 second.

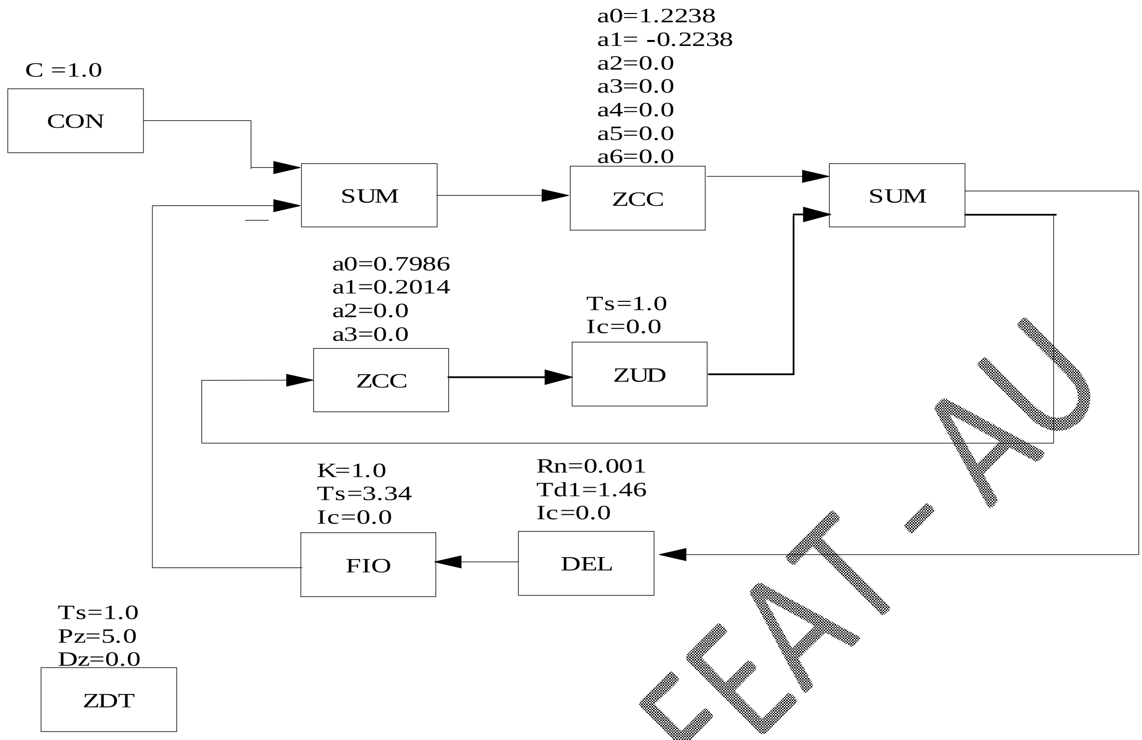


Figure 5. TUTSIM Block Diagram of Kalman's Controller for Approximated Process

DESIGN

First approximate the given process into first order plus dead time model using process reaction curve method. The block diagram schematic is as shown in Figure 2 and the open loop response obtained is shown in Figure 3.

The approximated first order plus time delay model is obtained from Figure 3 and is given by

$$G_p(s) = \frac{1e^{-1.889s}}{(2.943s+1)}$$

The pulse transfer function is given by

$$HG_p(z) = Z[H(s)G_p(s)]$$

$$\text{where } H(s) = \left(\frac{1-e^{-sT}}{s}\right) = \text{zero order hold transfer function}$$

$$HG_p(z) = Z\left[\left(\frac{1-e^{-sT}}{s}\right)\left(\frac{1 * e^{-1.889s}}{(2.943s+1)}\right)\right]$$

$$HG_P(z) = Z[H(s)\,G_P(s)]$$

$$\text{where } H(s) = \left(\frac{1-e^{-sT}}{s}\right) = \text{zero order hold transfer function}$$

$$HG_P(z) = Z\left[\left(\frac{1-e^{-sT}}{s}\right)\left(\frac{1*e^{-1.889s}}{(2.943s+1)}\right)\right]$$

$$= \left(1-z^{-1}\right)*Z\left[\frac{e^{-1.889s}}{s*(2.943s+1)}\right]$$

Sampling time T = 5 seconds

$$\lambda T = t_d$$

$$\lambda T = 1.889$$

$$\lambda = \frac{1.889}{5} = 0.3778$$

$$m = 1-\lambda$$

$$m = 1-0.292 = 0.6222$$

$$\therefore HG_P(z) = \left(1-z^{-1}\right)*Zm\left[\frac{1}{s(2.943s+1)}\right]_{m=0.6222}$$

$$= \left(1-z^{-1}\right)*Zm\left[\frac{1/2.943}{s\left(s+1/2.943\right)}\right]_{m=0.6222}$$

$$= \left(1-z^{-1}\right)*Zm\left[\frac{0.3398}{s(s+0.3398)}\right]_{m=0.6222,\,a=0.3398}$$

$$HG_P(z) = \left(1-z^{-1}\right)*Zm\left[\frac{0.3398}{s(s+0.3398)}\right]_{m=0.6222,\,a=0.3398}$$

We Know

$$Zm\left[\frac{a}{s(s+a)}\right] = \left[\frac{1}{z-1} - \frac{e^{-amT}}{z-e^{-aT}}\right]$$

$$HG_P(z) = \left(1-z^{-1}\right)*Zm\left[\frac{0.3398}{s(s+0.3398)}\right]_{m=0.6222,\,a=0.3398}$$

$$= \left(1 - z^{-1}\right)\left[\frac{1}{z-1} - \frac{e^{-(0.3398)(0.6222)(5)}}{Z - e^{-(0.3398)(5)}}\right]$$

$$= \left(1 - z^{-1}\right)\left[\frac{1}{z-1} - \frac{0.3475}{z - 0.18293}\right]$$

$$= \frac{z-1}{z}\left[\frac{1}{z-1} - \frac{0.3475}{z - 0.1829}\right]$$

$$= \frac{1}{z} - \frac{(0.3475z - 0.3475)}{(z^2 - 0.1829z)}$$

$$= \frac{z^2 - 0.1829z - 0.3475z^2 + 0.3475z}{z^3 - 0.1829z^2}$$

$$= \frac{0.6525z^2 + 0.1646z}{z^3 - 0.1829z^2}$$

$$= \frac{z(0.6525z + 0.1646)}{z(z^2 - 0.1829z)}$$

$$HG_P(z) = \frac{0.6525z + 0.1646}{z^2 - 0.1829z}$$

$$= \frac{z\left(0.6525 + 0.1646z^{-1}\right)}{z^2\left(1 - 0.1829z^{-1}\right)}$$

$$= \frac{z^{-1}\left(0.6525 + 0.1646z^{-1}\right)}{\left(1 - 0.1829z^{-1}\right)}$$

NORMALISATION

For normalization add the numerator co-efficients then divide both numerators and denominator by the added numerator co-efficient value.

Added numerator co-efficient value = 0.6525+0.1646=0.8171

$$G(z) = \frac{P(z)}{Q(z)} = \frac{\frac{0.6525}{0.8171} + \frac{0.1646}{0.8171}z^{-1}}{\frac{z}{0.8171} - \frac{0.1829}{0.8171}}$$

$$\frac{P(z)}{Q(z)} = \frac{0.7986 + 0.2014z^{-1}}{1.2238z - 0.2238}$$

$$= \frac{0.7986z^{-1} + 0.2014z^{-2}}{1.2238 - 0.2238z^{-1}}$$

$$D(z) = \frac{Q(z)}{1 - P(z)}$$

$$D(z) = \frac{1.2238 - 0.2238z^{-1}}{1 - \left(0.7986z^{-1} + 0.2014z^{-2}\right)}$$

$$D(z) = \frac{\left(1.2238 - 0.2238z^{-1}\right)}{\left(1 - 0.7986z^{-1} - 0.2014z^{-2}\right)}$$

$$\frac{m(z)}{e(z)} = \frac{\left(1.2238 - 0.2238z^{-1}\right)}{\left(1 - 0.7986z^{-1} - 0.2014z^{-2}\right)}$$

$$m(z) - 0.7986z^{-1}m(z) - 0.2014z^{-2}m(z) = 1.2238e(z) - 0.2238e(z)z^{-1}$$

Taking inverse Z-transform

$$m_n = 0.7986\, m_{n-1} + 0.2014\, m_{n-2} + 1.2238\, e_n - 0.2238\, e_{n-1}$$

EXPERIMENTAL PROCEDURE

i. Give the step input for the given higher order process as shown in Figure 2.

ii. Obtain the process reaction (S- shaped) curve as shown in Figure 3.

iii. From the process reaction curve find the steady state value(B), t_1 and t_2 where t_1=time corresponds to 28.32% of B and t_2= time corresponds to 63.2% of B.

iv. Approximate the first order plus dead time model using the given formulae as $G_P(s) = \frac{K_P e^{-tds}}{\tau s + 1}$ where K_P is the process gain, τ is the process time constant and t_d is the process dead time.

$$K = \frac{\text{Change in steady state value}}{\text{change in the input}} = \frac{B}{U}$$

$$\tau = 1.5*\left(t_2 - t_1\right) \qquad (9)$$

$$t_d = t_2 - \tau$$

v. Derive pulse transfer function $HG_P(z)$.

vi. Obtain the digital controller D(z) as given in the design.

vii. Implement the transfer function of the dead beat controller by using Z-blocks in TUTSIM for actual and approximated process as shown in Figure 5 and Figure 6 respectively.

viii. Analyse the response and comment on your results.

RESULT

The closed loop response of a sampled data system to a unit step change in the set point has been obtained after designing a Kalman's controller.

Exp. No :
Date :

DESIGN AND REALIZATION OF DIGITAL FILTER

AIM

To design and realize a digital IIR Low Pass Filter using Bilinear Transformation and transform to High pass, Band pass and Band stop filters using MATLAB.

Theory

A digital filter is essentially a computational process (algorithm) that converts one sequence of number x(n) representing the input, to another sequence y(n) that represents the output.

The input-output difference equation that relates the output to the input can be expressed in the discrete time domain as a summation of the form,

$$y[n] = \sum a_i x[n-1] - \sum b_i y[n-1]$$

or in the Z-domain as

$$G(z) = \frac{n(z)}{d(z)} = \frac{\sum a_i z^{-1}}{\sum b_i z^{-1}}$$

Digital filters are classified based on the

1. Impulse Response duration and
2. Realization form.

Impulse Response Duration:

a) An Infinite Impulse Response (IIR) digital filter has infinite number of samples in its impulse response h[n].
b) A Finite Impulse Response (FIR) digital filter has finite number of samples in its impulse response h[n].

Realization Forms:

a) In a *Recursive Realization* digital filter the output is dependent on the input and the previous values of the output. In a recursive digital filter, both the coefficients a_i and b_i are present.
b) In a *Non-Recursive Realization* digital filter the output depends on present and past values of the input only. In a non-recursive digital filter, only the coefficients a_i are present, i.e., b_i=0.

Filter design methods have been well established, along with their prototype circuits. Thus, we can choose the appropriate prototype to satisfy the requirements. Transformation methods are also available to map an analog prototype to an equivalent digital filter. Three well known transformation methods are as follows.

i. The ***Impulse Invariant Method*** which produces a digital filter whose impulse response consists of the sampled values of the impulse response of an analog filter.
ii. The ***Step Invariant Method*** which produces a digital filter whose step response consists of the sampled values of the step response of an analog filter.
iii. The ***Bilinear Transformation*** which uses the transformation

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

to transform the left half of the S plane into the interior of the unit circle in the Z plane.

Problem

Design a digital low pass single pole filter with 3 db cut off frequency of 0.2π using bilinear transformation method applied to analog filter H(s). Also transform the digital low pass filter to digital high pass, band pass and band stop filter.

$$H(s) = \frac{\Omega_c}{s+\Omega_c} \qquad (1)$$

Solution:

$$\Omega_c = \frac{2}{T}\tan\left(\frac{\omega_c}{2}\right)$$

Cut off frequency of digital filter $\omega_c = 0.2\pi$ (given)

$$\Omega_c = \frac{2}{T}\tan\left(\frac{0.2\pi}{2}\right)$$

$$\Omega_c = \frac{0.65}{T}$$

Applying bilinear transformation to equation (1), the digital low pass filter H(z) is obtained

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

$$H(z) = \frac{\frac{0.65}{T}}{\left[\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+\frac{0.65}{T}\right]}$$

$$H(z) = \frac{0.245(1+z^{-1})}{1-0.509z^{-1}}$$

$$H(z) = \frac{Y(z)}{X(z)}$$

Therfore,

$$\frac{Y(z)}{X(z)} = \frac{0.245(1+z^{-1})}{1-0.509z^{-1}} \qquad (2)$$

Cross multiplying and taking inverse 'Z' transform

$$y(n) = 0.509\,y(n-1) + 0.245x(n) + 0.245x(n-1) \qquad (3)$$

Equation (3) gives the difference equation digital low pass filter with input x(n) and output y(n).

Transformation from low pass to high pass filter

Let ω_p be the cut off frequency of high pass filter and $\omega_c = 0.2\pi$

Let $\omega_p = 0.4\pi$ rad/sec.

To transform low pass to high pass filter replace z^{-1} in transfer function of low pass filter (derived in equation (2)) by equation (4)

$$z^{-1} = -\left(\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}\right) \qquad (4)$$

Where $$\alpha = -\left(\frac{\cos\left(\frac{\omega_p+\omega_c}{2}\right)}{\cos\left(\frac{\omega_p-\omega_c}{2}\right)}\right)$$

Calculating α by substituting ω_p and ω_c

$$\alpha = -\left(\frac{\cos\left(\frac{0.4\pi+0.2\pi}{2}\right)}{\cos\left(\frac{0.4\pi-0.2\pi}{2}\right)}\right)$$

$$\alpha = -0.6180$$

Now z^{-1} in equation (4) becomes

$$z^{-1} = -\left(\frac{z^{-1}-0.6180}{1-0.6180z^{-1}}\right) \qquad (5)$$

Replacing z^{-1} in equation (2) by equation (5), we get the transfer function of high pass filter as in equation (6)

$$\frac{Y(z)}{X(z)}=\frac{0.245\left[1-\left(\frac{z^{-1}-0.6180}{1-0.6180z^{-1}}\right)\right]}{1+0.509\left(\frac{z^{-1}-0.6180}{1-0.6180z^{-1}}\right)} \qquad (6)$$

Simplifying

$$\frac{Y(z)}{X(z)}=\frac{0.3694(1-z^{-1})}{0.6854-0.109z^{-1}} \qquad (7)$$

Cross multiplying and taking inverse 'Z' transform

$$y(n)=0.5783x(n)-0.5783x(n-1)+0.1590y(n-1) \qquad (8)$$

Equation (8) gives the difference equation digital high pass filter with input x(n) and output y(n).

Transformation from low pass to band pass filter

Let ω_l and ω_u be the lower and upper cut off frequencies of band pass filter

$$\omega_l = 0.2\pi \text{ rad/sec}; \qquad \omega_u = 0.4\pi \text{ rad/sec}$$

To transform low pass to band pass filter replace z^{-1} in transfer function of low pass filter (derived in equation (2)) by equation (9)

$$z^{-1}=-\left(\frac{z^{-2}-\left(\frac{2\alpha k}{1+k}\right)z^{-1}+\left(\frac{k-1}{k+1}\right)}{\left(\frac{k-1}{k+1}\right)z^{-2}-\left(\frac{2\alpha k}{k+1}\right)z^{-1}+1}\right) \qquad (9)$$

$$\text{where } \alpha=\left(\frac{\cos\left(\frac{\omega_u+\omega_l}{2}\right)}{\cos\left(\frac{\omega_u-\omega_l}{2}\right)}\right) \quad \text{and} \quad k=\cot\left(\frac{\omega_u-\omega_l}{2}\right)\tan\left(\frac{\omega_c}{2}\right)$$

Substituting $\alpha=0.6180$ and $k=1$ in equation (9)

$$z^{-1}=-\left(\frac{z^{-2}-0.6180z^{-1}}{1-0.6180z^{-1}}\right) \qquad (10)$$

Replacing z^{-1} in equation (2) by equation (10), we get the transfer function of band pass filter as in equation (11)

$$\frac{Y(z)}{X(z)}=\frac{0.245\left[1-\left(\frac{z^{-2}-0.6180z^{-1}}{1-0.6180z^{-1}}\right)\right]}{1+0.509\left(\frac{z^{-2}-0.6180z^{-1}}{1-0.6180z^{-1}}\right)} \qquad (11)$$

Simplifying

$$\frac{Y(z)}{X(z)} = \frac{0.245(1 - z^{-2})}{1 - 0.9326z^{-1} - 0.509z^{-2}} \tag{12}$$

Cross multiplying and taking inverse 'Z' transform

$$y(n) = 0.245x(n) - 0.245x(n-2) + 0.9326y(n-1) + 0.509y(n-2) \tag{13}$$

Equation (13) gives the difference equation digital band pass filter with input x(n) and output y(n).

Transformation from low pass to band stop filter

Let ω_l and ω_u be the lower and upper cut off frequencies of band stop filter

$$\omega_l = 0.4\pi \text{ rad/sec}; \qquad \omega_u = 0.6\pi \text{ rad/sec}$$

To transform low pass to band stop filter replace z^{-1} in transfer function of low pass filter (derived in equation (2)) by equation (14)

$$z^{-1} = \left(\frac{z^{-2} - \left(\frac{2\alpha k}{1+k}\right) z^{-1} + \left(\frac{1-k}{1+k}\right)}{\left(\frac{1-k}{1+k}\right) z^{-2} - \left(\frac{2\alpha k}{k+1}\right) z^{-1} + 1} \right) \tag{14}$$

$$\text{where} \quad \alpha = \left(\frac{\cos\left(\frac{\omega_u + \omega_l}{2}\right)}{\cos\left(\frac{\omega_u - \omega_l}{2}\right)} \right) \quad \text{and} \quad k = \tan\left(\frac{\omega_u - \omega_l}{2}\right) \tan\left(\frac{\omega_c}{2}\right)$$

Substituting $\alpha = 0$ and $k = 0.105$ in equation (14)

$$z^{-1} = \left(\frac{z^{-2} + 0.8089}{1 + 0.8089z^{-1}} \right) \tag{15}$$

Replacing z^{-1} in equation (2) by equation (15), we get the transfer function of band stop filter as in equation (16)

$$\frac{Y(z)}{X(z)} = \frac{0.245\left[1 + \left(\frac{z^{-2} + 0.8089}{1 + 0.8089z^{-1}}\right)\right]}{1 - 0.509\left(\frac{z^{-2} + 0.8089}{1 + 0.8089z^{-2}}\right)} \tag{16}$$

Simplifying

$$\frac{Y(z)}{X(z)} = \frac{0.4432(1 - z^{-2})}{0.5883 + 0.3z^{-2}} \tag{17}$$

Cross multiplying and taking inverse 'Z' transform

$$y(n) = 0.7536x(n) + 0.7536x(n-2) - 0.5099y(n-2) \qquad (18)$$

Equation (18) gives the difference equation digital band stop filter with input x(n) and output y(n).

PROCEDURE

1. Design the Low pass filter using given specification and transform to High pass, Band pass, band stop filters.
2. Enter the program in m-file.
3. Run the program for various input frequencies and observe the output.

Program 1: LOW PASS FILTER

```
% program for low pas filter
clear all;
close all;
y(1)=0;
x(1)=0;
% cut off frequency =0.2*pi
for n=2:1000
    x(n)=sin(0.02*3.14*n)+sin(0.7*3.14*n);
  y(n)=0.509*y(n-1)+0.245*(x(n)+x(n-1));
end
for n=1:1000
  input(n)=x(n);
  out(n)=y(n);
end
subplot(4,2,1)
plot(1:1000,input)
title('input')
subplot(4,2,2)
plot(1:1000,out)
title('output')
num=[0.245 0.245];
den=[1 -0.509];
w=0:0.1*pi:pi;
y=freqz(num,den,w);
subplot(4,1,3)
plot(w,abs(y))
title('magnitude response')
```

```
xlabel('frequency')
ylabel('magnitude')
subplot(4,1,4)
plot(w,angle(y))
title('phase response')
xlabel('frequency')
ylabel('phase')
```

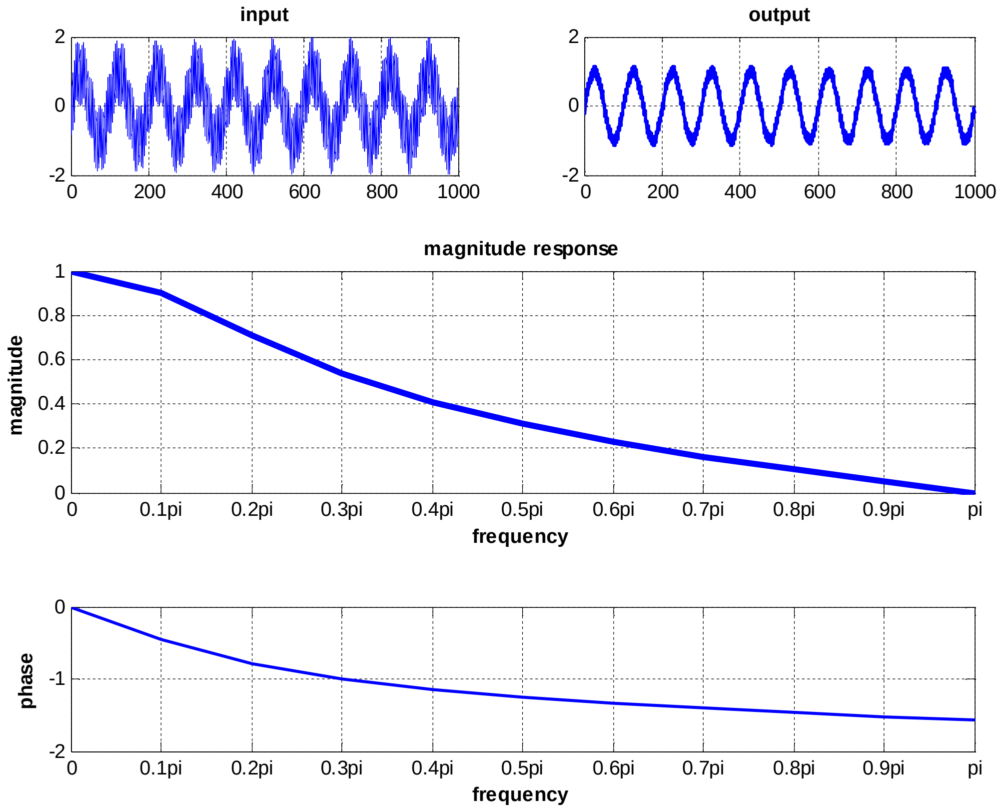


Fig.1 Results for Low Pass Filter

Program 2: HIGH PASS FILTER

```
clear all;
close all;
%HIGH PASS FILTER
y(1)=0;
x(1)=0;
for n=2:1000
   x(n)=sin(0.01*3.14*n);  %cutoff frequency=0.4*3.14
   y(n)=0.159*y(n-1)+0.5783*(x(n)-x(n-1));
end
for n=1:1000
   out(n)=y(n);
   input(n)=x(n);
end
subplot(4,2,1)
plot(1:1000,input)
title('input')
subplot(4,2,2)
plot(1:1000,out)
title('output')
num=[0.3964 -0.3964];
den=[0.6854 -0.109];
w=0:0.1*pi:pi;
y=freqz(num,den,w);
subplot(4,1,3)
plot(w,abs(y))
title('magnitude response')
xlabel('frequency')
ylabel('magnitude')
subplot(4,1,4)
plot(w,angle(y))
title('phase response')
xlabel('frequency')
ylabel('phase angle')
```

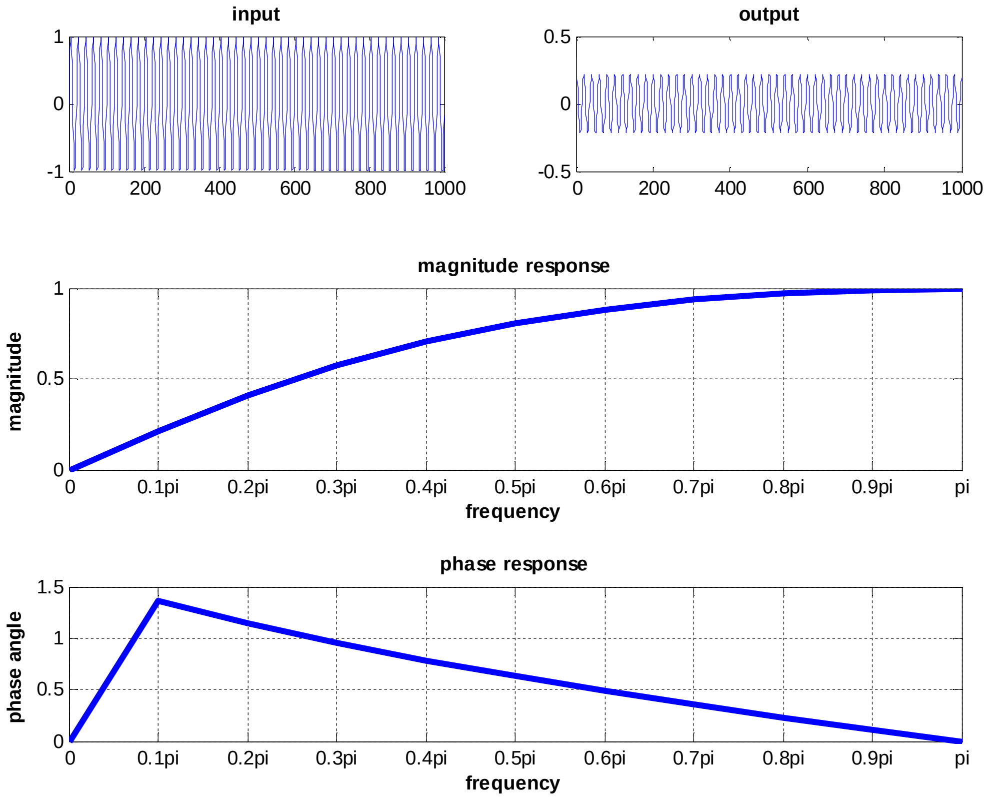


Fig.2 Results for High Pass Filter

Program 3: BAND PASS FILTER

```
clear all;
close all;
%BANDPASS FILTER
y(1)=0;
x(1)=0;
x(2)=0;
y(2)=0;
for n=3:1000
   x(n)=sin(0.2*3.14*n);
   %lower cutoff frequency=0.2*3.1;         %upper cutoff frequency=0.4*3.14
   y(n)=0.245*x(n)-0.245*x(n-2)+0.9326*y(n-1)-0.509*y(n-2);
end
for n=1:1000
   input(n)=x(n);
   out(n)=y(n);
end
subplot(4,2,1)
plot(1:1000,input)
title('input')
subplot(4,2,2)
plot(1:1000,out)
title('output')
num=[0.245 0 -0.245];
den=[1 -0.9326 0.509];
w=0:0.1*pi:pi;
y=freqz(num,den,w);
subplot(4,1,3)
plot(w,abs(y))
title('magnitude response')
xlabel('frequency')
ylabel('magnitude')
subplot(4,1,4)
plot(w,angle(y))
title('phase response')
xlabel('frequency')
ylabel('phase')
```

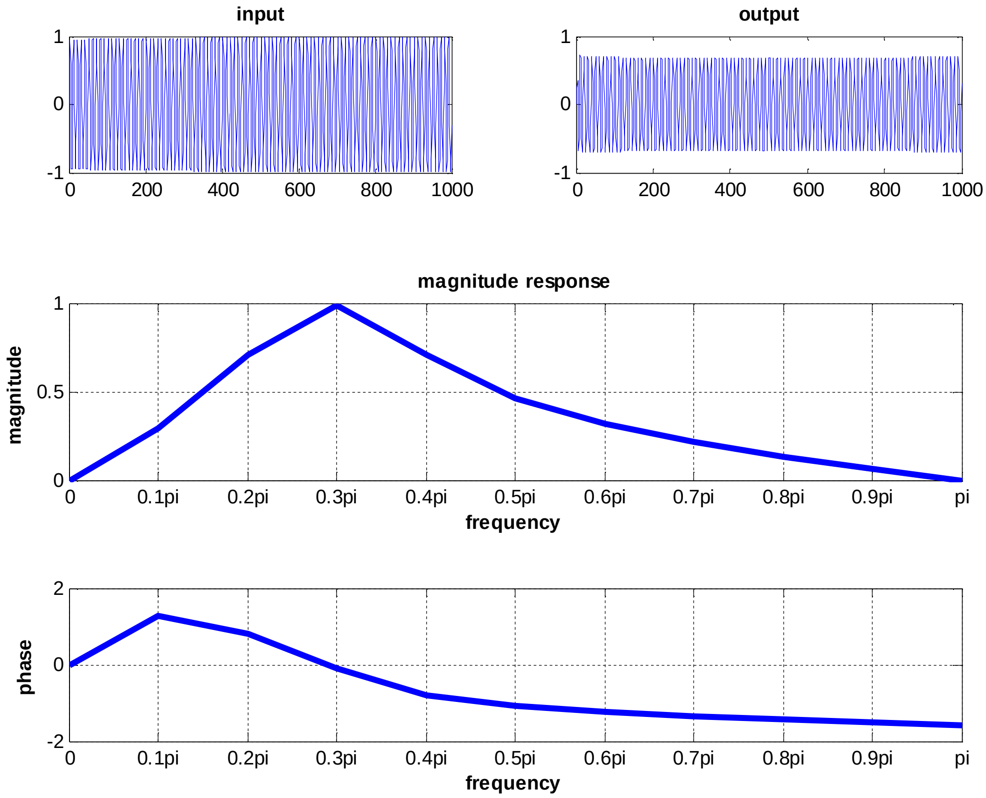


Fig.3 Results for Band Pass Filter

Program 4: BAND STOP FILTER

```
clear all;
close all;
%BANDSTOP FILTER
y(1)=0;
x(1)=0;
x(2)=0;
y(2)=0;
for n=3:1000
   x(n)=sin(0.01*3.14*n);
   %lower cutoff frequency=0.4*3.14;      %upper cutoff frequency=0.6*3.14;
   y(n)=-0.51*y(n-2)+0.7536*x(n)+0.7536*x(n-2);
end
for n=1:1000
   input(n)=x(n);
   out(n)=y(n);
end
subplot(4,2,1)
plot(1:1000,input)
title('input')
subplot(4,2,2)
plot(1:1000,out)
title('output')
num=[0.4432 0 0.4432];
den=[0.5883 0 0.3];
w=0:0.1*pi:pi;
y=freqz(num,den,w);
subplot(4,1,3)
plot(w,abs(y))
title('magnitude response')
xlabel('frequency')
ylabel('magnitude')
subplot(4,1,4)
plot(w,angle(y))
title('phase response')
xlabel('frequency')
ylabel('phase')
```

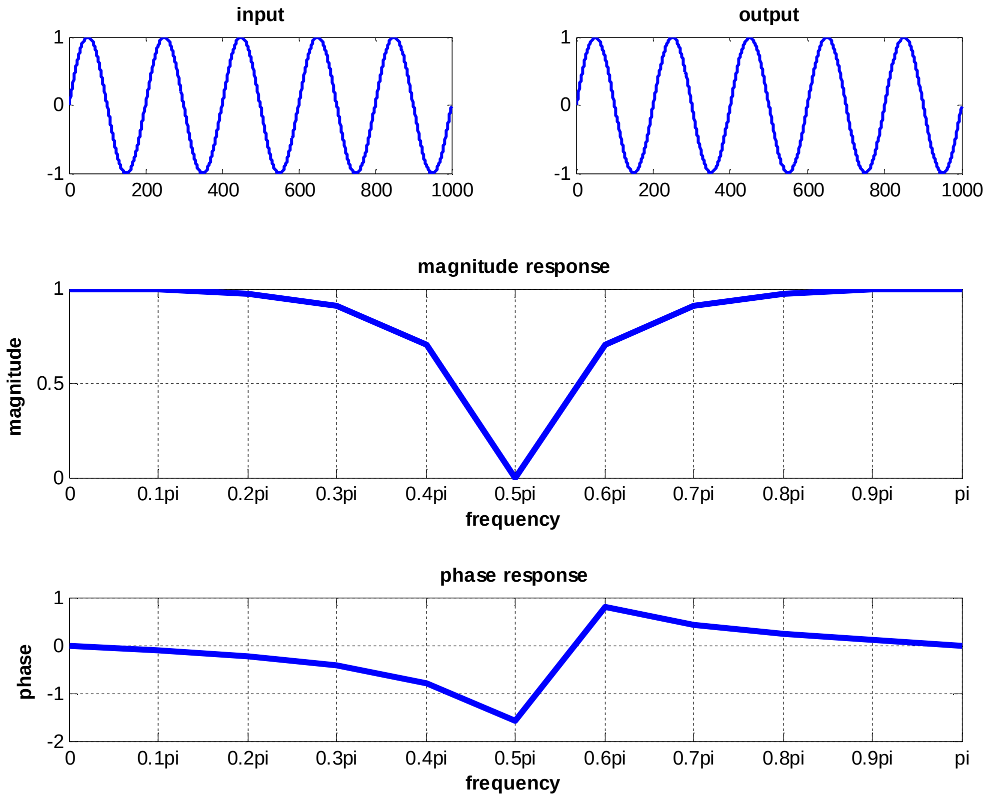


Fig.3 Results for Band Stop Filter

RESULT

Thus design and simulation of various filters has been done using SIMULINK.

Expt. No. **:**

Date **:**

DESIGN OF SAMPLED DATA CONTROL SYSTEM WITH DHALIN'S CONTROLLER AND SIMULATION USING TUTSIM.

AIM

To design a dahlin's controller for a given process and to find the closed loop response of the sampled data system to a unit step change in the set point incorporating dahlin's algorithm

THEORY

The block diagram of Direct Digital Control Loop (DDC) is as shown in Figure 1 (Exp.1). For a set point change the closed loop response is given by

$$\ddot{Y}(z) = \frac{HG_P(z)D(z)}{1 + HG_P(z)} \ddot{Y}_{SP}(z) \qquad (1)$$

Where $HG_P(z)$ =Pulse transfer function of the process with zero order hold.

In DDC for a given step change in the set point the discrete-time response should be specified. Then HG_P (z), $\ddot{Y}$ (z) and $\ddot{Y}_{sp}$(z) are known we can solve equation (1) with respect to unknown D (z).

The transfer function of the digital controller is given by

$$D(z) = \frac{1}{HG_P(z)} * \frac{\left(\frac{Y(z)}{Y_{SP}(z)}\right)}{1 - \left(\frac{Y(z)}{Y_{SP}(z)}\right)} \qquad (2)$$

PROBLEM

Design a Dahlin's controller for the process.

$$G_P(s) = \frac{1}{(0.5s+1)(s+1)(s+1)(2s+1)} \text{ with Sampling Time T=5 seconds} \qquad (6)$$

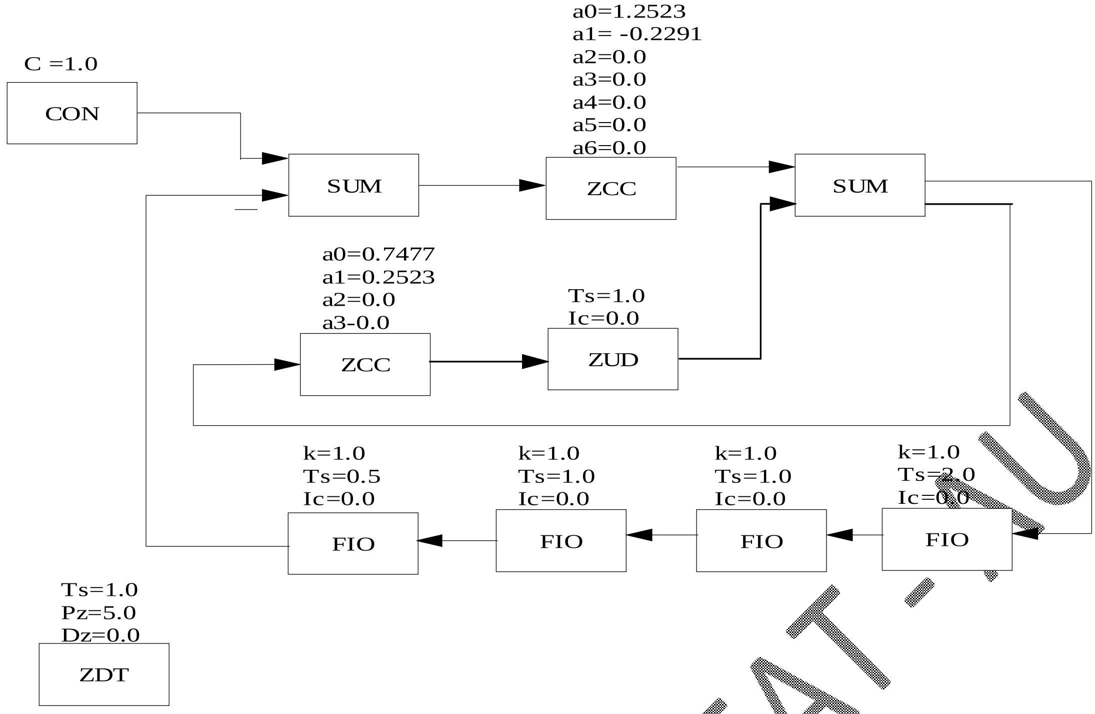


Figure 4. TUTSIM Block Diagram of Dahlin's Controller for actual process

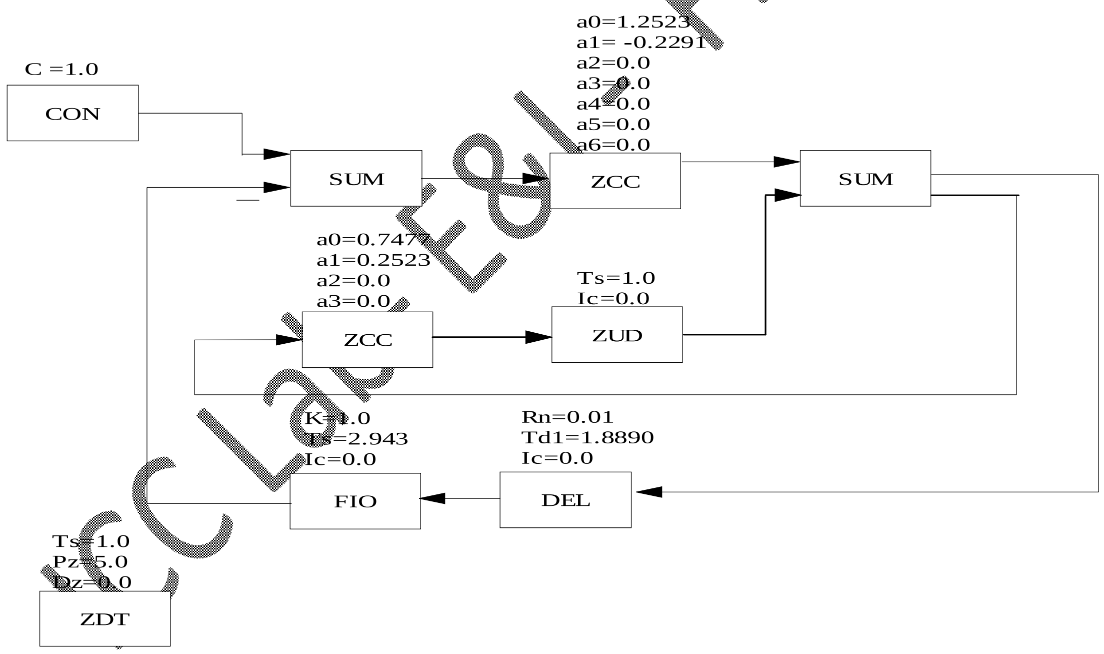


Figure 5. TUTSIM Block Diagram of Dahlin's Controller for approximated process

DESIGN

First approximate the given process into first order plus dead time model using process reaction curve method. The block diagram schematic is as shown in Figure 2 and the response obtained is as shown in Figure 3.

The approximated first order time delay model is obtained from Figure 3 and is given by

$$G_P(s) = \frac{1\,e^{-1.889s}}{(2.943s+1)} \tag{7}$$

The pulse transfer function is given by

$$HG_P(z) = Z[H(s)\,G_P(s)]$$

where $H(s)$ = zero order hold transfer function and is given by

$$H(s) = \left(\frac{1-e^{-sT}}{s}\right)$$

$$HG_P(z) = Z\left[\left(\frac{1-e^{-sT}}{s}\right)\left(\frac{1*e^{-1.889s}}{(2.943s+1)}\right)\right]$$

$$= (1-Z^{-1})*Z\left[\frac{e^{-1.889s}}{S*(2.943S+1)}\right]$$

Sampling time T = 5 seconds

$$\lambda T = t_d$$

$$\lambda T = 1.889$$

$$\lambda = \frac{1.889}{5} = 0.3778$$

$$m = 1-\lambda$$

$$m = 1-0.3778 = 0.6222$$

$$\therefore HG_P(z) = (1-z^{-1})*Zm\left[\frac{1}{s(2.943s+1)}\right]_{m=0.6222}$$

$$= (1-z^{-1})*Zm\left[\frac{\frac{1}{2.943}}{s\left(s+\frac{1}{2.943}\right)}\right]_{m=0.6222}$$

$$= (1-z^{-1})*Zm\left[\frac{0.3398}{s(s+0.3398)}\right]_{m=0.6222,\,a=0.3398}$$

$$HG_P(z) = (1-z^{-1})*Zm\left[\frac{0.3398}{s(s+0.3398)}\right]_{m=0.6222,\,a=0.3398}$$

We Know

$$Zm\left[\frac{a}{S(S+a)}\right] = \left[\frac{1}{z-1} - \frac{e^{-amT}}{z-e^{-aT}}\right]$$

$$HG_P(z) = (1-z^{-1})*Zm\left[\frac{0.3398}{s(s+0.3398)}\right]_{m=0.6222,\,a=0.3398}$$

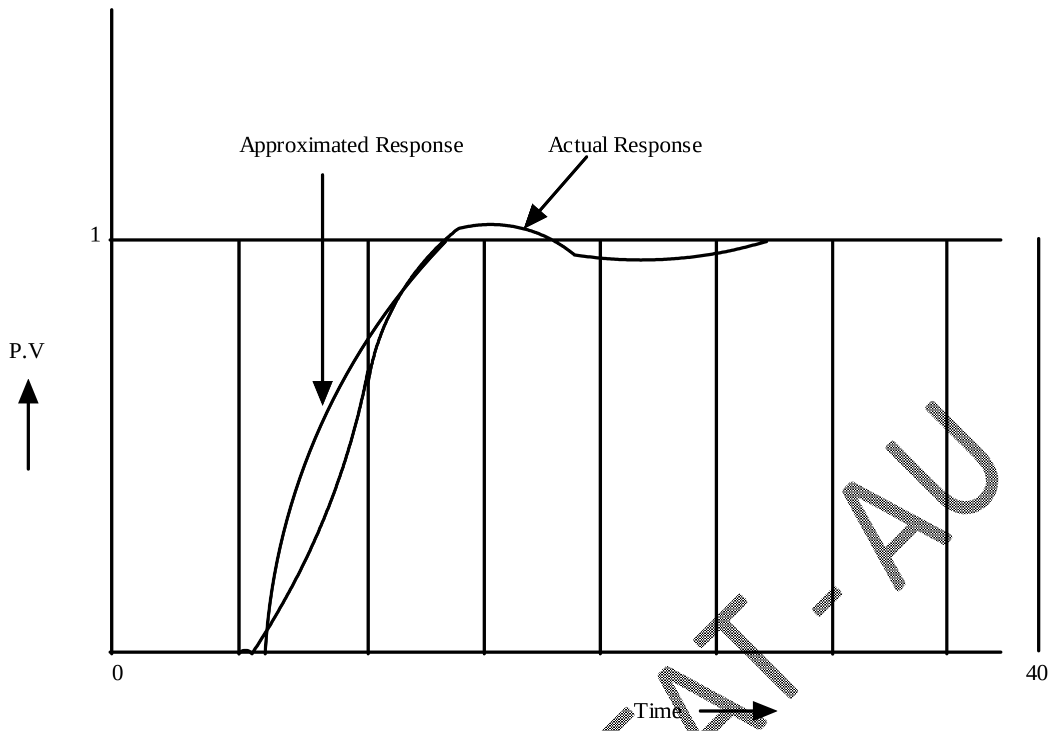


Figure 6. Closed loop Model Response of DAHLIN'S Controller for Actual process and Approximated process

$$
\begin{aligned}
HG_p(z) &= \left(1 - z^{-1}\right)\left[\frac{1}{z-1} - \frac{e^{-(0.3398)(0.6222)(5)}}{z - e^{-(0.3398)(5)}}\right] \\
&= \left(1 - z^{-1}\right)\left[\frac{1}{z-1} - \frac{0.3475}{z - 0.1829}\right] \\
&= \frac{z-1}{z}\left[\frac{1}{z-1} - \frac{0.3475}{z - 0.1829}\right] \\
&= \frac{1}{z} - \frac{\left(0.3475z - 0.3475\right)}{\left(z^2 - 0.1829z\right)} \\
&= \frac{z^2 - 0.1829z - 0.3475z^2 + 0.3475z}{z^3 - 0.1829z^2} \\
&= \frac{0.6525z + 0.1646}{z^2 - 0.1829z}
\end{aligned}
$$

$$
\begin{aligned}
HG_p(z) &= \frac{z\left(0.6525 + 0.1646z^{-1}\right)}{z^2\left(1 - 0.1829z^{-1}\right)} \\
&= \frac{z^{-1}\left(0.6525 + 0.1646z^{-1}\right)}{\left(1 - 0.1829z^{-1}\right)}
\end{aligned}
$$

$$D(z)=\frac{1}{HG_P(z)} * \frac{(1\text{-}\alpha)\, z^{-(N+1)}}{(1\text{-}\alpha\, z^{-1}) - (1-\alpha) z^{-(N+1)}} \quad where\ \alpha = e^{-(T/\tau)}$$

$$\alpha = e^{-(5/2.943)} = 0.1829$$

$$N = td/T = 0 \ (N\ is\ the\ integral\ \ multiples\ \ of\ deadtime)$$

$$D(z)=\frac{(z-0.1829)}{\left(0.6525+0.1646z^{-1}\right)}\ \frac{(1-0.1829)\, z^{-(0+1)}}{\left(1-0.1829z^{-1} - 0.8171z^{-1}\right)}$$

$$D(z)=\frac{(z-0.1829)}{\left(0.6525+0.1646\, z^{-1}\right)}\left[\frac{0.8171z^{-1}}{1-z^{-1}}\right]$$

$$\frac{m(z)}{e(z)} = \frac{\left(0.8171\text{-}0.1495z^{-1}\right)}{\left(0.6525 - 0.4879z^{-1} - 0.1646z^{-2}\right)}$$

Cross multiplying ,

$$0.6525m(z) - 0.4879z^{-1}\, m(z) - 0.1646z^{-2}\, m(z) = 0.8171e(z) - 0.1495e(z)\, z^{-1}$$

Taking inverse Z-transform

$$m_n = 0.7477m_{n-1} + 0.2523m_{n-2} + 1.2523e_n - 0.2291e_{n-1}$$

EXPERIMENTAL PROCEDURE

(i) Give the step input of magnitude 'u' for the given higher order process as shown in Figure 2.

(ii) Obtain the process reaction (S-shaped) curve as shown in Figure 3.

(iii) From the process reaction curve find the steady state value (B), t_1 and t_2 where t_1=time corresponds to 28.3% of B and t_2= time corresponds to 63.2% of B.

(iv) Approximate the first order plus dead time model using the given formulae as $G_P(s) = \frac{K_P e^{-tds}}{\tau s + 1}$ where K_P is the process gain, τ is the process time constant and t_d is the process dead time.

$$K = \frac{\text{Change in steady state value}}{\text{change in the input}} = \frac{B}{U}$$

$$\tau = 1.5 * (t_2 - t_1) \qquad (8)$$

$$t_d = t_2 - \tau$$

(v) Derive pulse transfer function $HG_P(z)$.

(vi) Obtain the digital controller D(z) as given in the design.

(vii) Implement the transfer function of the dahlin's controller by using Z-blocks in TUTSIM for actual and approximated process as shown in Figure 4 and Figure 5 respectively.

(viii) Analyze the response and comment on your results.

INFERENCE

From the closed loop response it is observed that the dahlin's response is similar to a First Order response with dead time (Without any Oscillations)

RESULT

The closed loop response of a sampled data system to a unit step change in the set point has been obtained after designing a Dahlin's controller.

Expt. No. **:**

Date **:**

STUDY OF LABVIEW SOFTWARE

AIM

To familiarize the LabVIEW software and to develop suitable blocks using LabVIEW for the given exercises.

INTRODUCTION TO LABVIEW:

LabVIEW is a graphical programming environment. A program written in LabVIEW is called Virtual Instrument (VI). Each VI must have a Front Panel and a Block Diagram. The Front Panel includes various controls and indicators, while the block diagram typically includes functions, terminals and sub VI's.

An object in the front panel has its counterpart called a terminal in the block diagram. All controls and indicators have their corresponding terminals in the block diagram. It is important to label the terminal to identify them.

The toolbar serves as the user interface, providing tools for editing and running the VI. It includes the text settings, move text menus as well as aligning and distributing objects and tools for debugging.

The menu bar includes options for file control, editing, operating a VI, window control and many options. The help options provide object description as well as indepth online help.
It is easy to troubleshoot and modify individual modules without affecting others. The Edit icon in the front panel is used to edit a VI. The icon editor has tools for drawing and entering text in color or black and white. Wires link objects in the block diagram and the wires carry data. The thickness and color of wire represent different data types. Numerical data is represented as floating point numbers, signed and unsigned integers or complex numbers. Format allows the user to express the numbers using floating point notation, scientific notation or engineering notation.

Trouble shooting a VI is necessary when the VI does not run because it has either syntax errors or run time errors. Single stepping, break points and probes help the trouble shooting task.
The timers are useful in providing a time delay or marking time. The 'tick count' timer may be used to time the operation, the 'wait' timer is useful in generating accurate time delays and the

‘wait until next ms multiple’ timer can also be used to provide time delays or time between loop iteration delays.

An array is a collection of objects such as numbers, square LEDs, Boolean switches or other objects. In a one dimensional array objects are placed along a straight line. A two dimensional array is made up of rows and columns. In LabVIEW an array can be made as an array control or array indicator.

A string is a collection of ASCII characters. Strings are used by LabVIEW for text messages, instrument control and for storing data to disk. A string control is used to pass string data to the block diagram. A string indicator is used to display a string generated by the block diagram.

Waveform chart is used to display one or more waveforms. The data to be displayed on the waveform graph must be in the form of an array. The X-Y graph is to plot mathematical functions or curves using the Cartesian co-ordinates. The special tools in the charts and graphs are used to configure and operate them.

The text as well as data that the created VI generates can be saved to a file on the disk. In LabVIEW special procedures must be followed to write text or data to a file on the disk or to retrieve information from the disk.

EXERCISE-1

To convert a temperature value in Celsius to Fahrenheit.

SOLUTION

$F = (C * 1.8) + 32$

Multiply the Celsius value by 1.8 and add 32.

Use the functions on the functions>> numeric palette to build a block diagram as shown in Fig.1

STEP 1

Select a ‘multiply’ function from the numeric palette and drag it to the block diagram. Select a ‘wiring tool’ from tools palette and place it over multiply block so the terminals are high lighted. Right click the mouse on one terminal of ‘multiply’ block and select ‘create control’. A control block is created in front panel, so that we can vary the Celsius temperature from the front panel itself. On another terminal of ‘multiply’ block, right click the mouse and select ‘create constant’ and enter the value 1.8.

STEP 2

Select 'add' block from numeric palette and drag it to block diagram. Output of 'multiply' block is connected to one terminal of 'add' block and to another terminal, right click mouse and select 'create constant' and enter the value 32.

On output terminal of 'add' block 'create indicator'. So, Fahrenheit temperature value is indicated in the front panel.

EXERCISE-2

To convert current (I) in the range 4 to 20mA to pressure signal (P) in the range 3 to 15 psi.

SOLUTION

$P = 3 + ((I - 4) / 16) * 12$

Select 'add', 'multiply' and 'divide' blocks from numeric palette and configure as per exercise 1. Refer Figure 2.

Another method to solve this exercise is by using formula node.

Select functions >> structure palette.

Drag formula node to block diagram. Select 'connect wire tool' and on the frame of formula node right click mouse and select 'add input', 'add output'. Declare input as I and output as P. Create 'control' and 'indicator'. Pressure value will be displayed in the indicator.

EXERCISE-3

a) To fill the tank level up to 100 from 0 feet.
b) To drain the tank level from 100 to 0 feet

SOLUTION

a) Select functions >> structure palette.
   Drag 'for' loop to block diagram. Give 100 as number of iterations. On the 'for' loop add 1 to iteration, connect output of 'add' block to a tank. Tank will fill upto 100 feet. Time can be varied by using timer block. Refer Figure 3(a).

b) Select functions >> structure palette.

Drag 'for' loop to block diagram. Give 100 as number of iterations. On the 'for' loop subtract number of iterations from every iteration. Connect output of subtract block to a tank. Tank will drain from 100 feet. Time can be varied by using timer block. Refer Figure 3(b).

EXERCISE-4

To increase the tank level to a desired set point and to decrease the level once it reaches the set point.

Configure blocks as shown in Figure 4 in similar way as in Exercises 1 to 3.

SIMULATION FOR EXERCISE-1

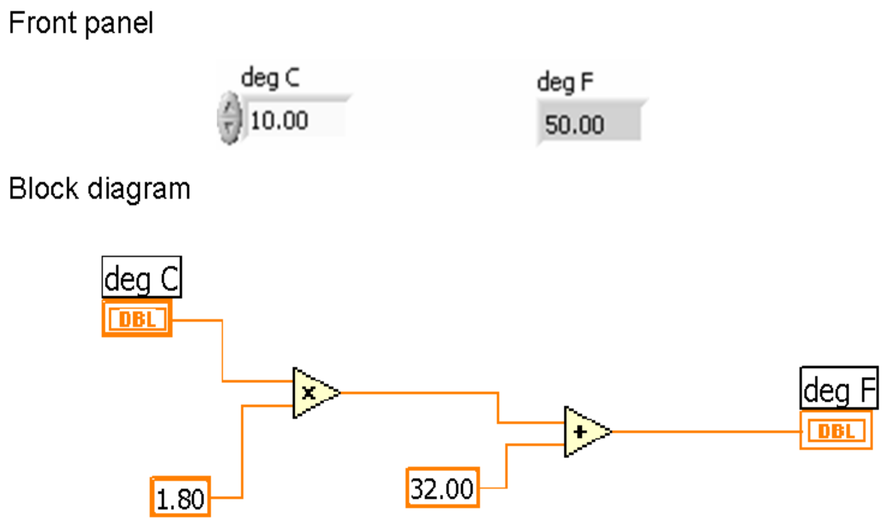
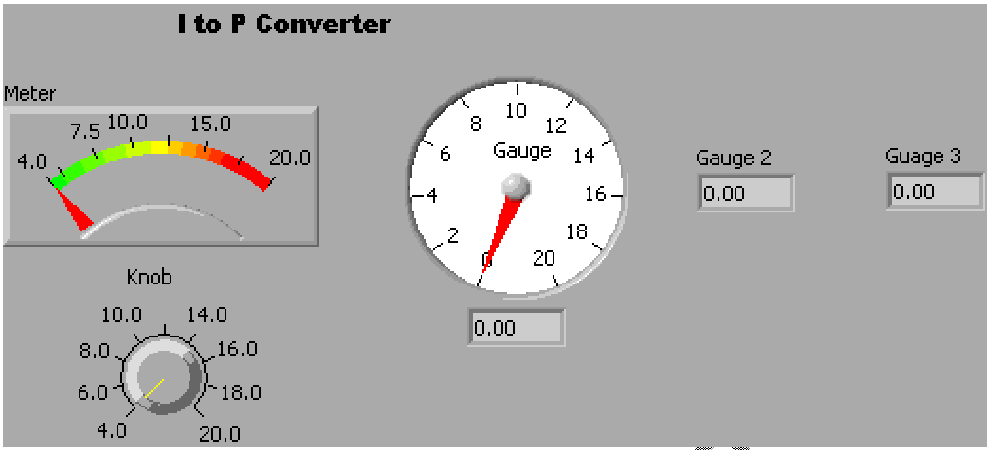


Figure 1. To convert temperature value in Celsius to Fahrenheit

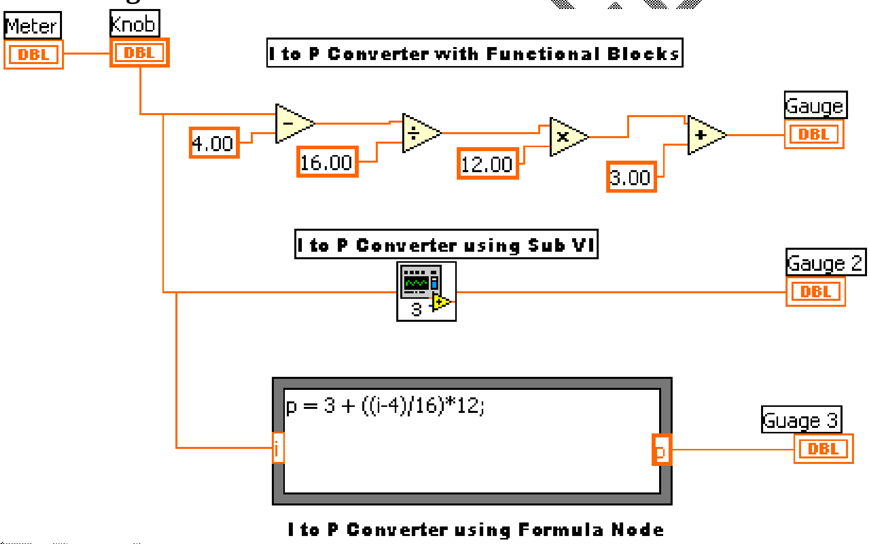
SIMULATION FOR EXERCISE-2

Front Panel



(A)

Block Diagram

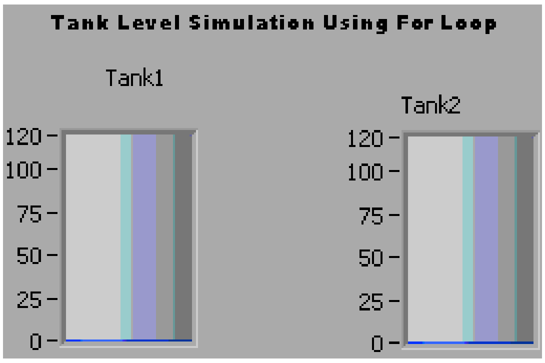


(B)

Figure 2. To convert current signal (4 to 20mA) to pressure signal (3 to 15psi) (A) Front panel (B) Block Diagram

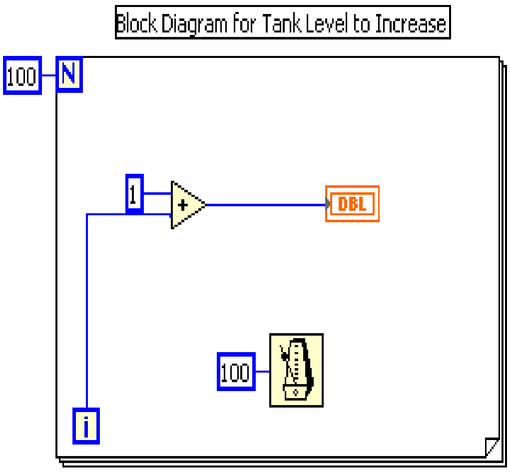
SIMULATION FOR EXERCISE-3

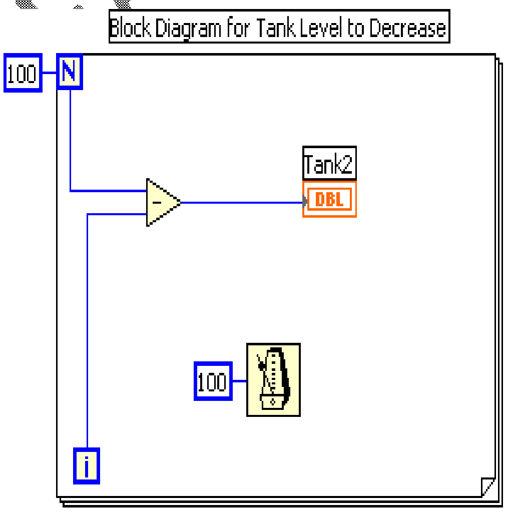
Front Panel



(A)

Block Diagram





(B) (C)

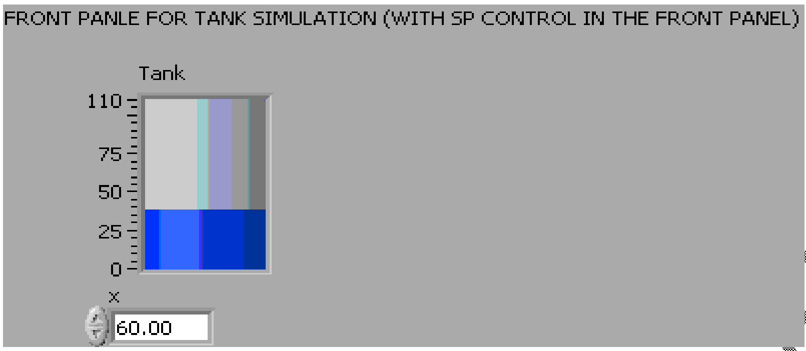
Figure 3. Filling up the tank

(A) Front panel (B) Block Diagram to fill the tank up to 100 from 0 feet

(C) Block Diagram to drain the tank from 100 to 0 feet

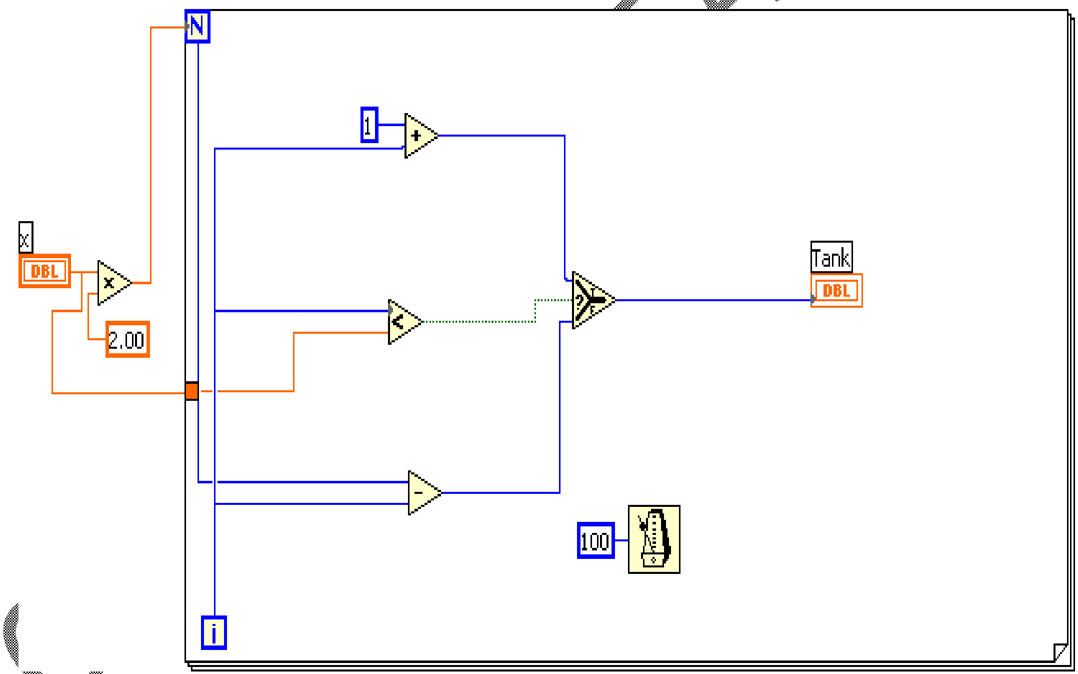
SIMULATION FOR EXERCISE-4

Front Panel



(A)

Block Diagram



(B)

Figure 4. To increase the tank level to a desired set point and to decrease the level once it reaches the set point. (A) Front panel (B) Block Diagram

RESULT

The LabVIEW software was studied.

Ex No:

Date:

STUDY OF SCADA SOFTWARE

AIM

Study and implementation of Intouch wonderware SCADA software for a batch process.

SOFTWARE REQUIRED

SCADA Intouch wonderware software

INTRODUCTION TO SCADA

SCADA is an acronym that stands for Supervisory Control And Data Acquisition. SCADA refers to a system that collects data from various sensors and controllers at a factory, plant or in other remote locations and sends these data to a central computer which then manages and controls the data. As its name indicates, SCADA is not a full control system, but rather focuses on the supervisory level. As such, it is a pure software package that is positioned on top of hardware to which it is interfaced, in general via Programmable Logic Controllers (PLCs), or other commercial hardware modules. SCADA is used as a user interface in DCS architecture.

SCADA is a term that is used broadly to portray control and management solutions in a wide range of industries. Some of the industries where SCADA is used are Water Management Systems, Electric Power, Traffic Signals, Mass Transit Systems, Environmental Control Systems, and Manufacturing Systems. There are many parts of a working SCADA system. A SCADA system usually includes signal hardware (input and output), controllers, networks, user interface (HMI), communication equipments and software. All together, the term SCADA refers to the entire central system. The central system usually monitors data from various sensors that are either in close proximity or off site (sometimes miles away).

For the most part, the brain of a SCADA system is the Remote Terminal Unit (RTU). The Remote Terminal Units consists of a programmable logic converter. The RTU are usually set to specific requirements. One of key processes of SCADA is the ability to monitor an entire system in real time. This is facilitated by data acquisitions including meter reading, checking status of sensors, etc that are communicated at regular intervals depending on the system. Besides the data being used by the RTU, it is also displayed to an operator who is able to interface with the system to override settings or make changes when necessary.

A SCADA system includes a user interface, usually called Human Machine Interface (HMI). The HMI of a SCADA system is where data is processed and presented to be viewed and monitored by a human operator.

Real time control system is built across PLC and controllers. This integrated real time automated control system quickly responds to compensate for any changes in the process. The SCADA system just provides the HMI and supervisory that is placed on top of a real-time control system.

A SCADA system is composed of the following:

1. Field Instrumentation
2. Remote Terminal Unit (RTU)
3. Communications Network
4. Master Terminal Unit (MTU)

1. Field Instrumentation

It refers to the sensors and actuators that are directly interfaced to the plant or equipment. They generate the analog and digital signals that will be monitored by the remote station. Signals are also conditioned to make sure that they are compatible with the inputs/outputs of the (Remote Terminal Unit) RTU of PLC at the remote station. This part is also called as the local system. It is responsible for acquiring process status and actual value of process variables. They communicate with the control center at the lowest level of control hierarchy.

2. The RemoteTerminal Unit

It is installed at the remote plant and controlled by the central host computer. A PLC can act as a RTU. It gathers information from their remote site from various input devices, like valves, pumps, alarms, meters etc. Data is either analog or digital. The RTU's are designed for extremes in temperature, humidity and power outage conditions. This is one of the main advantages of the RTU. It can survive through adverse environmental conditions.

3. The Communications Network

It is the medium for transferring information from one location to another. It is quite possible that systems employ more than one means to communicate to remote sites. SCADA systems are capable of communicating using a wide variety of media such as fiber optics, dial-up, or dedicated voice grade telephone lines, or radio.

4. The Master Terminal Unit (MTU)

It refers to the location of the master or host computer. Several workstations may be configured on the MTU, if necessary. It uses a Man Machine Interface (MMI) program to monitor various types of data needed for the operation. Input to the system normally is initiated from the operator via MTU's keyboard. The MTU monitors information from remote sites and displays information for the operator. The relationship between MTU and RTU is analogous to master and slave.

A typical master unit consists of a base processing unit, operator interface, power supply, modem, input/output cards/relays, enclosure, battery backup, wiring terminals, lighting protection, programming and startup. The block diagram of a SCADA system is shown in the following figure.

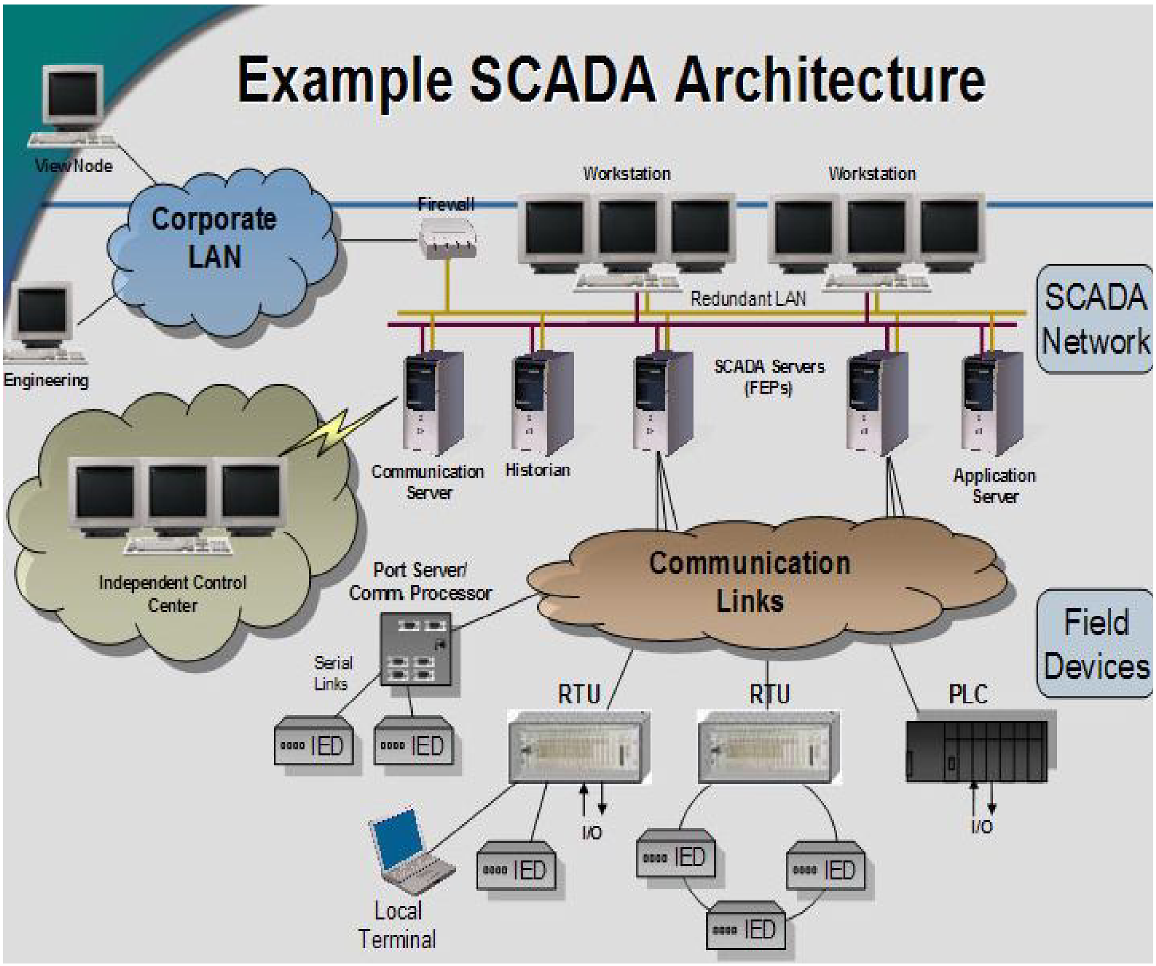


Figure 1. Block diagram of a SCADA system

Features of Scada Systems

A smart way to operate SCADA systems are robust and "failure aware". The SCADA doesn't wait for things to go wrong. It continuously monitors itself for signs of trouble. By running multiple components across robust, redundant processors, when things go wrong, the system is programmed to take immediate steps to circumvent it. The remote telemetry units that perform most of the data acquisition and control employ similar redundant technology, so the personnel can stay focused on recovery of service, not on keeping the SCADA system up and running.

The SCADA system can be programmed to monitor its major physical components and to automatically reconfigure new "virtual" components as necessary whenever failures are detected. Virtual remote telemetry units, for example automatically are redeployed to a functioning server whenever a communication link server or a database server fails.

Well Connected

A SCADA connects to third parties at both the process level(substations, signaling equipments and valves), and at the application level(database, control and management functions). Using standard communication protocols, standard media interfaces, SCADA can interoperatewith a wide range of different telemetry systems.

Alarm functions

It can use voice recording to communicate alarm conditions over mobile phones and telephones, two-way radios, or public address system. Remote computers can dial into the SCADA system by modem and troubleshooting can be done remotely before traveling to the site.

SCADA SOFTWARE(Intouch Wonderware)

Intouch is the quickest and easiest way to create man-machine interface applications for the microsoft windows operating systems. The Intouch software consists of two major components-window maker and window viewer.

Window maker – is a development environment through which application is developed.

Window viewer – is a run time environment through which the executed applications is viewed.

Hisdata - This utility that acts as a DDE server for Intouch encrypted historical log files (*.LGH). hisdata is used with Intouch to retrieve requested historical data from the log file and if required converts it to a comma-separated variable (.CSV) file for use with spreadsheets or text editors.

Tag name data dictionary

The tag name data dictionary (runtime database) is the heart of Intouch. At runtime it contains the current value of all of the items in the database. In order to create the runtime database, Intouch requires information about all of the variables being created. Each variable must be assigned a tagname and type. Intouch also requires additional information in order to acquire the value and convert it for internal use. The tagname data dictionary is the mechanism used to enter this information.

Tags in wonderware and its types

A tag may be considered as a source of information from the real time devices like PLC, DCS, single or multi-loop controller or equipment. A tag typically may be a digital output, or an analog output.

Memory type tag-names

These exist internally within the Intouch application. They are used for creating system constants and simulations. There are four memory types.

- **Memory discrete** –internal discrete tag name with a value of either 0 or 1
- **Memory integer** –a 32 bit signed integer value between 0 to 2147483647

► **Memory real** – floating point memory tagname the floating point value between +/- 3.4 E38

► **Memory message -** text string tagname that can be upto 131 characters long.

Creating scripts in touch

In touch scripting capabilities allows to execute commands and logical operations based on special criteria being met for example a key pressed a window being opened, a value changing etc. By using scripts a wide variety of customized and automated system functions can be created.

- Application scripts
- Window scripts
- Key scripts
- Condition scripts
- Data charge scripts
- Touch push button action scripts

MIMIC DEVELOPMENT

Mimic screen development is an animated representation of the process undertaken. The mimic helps the user to get acquainted with the process and its working can be visualized by initiating the process in mimic screen itself. This type of developing mimics makes the job simpler for the use as he can control the process, verify the sequential execution of the same and also visualize text displays that tell about the current status of process.

Developing the mimic for the process depends on the P&I diagram and the process write up. A process can be a very complex system consisting of large number of field instruments like pressure gauge, valves, actuators, tanks and reactors. All these cannot be developed in a single mimic screen. Generally when a process is considered it will be composed of overall plant view, its individual stages, power distribution, alarms, trending and reports. All these cannot be viewed in a single mimic diagram. So they are divided according to the process being considered. When a number of mimics are developed, there must be parent window that gives the pathway of all the sub windows developed. This parent window must be such that it guides the operator to any part of the process.

To simulate the temperature process control here the various blocks are obtained from the wizards and pasted in the development environment. The blocks had properly assigned a tagname.

Each block is configured and then saved. The alarm conditions for this process are also given in the proper locations.

PROBLEM STATEMENTS

Problem 1

1. Switch on the lamp using Tag Name directory.

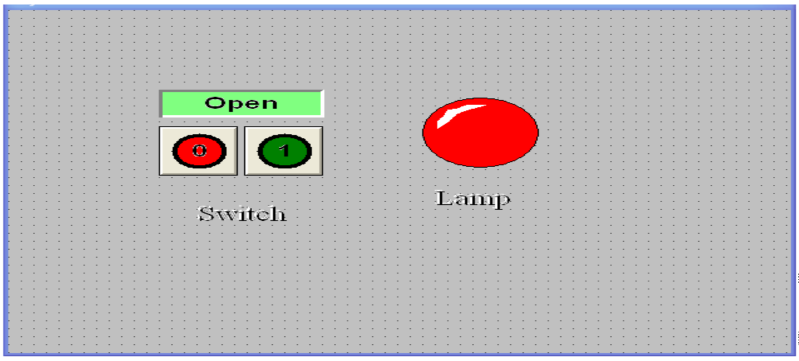


Figure 1: Scada Mimic for Switching the Lamp

If Switch == 1 then

Lamp = 1;

Else

Lampl = 0;

Endif

Problem 2

2. Using the window script repeat the same example above.

Problem 3

3. Using symbols from the Symbols Factory develop a mimic to do the following using scripting:
   i. Start the process using a switch.
   ii. Increase the level in the Tank and indicate the level using a Analog Meter and a Tag Name Display unit.
   iii. Stop the process when the level reaches 95 cms.

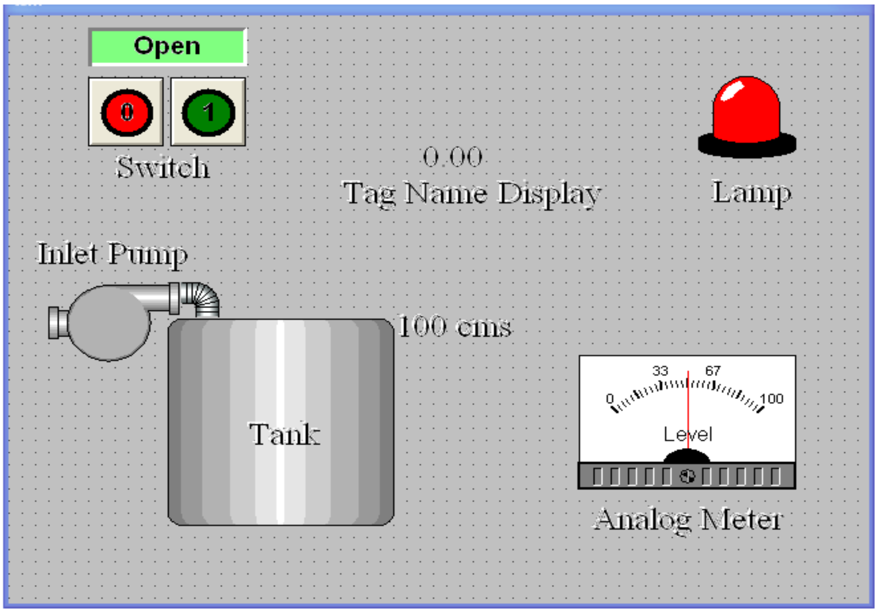


Figure 2: Scada Mimic for Level process

4. Using the symbol factory utility, develop a mimic for the following.

   i. Have a Start/Stop button to run the process.

   ii. Switch ON the pump when the process is started.

   iii. When the level reaches a Set value, Stop the pump and start the Heater.

   iv. When the temperature of the Fluid reaches a Set value, Stop the Heater and Open the Outlet Valve.

   v. When the level in the tank reaches a minimum level, close the Outlet valve.

   vi. Repeat the sequence until the Stop button is pressed.

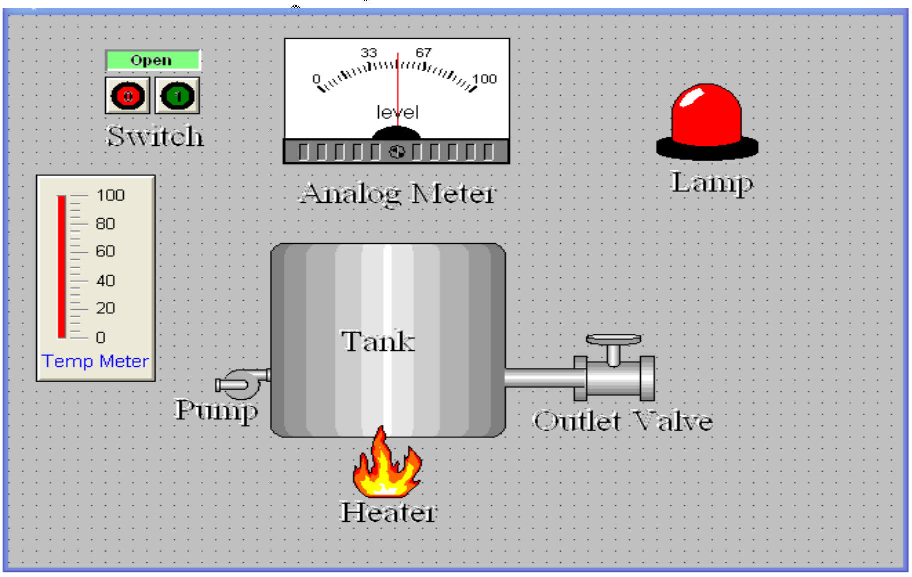


Figure 3: Scada Mimic for batch process

SCRIPTS FOR BATCH PROCESS

Tag Names Used

SW - switch to turn the process ON/OFF

Lmp – Lamp

Pmp – Inlet Pump

Out – Outlet Valve

Heat – Heater

Lev – Level in the Tank

Temp – temperature of the fluid.

Window Script:

```
If sw == 1 then
Lmp = 1
If level <= 5 then
Pmp  =  1;
Out   =  0;
Heat =  0;
Endif;
If level > 95 then
Pmp  =  0;
Out   =  0;
Heat =  1;
Endif;
If temp > 95 then
Pmp = 0;
Out = 1;
Heat = 0;
Endif;
If pmp = = 1 then
Lev = lev + 1;
Else
If out = = 1 then
Temp = 0;
Lev = lev – 1;
Else
If heat = = 1 then
Temp = temp + 1;
Endif;
```

```
Endif;
Endif;
Else
Lmp = 0;
Lev  = 0;
Pmp = 0;
Heat = 0;
Out  = 0;
Endif;
```

RESULT:

A basic mimic is developed for a temperature process and it is simulated in the intouch wonderware environment.

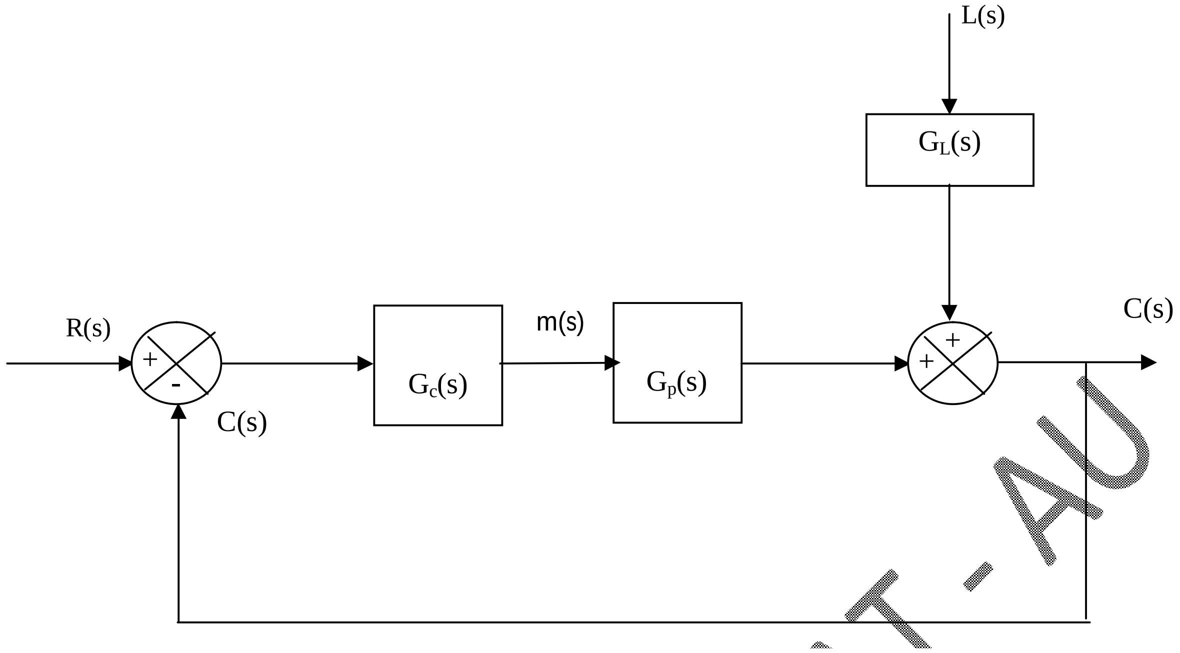


Figure 1. Structure of Feed Back controller

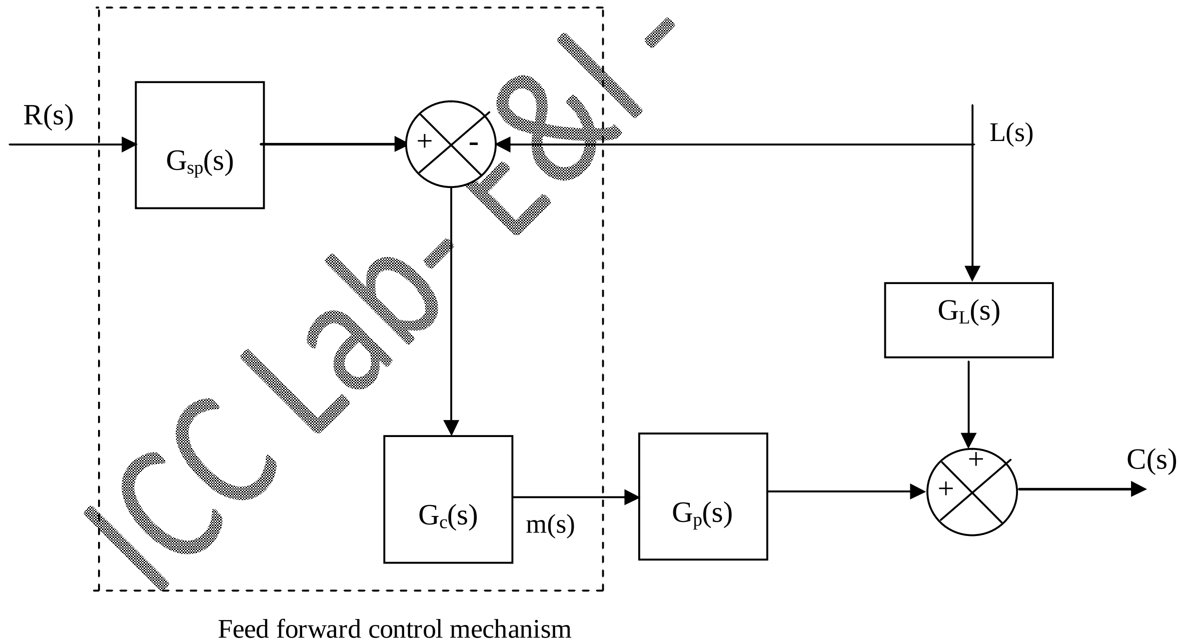


Figure 2. Structure of feed forward controller

Expt. No. :

Date :

DESIGN OF FEED FORWARD – FEEDBACK CONTROLLER AND SIMULATION USING SIMULINK

AIM

To design and implement Feed forward and Feed Back controllers for the given process using MATLAB software.

THEORY

Feed Back Controller

The feed back system has measured value of the controlled variable which is fed back to a comparator. In the comparator, the controlled variable is compared with the reference input (the set point), if there is any difference, the controller will take corrective action in order to bring the controlled variable to the desired set point. The structure of the feedback control scheme is shown in Figure 1.

Thus feedback controller reacts only after it has detected a deviation in output from the set point. Therefore the performance of the feedback controller especially in the presence of disturbance is comparatively not satisfactory.

Advantages

1. Insensitive to modeling error.
2. Insensitive to parameter changes.
3. No need for the 'identification and measurement' of disturbance.

Disadvantages

1. Waits until the effect of disturbance is felt by the process.
2. Unsatisfactory for slow processes.
3. Unsatisfactory for processes with large dad time.
4. May cause instability in the closed loop response.

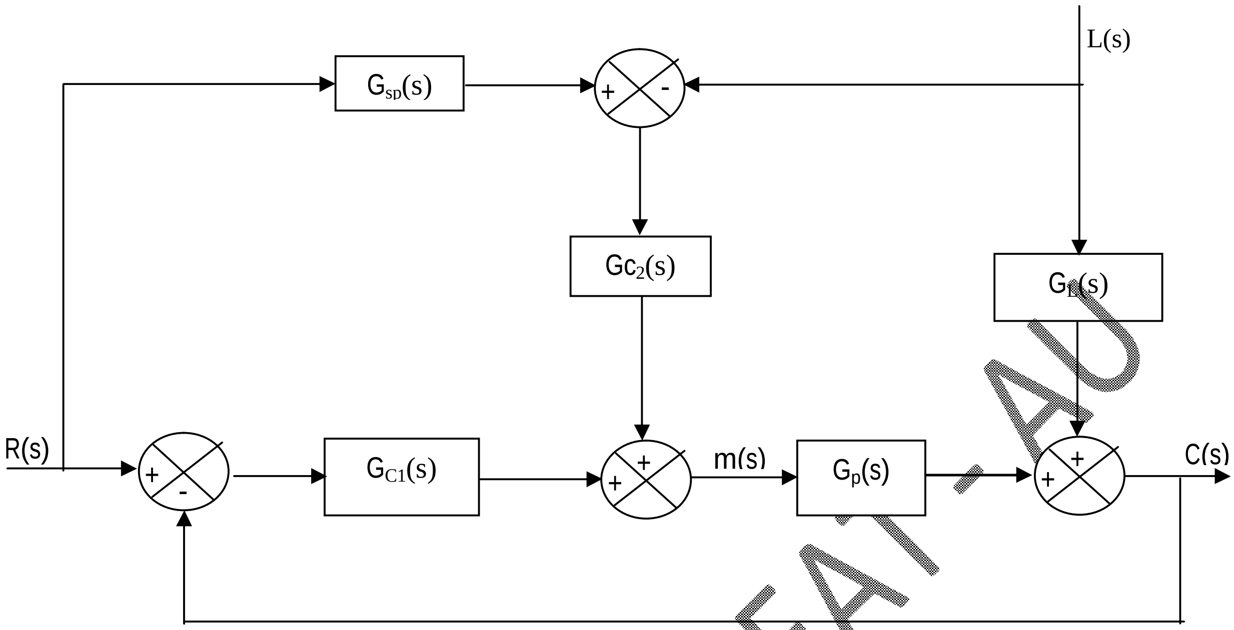


Figure 3. Structure of Feed Forward - Feed Back controller

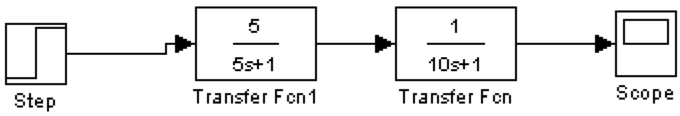


Figure 4. Block diagram of the process to obtain open loop step response

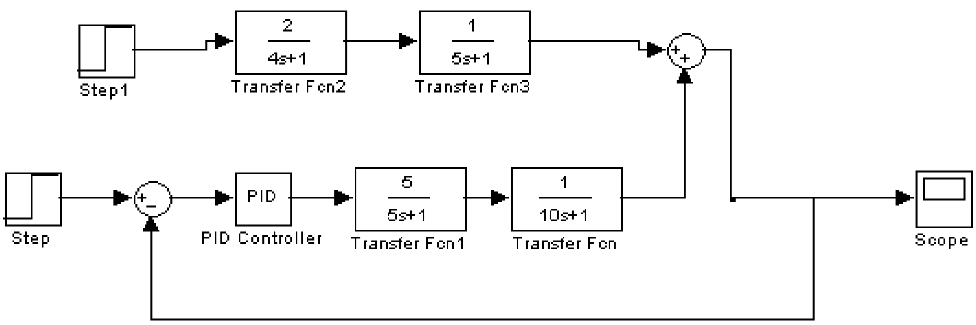


Figure 5 Block diagram of the process with Feedback controller

Feed Forward Controller

The feed forward control configuration measures the load disturbance directly and takes control action to eliminate its impact on the process output. The structure of the feed-forward control scheme is shown in Figure 2. It measures the disturbance directly and then it anticipates the effect that it will have on the process output. Subsequently it changes manipulated variable by such an amount as to eliminate completely the impact of the disturbance on the process output. Control action starts immediately after a change in the disturbance has been detected.

Advantages

1. Acts before the effect of a disturbance has been felt by the process.
2. Good for slow processes.
3. Good for processes with large dead time.
4. Doesn't introduce instability in the closed- loop response.

Disadvantages

1. Requires identification of all possible disturbances.
2. Sensitive to process parameter variation.
3. Good knowledge of the process model required.

Considering the drawbacks of feed-forward controller and advantages of feedback controller it could be expected that a combined feed-forward-feedback control system will retain the superior performance of the controllers. The structure of feed-forward-feedback control scheme is shown figure 3.

DESIGN PROCEDURE

From Figure 1,

$$C(s) = G_p(s)\, M(s) + G_L(s)\, L(s)$$

The value M(s) required to keep output C(s) equal to set point R(s) is

$$M(s) = \left[\frac{R(s)}{G_L(s)} - \frac{G_L(s)}{G_p(s)} L(s) \right]$$

From the above equation the set point element and feed forward element are as below.

Feed forward element, $G_f(s) = \dfrac{G_L(s)}{G_p(s)}$

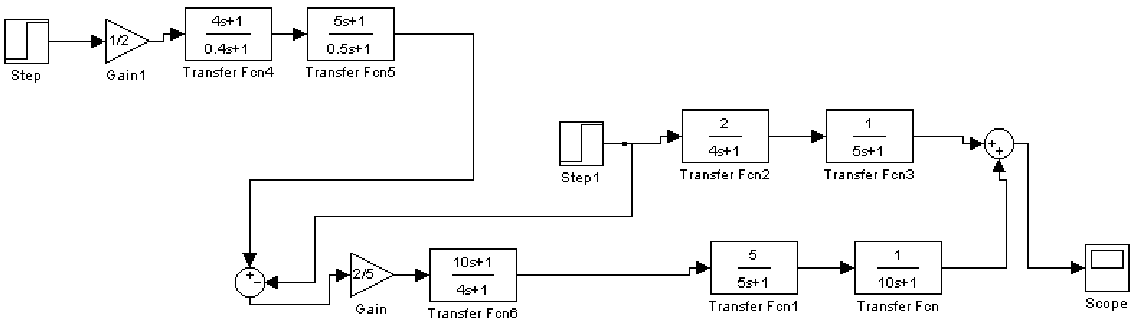


Figure 6. Block diagram of the process with Feed Forward Controller

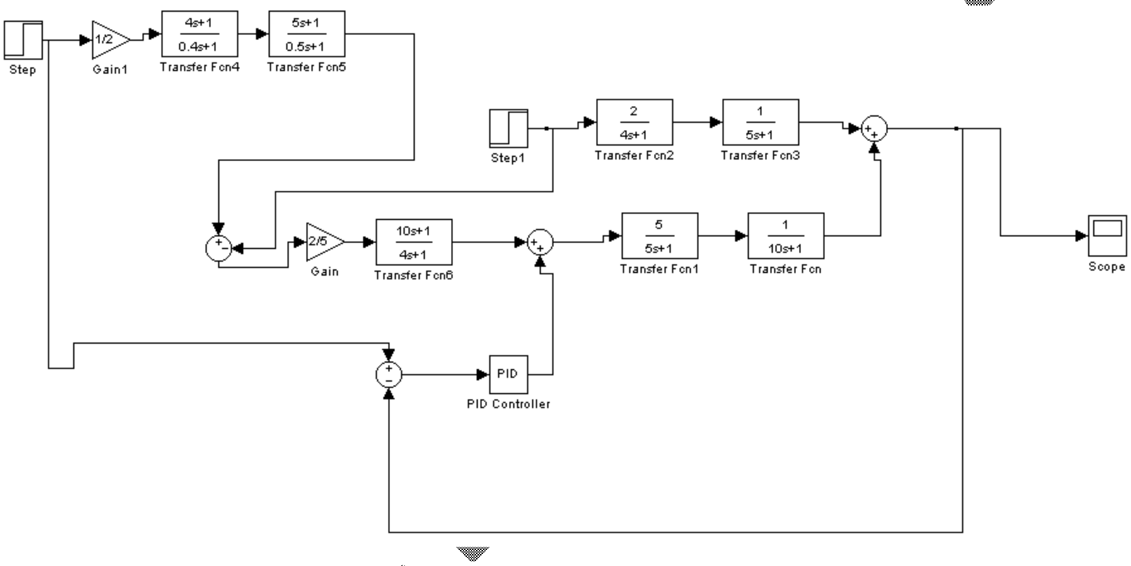


Figure 7. Block diagram of the process with Feed Forward-Feedback controller

Set point element, $G_{sp} = \dfrac{1}{G_L(s)}$

Feed forward controller requires an exact model of the process and its dynamics. The load disturbance has also to be measured.

$$\text{Let}\quad G_p(s) = \frac{5}{(5s+1)(10s+1)}$$

$$G_L(s) = \frac{2}{(4s+1)(5s+1)}$$

Feed forward controller:

$$G_{sp}(s) = \frac{1}{G_L(s)}$$

$$= \frac{(4s+1)(5s+1)}{2}$$

$$= \frac{1}{2} \frac{(4s+1)}{0.4s+1} \frac{(5s+1)}{0.5s+1}$$

$$G_c(s) = \frac{G_L(s)}{G_p(s)} = \frac{\frac{2}{(4s+1)(5s+1)}}{\frac{5}{(5s+1)(10s+1)}} = \frac{2}{5} \frac{10s+1}{4s+1}$$

To design the feed forward-feedback controller approximate the given second-order process to first-order with delay and then find the parameters of PI controller using Z-N (open loop) tuning technique. The simulink block diagram to simulate the open loop response of the process, the process with feedback controller, the process with feed forward controller and the process with Feedforward-feedback controller are shown in Figures 4, 5, 6 and 7 respectively. The simulated open loop and closed loop responses are shown in Figures 8, 9, 10 and 11 respectively.

From the open loop response,

t_1 = 7.59 secs; t_2 = 15.86 secs.

τ = 1.5(15.86- 7.59)

= 12.405 secs.

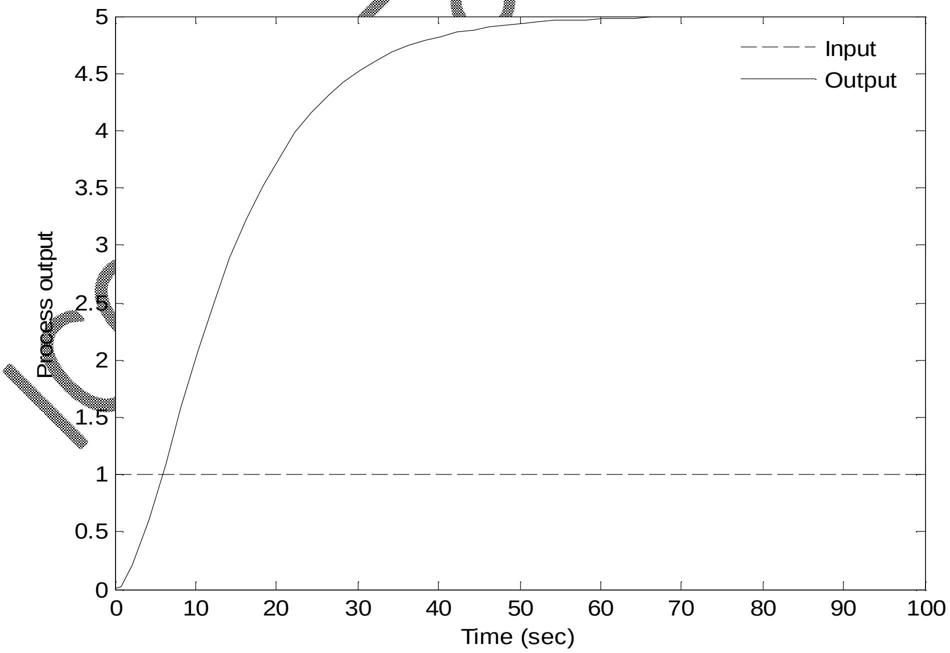


Figure 8. Open loop response of the given process

$t_d = t_2 - \tau = 15.86 - 12.405 = 3.455$

$K_p = 5/1 = 5$

$G(s) = 5/(12.405\, s+1).e^{-3.455s}$.

PI controller settings:

$K_c = 0.9\tau / (K_p.t_d) = 0.6463$.

$T_i = 3.33 * t_d = 11.5167$.; $K_i = 0.0561$

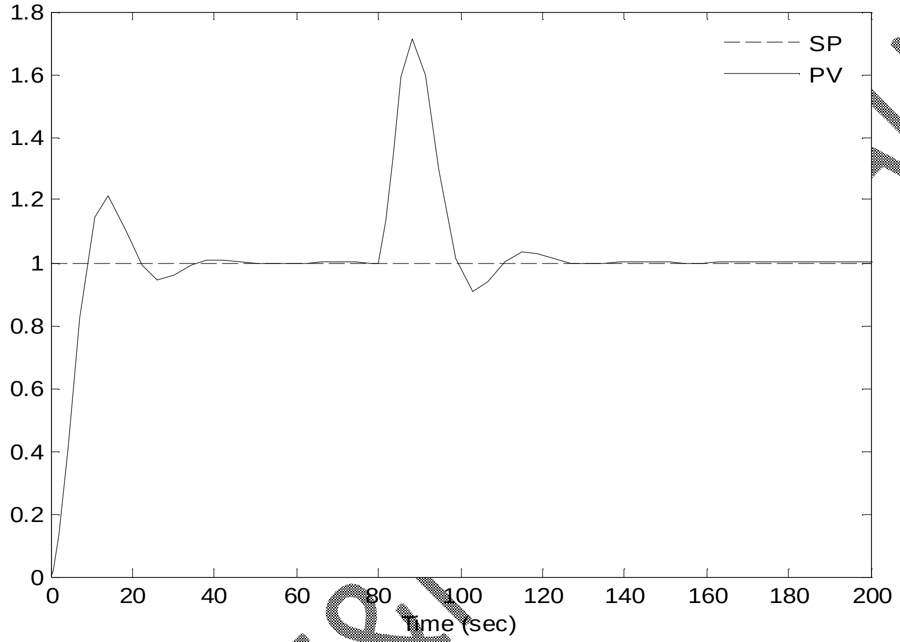


Figure 9. Closed loop response of the process with FeedbackController

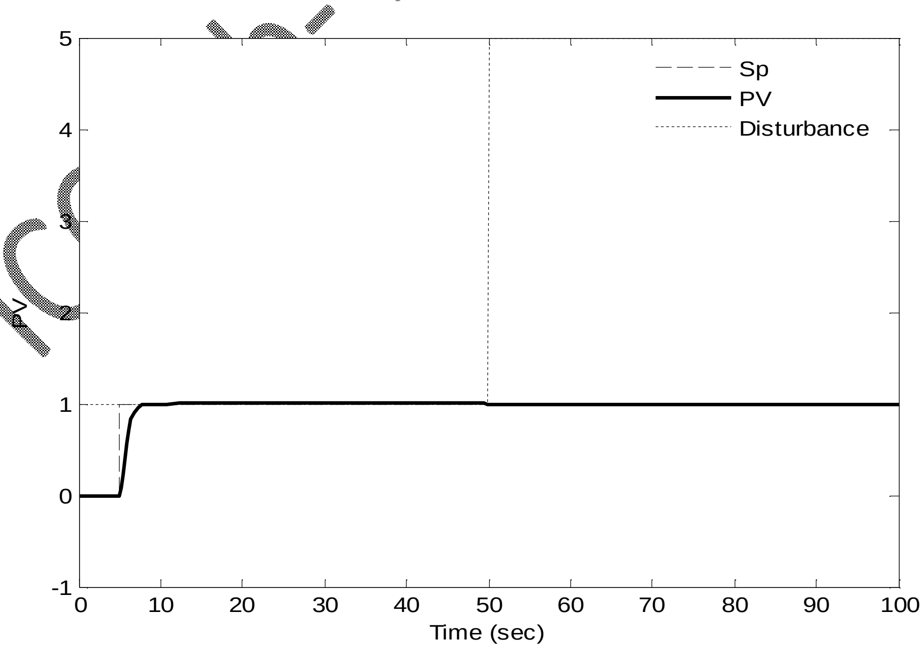


Figure10. Closed loop response of the process with Feed forward Controller

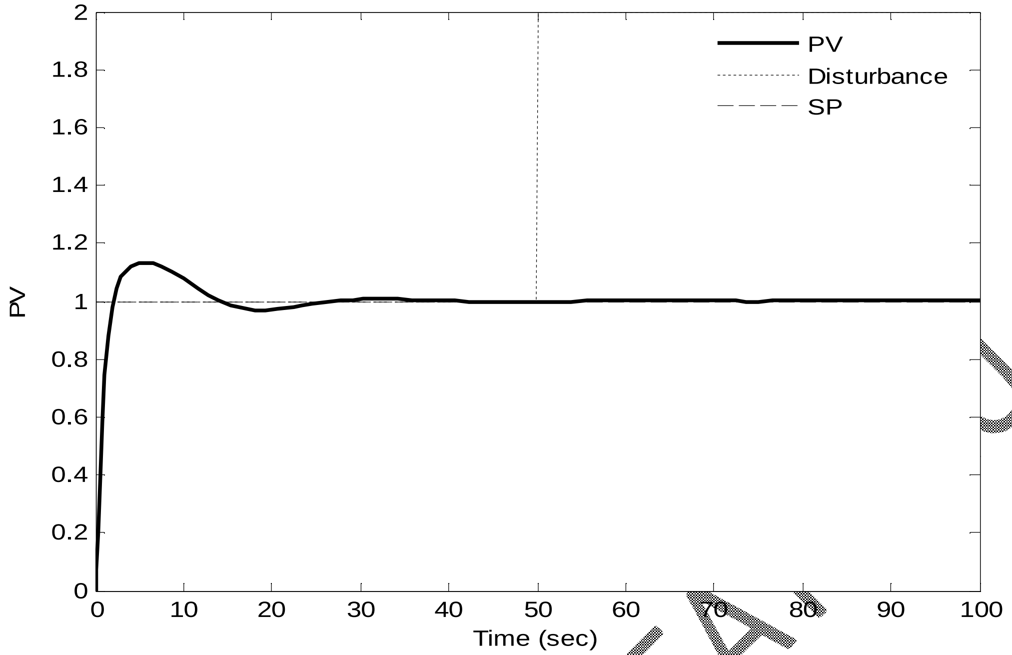


Figure11. Closed loop response of the process with Feedbackand Feedforward Controllers

INFERENCE

Feed back control takes control action only after the disturbance has been felt by the process. Feel forward controller measures the disturbance before it is felt by the process and takes the corrective action accordingly. If there is any model mismatch or if the disturbance model is not perfectly known, then the forward controller fails. On the other hand, the combination of both feed forward-feed back controller is able to maintain the stability as well as it is insensitive to parameter variations.

RESULT

Thus the Feed Forward-Feed Back Controller is designed and implemented using SIMULINK software.

Expt. No. :

Date :

SIMULATION OF A TEMPERATURE PROCESS USING LabVIEW

Aim:

To simulate a temperature control system using LabVIEW software and to study the PID toolkit accompanying LabVIEW.

Introduction to PID control toolkit:

1. The PID toolkit adds sophisticated control algorithms to the instrumentation software development system.
2. By combing PID toolkit with math and logic functions in LabVIEW one can develop programs for the control system.
3. The closed loop or open loop tuning procedures can be adopted.
4. The PID function implement wide range of PID control algorithms.
5. The PID control algorithms incorporate
   - Bump less auto/manual transfer
   - Anti reset windup
   - Direct or Inverse action
   - Manual output adjustments
6. The PID control strategies can be designed and I/O values can be scaled from engineering units to percentage.
7. The National Instruments DAQ hardware can be integrated with the system for real world control.

The PID control algorithm:

In the PID controller, the set point is compared with the process variable to obtain the error.

$$e = sp - pv$$

The controller action can be basically calculated as

$$u(t) = K_c \,(e + 1/T_i \int e\,dt + T_d * de/dt) + bias$$

The required bias is to be added with u(t). Here K_c is the controller gain, T_i is the integral time in minutes and T_d is the derivative time in minutes. The PID VI's implement the 'position form PID algorithm'. The process variable filtering minimizes the effect of noise.

The controller output is limited to the range of specified controller outputs.

If

$u(k) >= u_{max}$ then $u(k) = u_{max}$

$u(k) <= u_{min}$ then $u(k) = u_{min}$

One practical model of the PID controller is

$$u(k) = K_c[(\beta * sp - pv) + 1/T_i \int 1/ (sp - pv)dt - T_d\, dpv_f / dt] + bias$$

where β is the set point factor and PV_f is the value of process variable after filtering. The PID VI's use an integral sum correction algorithm that facilitates anti-reset windup and bump less transfer. Anti reset windup is the upper limit of the controller output, for example when 100% error immediately changes sign, the controller output starts to change appropriately in view of anti-windup feature. The algorithm prevents abrupt controller output changes when it is switched from manual to automatic mode.

Reverse action is the normal controller mode in which controller output decreases as process variable increases. Switching to 'hold' mode or 'manual' mode freezes the output at the current value of the input. All transfers are bumpless. These PID VI's can be called from inside a 'while loop' with fixed cycle time.

PID VI:

The PID VI is the basic PID algorithm. The PID VI has inputs for set point, process variable, manual control and PID parameters. The PID parameter input is cluster of three values proportional gain (K_c), integral time (T_i) and derivative time (T_d). The 'options' input specify additional parameters like hold, PB/K_c, Auto/ Manual and Reverse/Forward. The Boolean input 'hold' is used to hold the controller output at current state. The Boolean input PB/K_c specifies whether the proportional input parameter is the gain (K_c) or proportional band (PB). Gain is related to PB as

$$K_c = 100 / (\% PB)$$

Additionally the ranges for set point and controller output can be set.

Trapezoidal integration is used to avoid sharp changes in integral action when there is a PV or SP jump.

$$u_i(k) = K_c / T_i \sum[\{ e(i) + e(i - 1)\}/ 2] *\Delta t [1 / \{ 1 + (10 * e(i^2) / SP_{rng}{}^2)\}]$$

where SP_{rng} = set point range.

Because of the abrupt changes in the set point only filtered PV is applied to derivative action instead of error 'e' to avoid derivative kick.

$$u_d(k) = -K_c\, T_d / \Delta t \; \{ PV_f(k) - PV_f(k\text{-}1)$$

Now the controller output is

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

where

$u_p = K_c * e(k)$

The PID software VI:

This VI calculates the analog value of the process variable and set point using PID algorithm. The block diagram of a PID VI is shown in Figure 1.

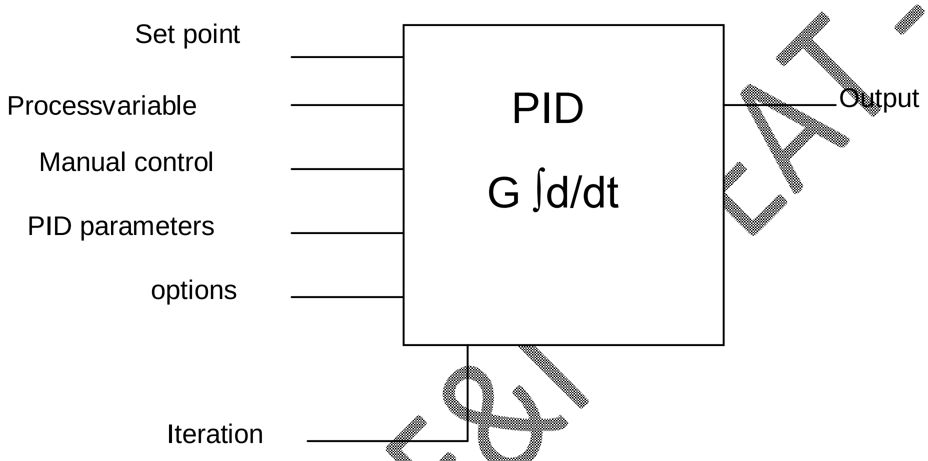


Figure 1. The PID software VI

Set point is the desired value for the process variable.

Process variable is the value of the feedback control loop.

Manual control is relative controller output value.

PID parameters are a cluster of Kc, Ti and Td where Kc is proportional gain,Ti is integral time in minutes and Td is the derivative time in minutes.

An 'option' is a cluster of 11 elements specifying optimal parameters for PID algorithm.

1. SP low is the minimum value for process variable and set point.
2. SP high is the maximum value for process variable and set point.
3. OUT low is the minimum value for the controller output.
4. OUT high is the maximum value for the controller output.
5. Hold (F), when 'true' places the controller in hold mode. Reset action stops and output freezes.

6. Auto (T), if 'true' selects automatic control and places the controller in manual mode when' false'. Bumpless transfer is used from auto to manual.
7. Proportional band(F), selects whether the proportional value of the PID parameters input is proportional gain or proportional band.
8. Reverse acting(T), if 'true' selects reverse action, the usual mode for controllers.
9. Beta is relative emphasis of disturbance rejection to set point tracking.
10. Linearity sets the linearity of error response, ranging from zero to one. '1' gives a plain linear response, while '0.1' gives an approximately parabolic (square law) response.
11. dt(s) is the interval in seconds at which the VI is called.

Iteration is the control loop iteration value.

Output is the output of control algorithm.

Exercise problem:

Simulate a temperature control system given by the transfer function

$$G(s) = K\,e^{-tds} / (\tau s + 1)$$

Where K = 77, τ = 1 min and td = 0. Design a PID controller for the given process.

Procedure:

1. Construct the front panel and block diagram as shown in Figures 2 and 3.
2. Enter the PID parameters.
3. Give a step change in set point.
4. Run the VI and check the response.

Result:

Thus a temperature control system is simulated using LabVIEW software.

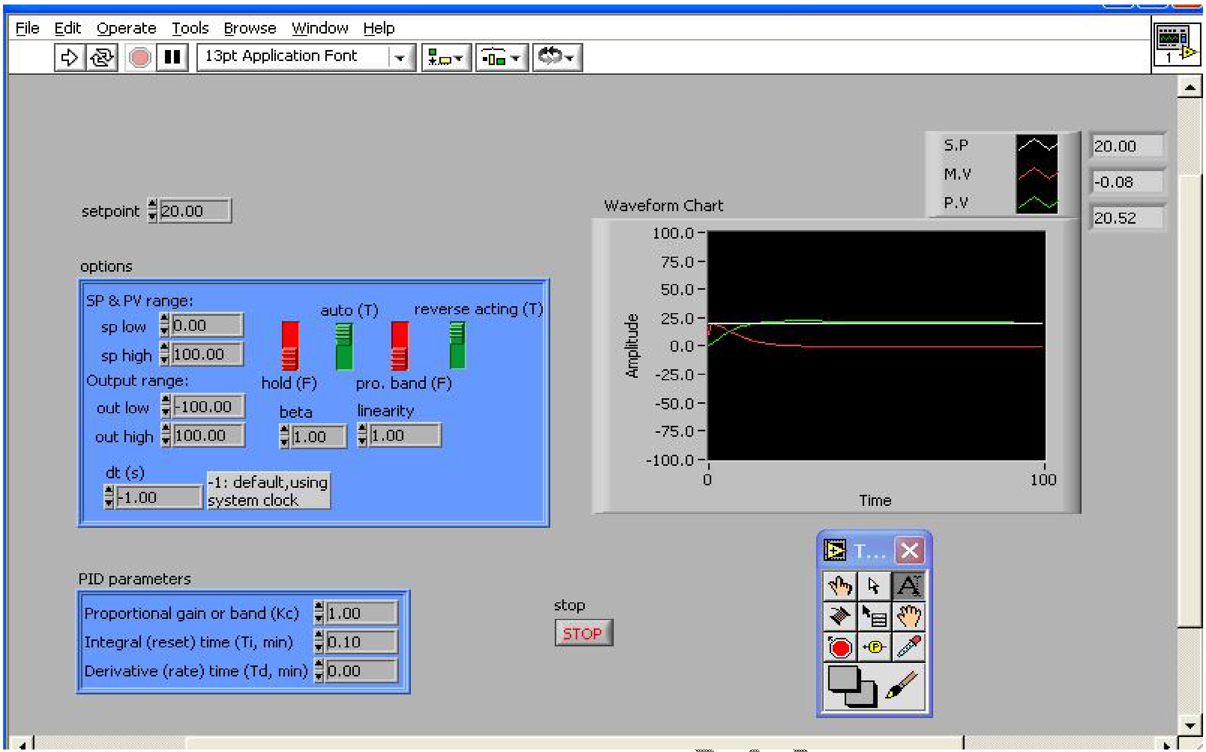


Figure 2.LabVIEW Front Panel for Temperature Process

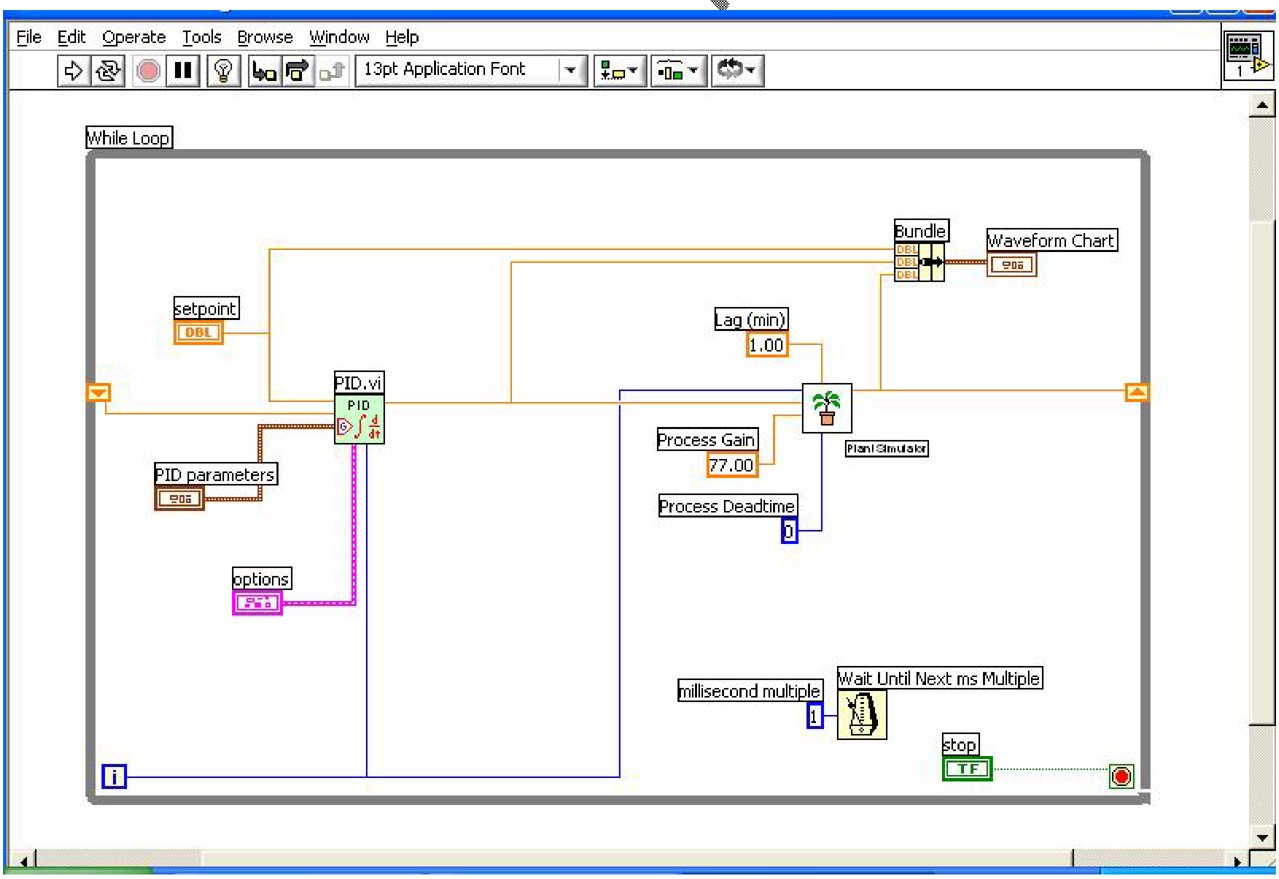


Figure 3.LabVIEW Block Diagram for Temperature Process

Expt. No. :

Date :

DESIGN OF INVERSE RESPONSE COMPENSATOR AND SIMULATION USING SIMULINK

AIM

To design and implement a inverse response compensator for a given process using simulink.

THEORY

The dynamic behavior of certain processes moves in an opposite direction initially to where it eventually ends up for a step change at its input. Such behavior is called inverse response or non minimum phase response. Table.1 shows several such opposing effects between first order (or) second order systems. In all the cases, the system possesses an inverse response as its transfer function has a positive zero. Systems with inverse response are particularly difficult to control and require special attention.

INVERSE RESPONSE COMPENSATOR

There are two very popular ways to control systems with inverse response.

1. PID Controller with Ziegler Nichol's tuning.
2. Inverse response compensator.

1. SIMPLE PID CONTROL

From all types of feedback controllers only PID can be used effectively because of the simple reason the derivative control mode by its nature will anticipate the "wrong" direction of the system's response and will provide the proper corrective action to limit (never eliminate) the inverse response.

INVERSE RESPONSE COMPENSATOR:

A Smith Predictor (dead time compensator) cancels the effect of dead time. The same general concept of the predictor (compensator) can be used to cope with the inverse response of a process. To eliminate the inverse response it is enough to eliminate the positive zero of the above open-loop transfer function. Figure 1 shows conventional feed back control of process with dead time.

DESIGN AN INVERSE RESPONSE FROM TWO FIRST ORDER SYSTEMS

$$G(s) = \left(\frac{k_1}{\tau_1 s + 1} - \frac{k_2}{\tau_2 s + 1} \right)$$

Figure 2 two opposing first order systems and Figure 3 gives the individual response and overall response of the first order systems.

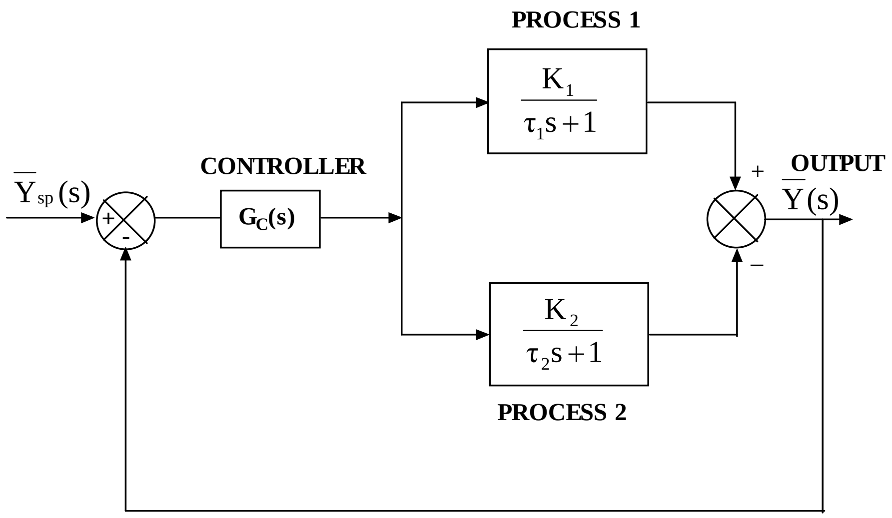


FIGURE 1. CONVENTIONAL FEEDBACK CONTROLLER OF PROCESS WITH INVERSE RESPONSE

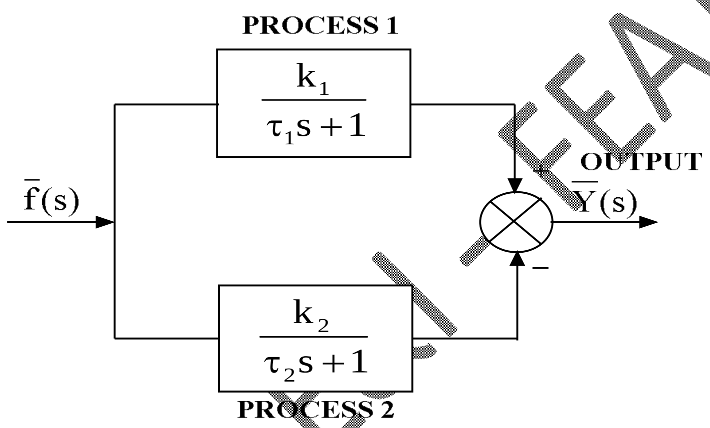


FIGURE .2 BLOCK DIAGRAM OF TWO OPPOSING FIRST - ORDER SYSTEMS

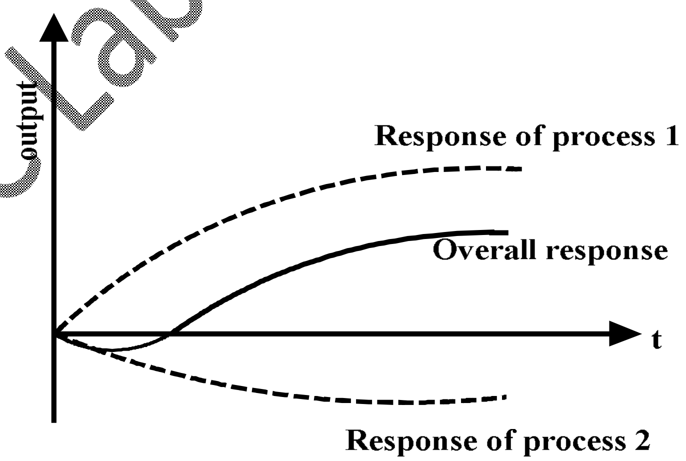


Figure.3 Response of process 1, process 2 and overall response

TABLE.1 SYTEMS WITH INVERSE RESPONSE

1.Pure capacitive minus first-order response $G(s) = \frac{k_2}{s} - \frac{k_1}{\tau_1 s + 1} = \left(\frac{[k_2\tau_1 - k_1]s + k_2}{s(\tau_1 s + 1)}\right)$ For $k_2\tau_1 < k_1$ $\quad Z = -\frac{k_2}{(k_2\tau_1 - k_1)} > 0$
2. Difference between two first-order responses $G(s) = \frac{k_1}{\tau_1 s + 1} - \frac{k_2}{\tau_2 s + 1} = \left(\frac{[k_1\tau_2 - k_2\tau_1]s + [k_1 - k_2]}{(\tau_1 s + 1)(\tau_2 s + 1)}\right)$ For $\frac{\tau_1}{\tau_2} > \frac{k_1}{k_2} > 1$ $\quad Z = -\frac{(k_1 - k_2)}{(k_1\tau_2 - k_2\tau_1)} > 0$
3. Difference between two first-order responses with dead time $G(s) = \frac{k_1 e^{-t_1 s}}{\tau_1 s + 1} - \frac{k_2 e^{-t_2 s}}{\tau_2 s + 1}$ For $k_1 > k_2$ and $t_1 > t_2 \geq 0$
4. Second order-minus first order responses $G(s) = \frac{k_1}{\tau^2 s^2 + 2\delta\tau s + 1} - \frac{k_2}{\tau_2 s + 1}$ For $k_1 > k_2$
5. Difference between two second-order responses $G(s) = \frac{k_1}{\tau_1^2 s^2 + 2\delta_1\tau_1 s + 1} - \frac{k_2}{\tau_2^2 s^2 + 2\delta_2\tau_2 s + 1}$ For $\frac{\tau_1^2}{\tau_2^2} > \frac{k_1}{k_2} > 1$
6. Differences between two second-order responses with dead time $G(s) = \frac{k_1 e^{-t_1 s}}{\tau_1^2 s^2 + 2\delta_1\tau_1 s + 1} - \frac{k_2 e^{-t_2 s}}{\tau_2^2 s^2 + 2\delta_2\tau_2 s + 1}$ For $k_1 > k_2$ and $t_1 > t_2 \geq 0$

The transfer function of the overall system is

$$\overline{Y}(s) = \left(\frac{k_1}{\tau_1 s+1} - \frac{k_2}{\tau_2 s+1}\right)\overline{f}(s)$$

$$\overline{Y}(s) = \left(\frac{[k_1\tau_2 - k_2\tau_1]s + [k_1 - k_2]}{(\tau_1 s+1)\,(\tau_2 s+1)}\right)\overline{f}(s)$$

We have inverse response when initially (at t =0+) process 2 which reacts faster than process 1 (i.e $\frac{k_1}{\tau_2} > \frac{k_1}{\tau_1}$) dominates the response of the overall system, but ultimately process 1 reaches a higher steady-state value than process 2 (i.e $k_1 < k_2$) and forces the response of the overall system in the opposite direction. Figure 3 shows the inverse response of the overall system.

Consider the feedback system of Figure 3. The controlled process exhibits inverse response

when, $$\frac{\tau_1}{\tau_2} > \frac{k_1}{k_2} > 1$$

The open-loop response of the system is

$$\overline{Y}(s) = G_c(s)\left(\frac{[k_1\tau_2 - k_2\tau_1]s + [k_1 - k_2]}{(\tau_1 s+1)\,(\tau_2 s+1)}\right)\overline{Y_{sp}}(s)$$

It has a positive zero at the point,

$$Z = -\frac{(k_1 - k_2)}{(k_1\tau_2 - k_2\tau_1)} > 0$$

To eliminate the inverse response it is enough to eliminate the positive zero of the above open-loop transfer function. This is possible if in the open loop response $\overline{Y}(s)$ we add the quantity $\overline{Y}^{'}(s)$ given by,

$$\overline{Y}^{'}(s) = G_c(s)\,k\left(\frac{1}{\tau_1 s+1} - \frac{1}{\tau_2 s+1}\right)\overline{Y_{sp}}(s)$$

$$\overline{Y}^{*}(s) = \overline{Y}(s) + \overline{Y}^{'}(s)$$

$$\overline{Y}^{*}(s) = G_c(s)\left(\frac{[k_1\tau_2 - k_2\tau_1] + K(\tau_1 - \tau_2)s + [k_1 - k_2]}{(\tau_1 s+1)\,(\tau_2 s+1)}\right)\overline{Y_{sp}}(s)$$

$$K \geq \frac{(k_2\tau_1 - k_1\tau_2)}{(\tau_2 - \tau_1)}$$

We find that the zero of the resulting open loop transfer function is non positive.

$$z = -\frac{k_1 - k_2}{(k_1\tau_2 - k_2\tau_1) + K(\tau_1 - \tau_2)} \leq 0$$

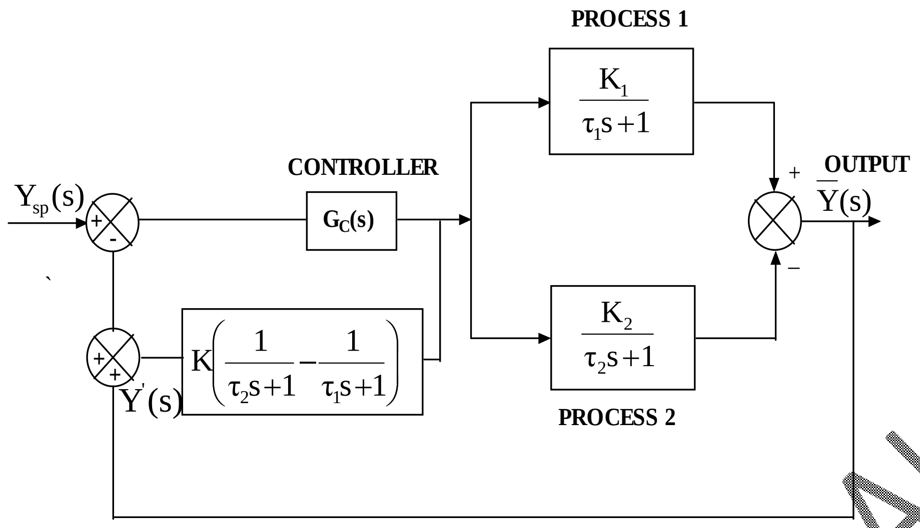


FIGURE 4. FEEDBACK CONTROL OF PROCESS WITH INVERSE RESPONSE WITH COMPENSATOR

Adding the signal $\overline{Y}'(s)$ to the main feedback signal $\overline{Y}(s)$ means the creation of the local loop around the controller as shown in Figure 4. The system in this local loop is the modified Smith Predictor and the actual compensator of the inverse response.

$$G_{compensator} = K\left(\frac{1}{\tau_2 s+1} - \frac{1}{\tau_1 s+1}\right)$$

PROBLEM

Design an inverse response for the given transfer function

$$G(s) = \frac{12}{2s+1} - \frac{9}{s+1}$$

PROCEDURE

1. Give the step input for the higher order process. Vary the gain until the sustained oscillation is produced as shown in Figure 5.
2. From the process response find the ultimate period pu and ultimate gain ku.
3. Find the PID controller settings from (Z-N closed loop tuning rules)

Kc = 0.6 * ku;

$$T_i = \frac{p_u}{2}$$

$$T_d = \frac{p_u}{8}$$

CALCULATION

PID settings:

Ku = 0.5;
Pu = 5.73
Kc = 0.6 * ku
Kc = 0.6 * 0.5 = 0.3
T_i = pu/2
T_i = 5.73/2 = 2.865
T_d =pu/8
T_d =5.73/8 = 0.71625

FIGURE 1. BLOCK DIAGRAM OF CONVENTIONAL FEEDBACK CONTROL OF PROCESS WITH INVERSE RESPONSE

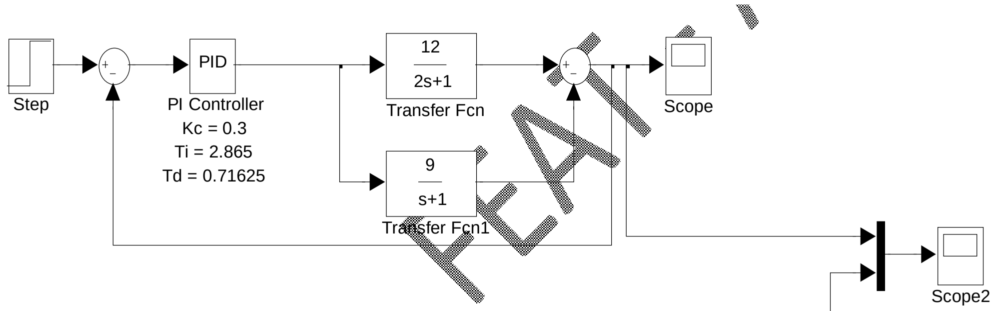


FIGURE 5. BLOCK DIAGRAM OF FEED BACK CONTROL OF PROCESS WITH INVERSE RESPONSE COMPENSATOR

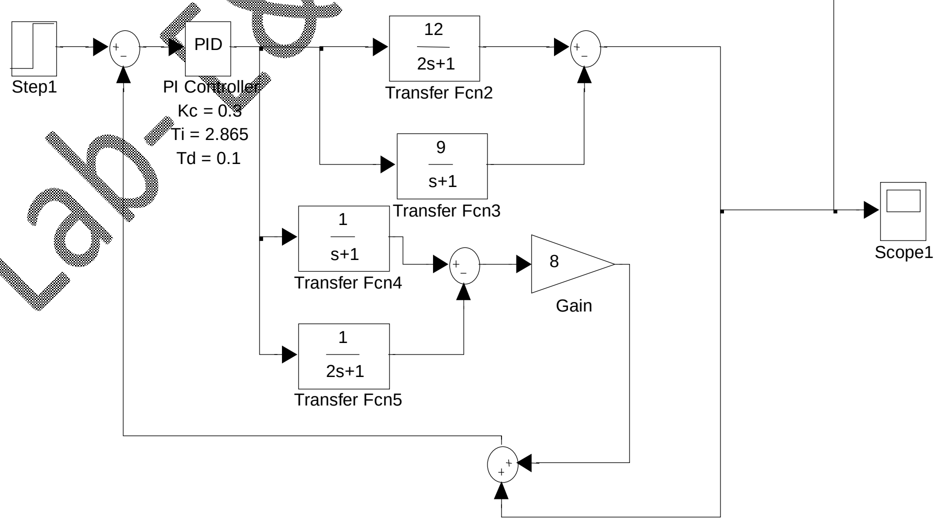
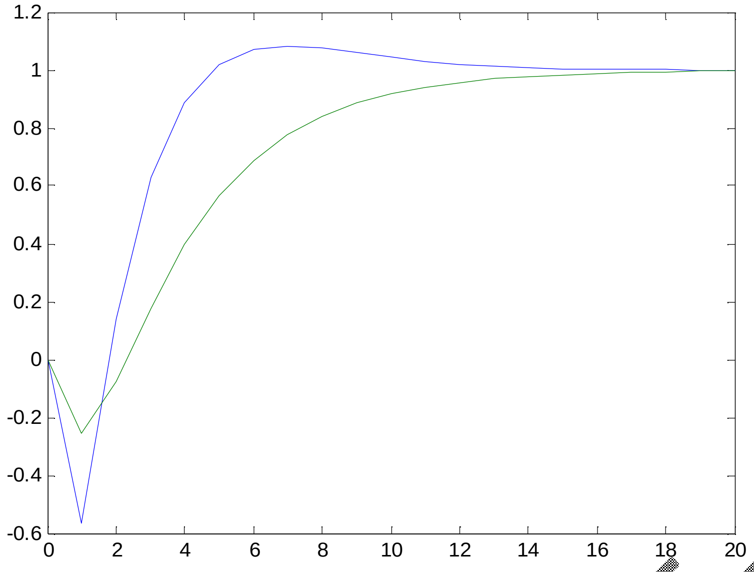


FIGURE.6 PROCESS WITH INVERSE RESPONSE /WITH INVERSE COMPENSATOR



RESULT

The inverse response compensator of a given transfer function has been obtained.

Expt. No. :

Date :

(b) STUDY OF BIO SIGNALS

Aim:

To obtain and analyze the bio signals such as ECG, Pulse Rate from an EPR system.

Equipments needed:

EPR system, Digital Storage Oscilloscope (DSO)& appropriate electrodes& sensors.

Theory:

An EPR system is for monitoring three parameters namely ECG, Pulse and Respiration rate.

ECG System:

The Electrocardiography deals with the study of the electrical activity of the heart muscles. The potentials originated in the individual fibres of the heart muscles are added to produce the ECG waveform. Electrocardiogram is the recorded ECG wave pattern. Electrocardiogram reflects the electrical depolarisation and repolarisation of the myocardium (heart muscles) associated with the contractions of the atria and ventricles. The shape, time interval and amplitude of the ECG detail the state of the heart. Any form of arrhythmia (disturbances in the heart rhythm) can be easily diagnosed using electrocardiogram.

Peripheral Pulse System:

It is the series of waves of arterial pressure caused by left ventricular systoles as measured in the limbs. Pulse rate is the number of pulsations per minute palpable in an artery, usually of a limb. The normal rate per minute is 60-100. A peripheral pulse sensor is a particularly convenient non-invasive method of measuring the peripheral pulse. Typically, the sensor has a pair of small Light Emitting Diode (LED) and Light Dependent Resistor (LDR) placed over a translucent part of the subject's body usually a fingertip or an earlobe.

Procedure:

1. Switch on the system.
2. Connect the output of ring electrodes to the system.
3. Monitor the ECG waveform on the DSO through the corresponding amplifier output.
4. Connect the output of peripheral pulse sensor to the pulse system.
5. Monitor the pulse waveform on the DSO through the corresponding amplifier output.

Analyse the ECG waveform obtained with the pattern given and infer your results. Calculate the heart rate in beats per minute.

From the pulse waveform calculate the time difference between any two peaks and calculate the pulse rate.

Result:

The ECG and Pulse waveforms are obtained from an EPR system.

Elements of ECG:

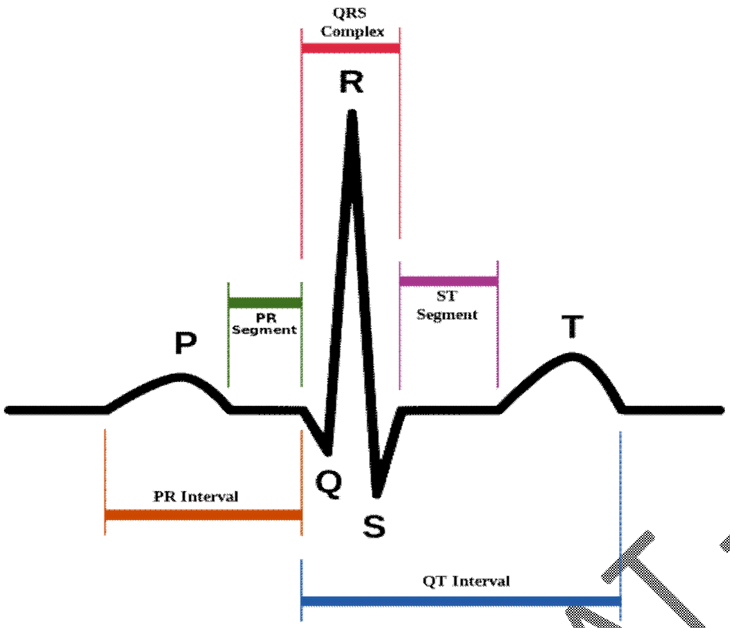


Figure 1. ECG waveform

Waves	Amplitude (mV)	Duration (Sec)	Origin
P wave	0.25	0.12 to 0.22 (P-R Interval)	Atrial Depolarisation
R wave	1.6	0.07 to 0.1	Repolarisation of atria and depolarisation of ventricles
T wave	0.1 to 0.5	0.05 to 0.15 (S-T Interval)	Ventricular repolarisation (Relaxation of myocardium)
U wave	<0.1	0.2 (T-U Interval)	Slow repolarisation of Intra ventricular (Purkinje Fibres) System
S-T Interval			Ventricular contraction

Expt. No. :

Date :

STUDY OF PLC (GE FANUC make)

AIM

To study the functions of GE Fanuc PLC and control of Bottle filling system using PLC.

APPARATUS REQUIRED

1. PLC & VERSAPRO SOFTWARE
2. Bottle Filling System Controller Kit
3. PC
4. RS-232 cable
5. Patch-Chords

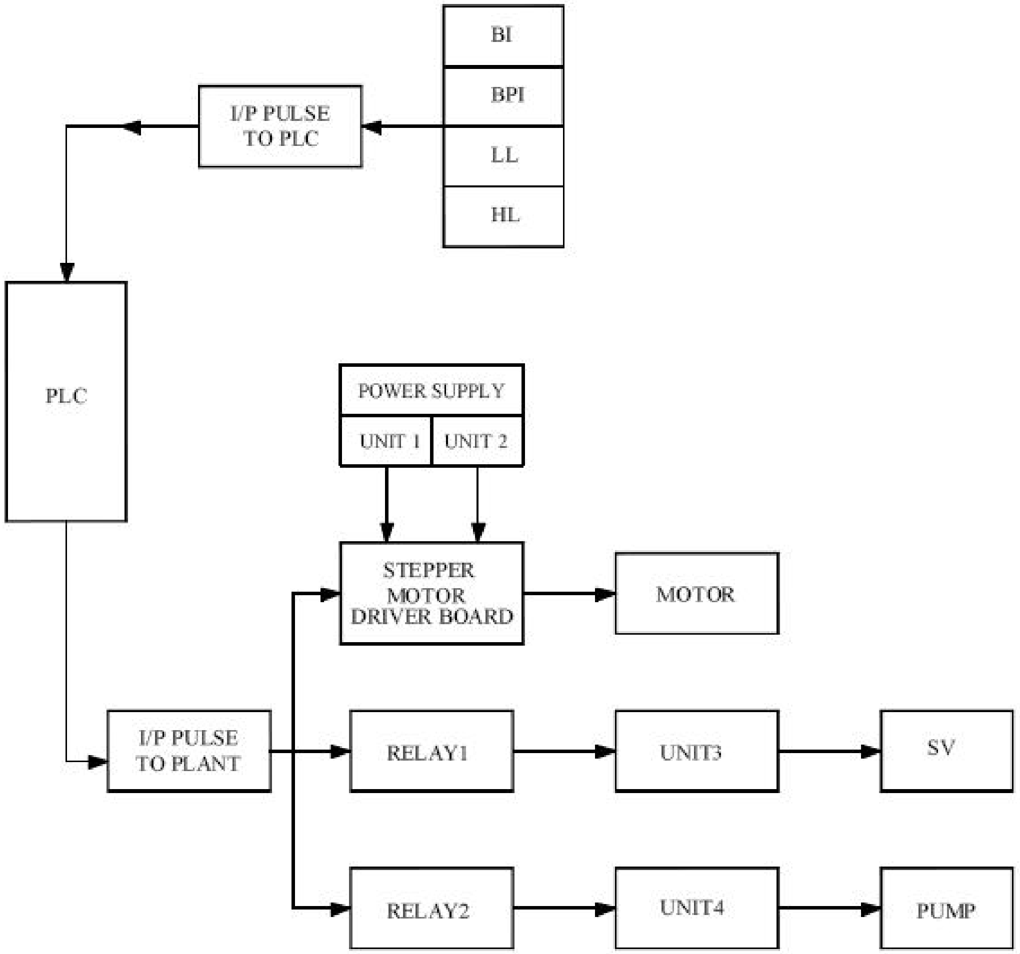
BOTTLE FILLING SYSTEM

The bottle filling system is used to control the liquid level in bottle & moved to next bottle, then maintain the liquid level in process tank, solenoid valve operations is maintained automatically & continuously. In this processing system, Low-level & High level sensors are used to sense the liquid level in process tank. Then solenoid valve will open & stepper motor will stop the rotation whenever Bottle sensor & Bottle position sensor will get the output value. After average value of time fill the liquid in Bottle, automatically solenoid valve will close & stepper motor rotates now. This process repeated again and again.

WORKING OF THE SYSTEM

First of all, in the process tank is in low level switch will get control. It causes for switch ON the motor, then fill the liquid to process tank, liquid level increases continuously. The motor will goes to OFF state, whenever the liquid level will reach the high level. Stepper motor is rotating now. It will be stop depends upon two factors such as result of bottle sensor [Proximity Inductive Sensor] result of bottle position sensor [Opto-Coupler]. If the two sensors senses the bottle, when the stepper motor is in stop mode, then fill the liquid. Simultaneously, solenoid valve will open which is depends upon the output value of two sensors. We will set into particular time-delay in PLC, after the time delay, solenoid valve will move into close position, which is depends upon the average value of time to fill the bottle. Simultaneously, stepper motor starts now then rotate the platform, whenever the two sensor will sense then stop the stepper motor and solenoid valve will open. This process is going on continuously upto reach the low level of liquid in process tank then starts ON the motor.

BOTTLE FILLING SYSTEM BLOCK DIAGRAM



UNIT	V/I RANGE	PURPOSE
1	6V 6AMS	STEPPER MOTOR SUPPLY
2	15V 1AMS	MOSFET OPERATING VOLTAGE
3	230V 5AMS	SOLENOID VALVE OPERATING VOLTAGE
4	12V 4AMS	PUMP OPERATING VOLTAGE

HARDWARE USED

FLOAT SWITCH

It is a sensor, which is used to detect the water level. It is a magnetic type sensor water level is below the sensor means the sensor output is low. The sensor output is high when the water level is equal (or) above the sensor. We are using two float switches, first one is used to detect the low -level & another one is used to detect the high-level of liquid in process tank.

STEPPER MOTOR

It produces the liquid rotating motion to bottle filling system, Stepper motor having 4 coils, which can be excited with DC current to run the motor in clock-wise direction. By reverse direction, we will change the phase sequences coil-1, coil-4, coil-3 & coil-2. Four coil of stepper motor are arranged into rotating for step-wise movement.

$$\text{Step angle} = 360 / N_s \times N_R$$

PROXIMITY INDUCTIVE SENSOR

This sensor connected vertically detects the presence of bottle on the platform. It monitoring the proper movement of bottles on the platform and indicates the during failures. This sensors also eliminated the air gap between the bottle and sensor.

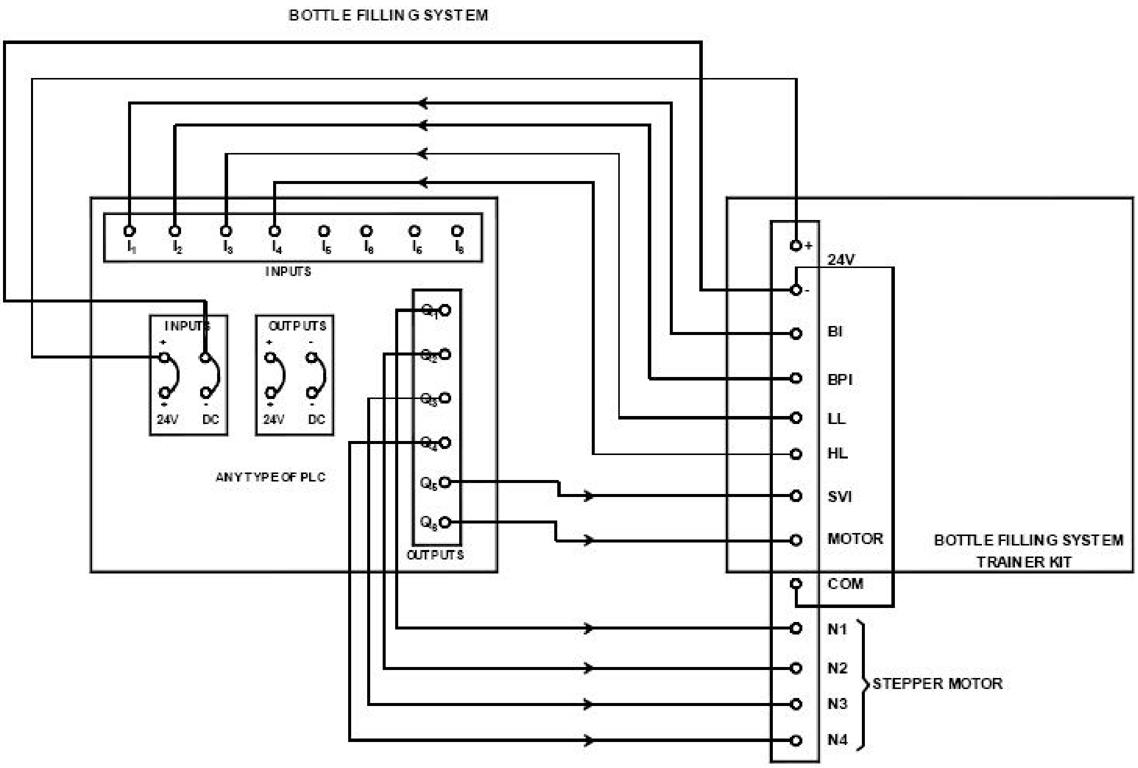
POSITION SENSOR

It is nothing but, opto-coupler sensor is a Bottle position-sensor. It senses the exact position of bottle-mouth, which can be cut the gap of opto-coupler sensors. Then bottles have to be filled up which are programmed for a certain delay. The delay between the platform rotation and solenoid valve excitation is held simultaneously whose failure is detected using the position sensor.

SOLENOID VALVE

It is positioned above of the bottle mouth. Filling process of liquid in bottle is carried out through solenoid valve. Solenoid valve is excited with a supply of 230V. On excitation, liquid passes through the valve for the time programmed and till up the bottle. After tilling of one bottle is finished solenoid valve is de-excited to stop the flow of liquid. A device is introduced in between one DC excitation and the next excitation is after positioning the bottle.

WIRING DIAGRAM



LADDER LOGIC PROGRAM EXPLANATION

In automatic bottle filling system, in rung-5 we will connect the program then automatically start the motor because %I0003 & I0004 are normally closed contact. It is in high level of liquid in process tank, motor will stop, because the will get open contacts. If the Bottle sensor & Bottle position sensor's output are in high-state, when %M0001 [Open coil] are energized in rung-1. In rung-2, %Q0005 [Open coil Q5] energized now, because of %M0003 [M3-Memory] is closed contact. That means, output Q5 coil is solenoid value, hence solenoid value opening now.

In rung-3, decided the time-delay of solenoid valve open/close position. FOR ON DELAYTIMER in rung-3, after reach the preset value or time-delay %M0002 coil is energized now, it causes for again reset the timer value. Simultaneously, %M0003 [M3-coil] is energized now in rung-4. Because of get contact %M0001 & %M0002, then the coil 3 energized continuously. Causes for solenoid valve will get OFF state, because %M003 will more open position & Q0005 will more to de-energization.

In rung-6 designed for stepper motor rotation, when ever %Q0006 [MOTOR] & %Q0005 [Solenoid valve] will move in OFF state. In rung-8, automatically timer will function, opto the preset value time-delay. Because ON-DELAY TIMER is connected on directly in rung-8. %M0004 [M4-Contact] opencontact will move into close contact, due to %M0004 [M4-coil] energization in rung-8. In rung-7 up counter is functioning now, counter will incremented by one, whenever the ON delay timer will run reset. Since we have four coils, so we store the preset value to our. After completion of 4 counts in up-counter is resetting now, the input or SNX is given through up counter register [%R00030]. SNX is to incremented the starting register [SR] in array more function. The register data will be more to P1 and incremented till Q4 [Fourth coil], that means four coils data is moving to counter, hence the stepper motor starts rotating. The rotation of stepper motor stops, when Q6 is low as I1 & I2 are high. This operation is repeating in these sequence.

RESULT

Thus functions of GE Fanuc PLC are studied and bottle filling system is controlled.

Expt. No. :

Date :

PC BASED CONTROL OF AIR FLOW/PRESSURE USING V-MAT DATA ACQUISITION CARD

Aim

(i) To study and interface PCS with personal computer.

(ii) To obtain the closed loop response of a first order system (or for a given process) using PID algorithm

APPARATUS REQUIRED:

Process control simulator, VMAT - 01 [ADC/DAC – ADD ON CARD], PC[PC-XT], patch chords.

The ADC allows the digital system to take in information from the analog system. [The o/p of the process]. The digital system [PC] can now rapidly analyze and process this information.

The DAC allows the result of such analysis to be communicated back to the analog system (process control simulator)

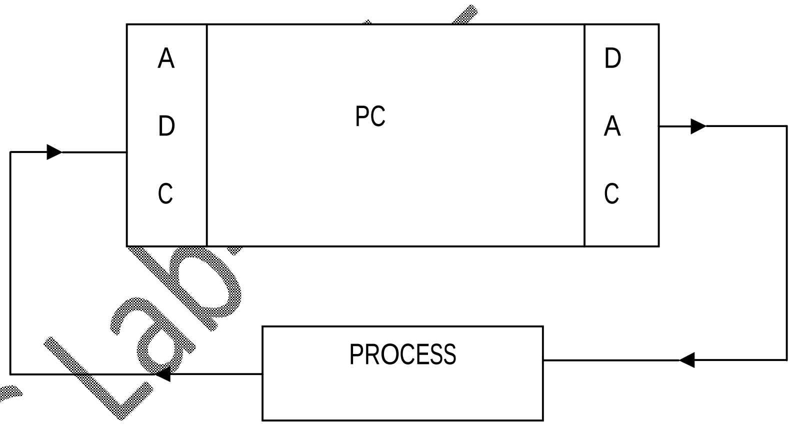


Figure 1 General Block Diagram

INTRODUCTION TO VMAT-01

In every educational field most of them are familiar with matlab because it is the environmentto develop any algorithm easily. Our real time MATLAB interface card(VMAT - 01) that allowsyou to interface with MATLAB for controlling real time applications. We can implement ourrequired closed loop algorithm by using MATLAB simulink tools like PI, PID, FUZZY logic,etc.,

The VMAT - 01 Controller is specially designed for Real time interface for MATLAB simulink,Orcad simulation software. Based on advanced Risc Micro controller, a useful tool for researchers to interface their hardware with MATLAB to research in the fields like processcontrol loop, DC - DC converter, DC motor controller, control Engineering etc.,If we want to design this type of closed loop algorithms in any digital controller or some othercontrollers it is so difficult than MATLAB. Using our real time interface card (VMAT - 01) wecan gather real time signals and define our control technique in MATLAB then transfer thecontrol signals to real time application.

FEATURES

* dSpic- Micro controller Based Controller.
* Can be used for instrumentation PID control Applications.
* Can be used for Power Electronics, DC-DC Converter, DC Motor control Applications.
* Single Channel ADC Input (0-5V).
* Single Channel DAC Output (0-5V).
* 2 Channel PWM Output (250 KHz).
* PC Interface Through RS232.
* +5 V DC Operation.

FRONT PANEL DESCRIPTION

Serial Port Interface - Interface PC and VMAT - 01 unit through RS232 cable.
5V Adaptor - To give supply to the unit.
GND - To connect the GND terminal.
ADC - To give the ADC input (0 - 5)V
DAC - To measure the DAC output (0 - 5)V
PWM1 - To measure the PWM1 signal.
PWM2 - To measure the PWM2 signal.

COMMUNICATION

Using serial port we can communicate our real time interface card (VMAT - 01) with MATLAB.For transmitting and receiving the data from the mat lab we provide some libraries. For receivingdata from real time application use the **vi read** block. To transmit the control data to VMAT -01 card use the **vi write** block.

VI_READ

The ADC input of the VMAT-01 real time interface card can be gather into MATLAB as a output of VI_READ . The value of ADC input range 0-5V is proportional to the vi_read output integer value of (0-255). For example when we give 5V as a ADC input then the output of vi_read will be 255. By using this proportionality we can configure the real time data in simulink environment.

VI_WRITE

Above figure of vi_write is used to transfer the data to our VMAT-01 card. The main thing is we need to define our switching frequency(switching Time) of the pulse width modulation(PWM) signal. **DAC** terminal is used to control the output voltage of digital to analog converter from VMAT - 01 card.

255 data = 5V

For 1v output

= 255/5.

= 51.

For getting 1v output we have to give the input 4 as 51.

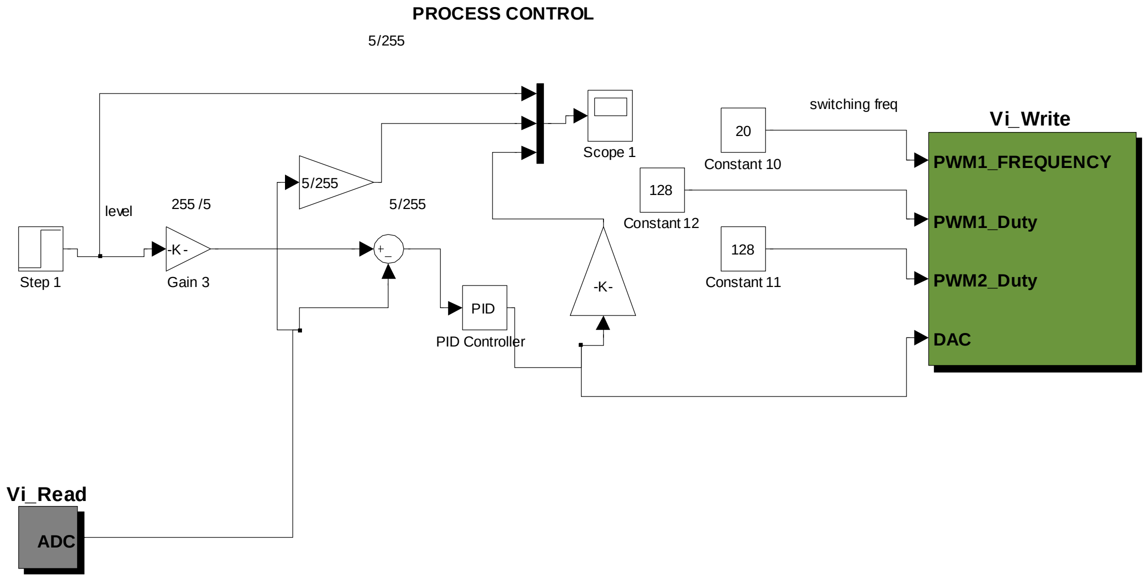


Figure 2.Simulink diagram of closed loop control

Procedure

1. ulate the controller parameter for the given process using synthesis formula Kc=τc/Kτ; Ti=τc where τc=closed loop time constant and τ=open loop time constant of process
2. Connect the RS232 serial port with the PC and VMAT-01 real time MATLAB Interface card.
3. Connect the 5V power supply to the VMAT-01 real time MATLAB interface card.
4. Give your feed back voltage to the ADC terminal. That must be in 0 - 5V range.
5. Take DAC output from the VMAT-01 card.
6. Configure the simulink diagram as shown in figure2 and execute it for various values of setpoint change and comment on the results

FRONT PANEL DIAGRAM

Figure - 1

Result

Thus a process is interfaced with PC using VMAT -01 and closed loop control is achieved.

STUDY OF DCS

Aim: To study the features of Yokogawa DCS

To control the level process using DCS

Distributed Control System

A distributed control system (DCS) refers to a control system usually of a manufacturing system, process or any kind of dynamic system, in which the controller elements are not central in location (like the brain) but are distributed throughout the system with each component sub-system controlled by one or more controllers. The entire system of controllers is connected by networks for communication and monitoring. DCS is a very broad term used in a variety of industries, to monitor and control distributed equipments.

A DCS typically uses custom designed processors as controllers and uses both proprietary interconnections and communications protocol for communication. Input and output modules form component parts of the DCS. The processor receives information from input modules and sends information to output modules. The input modules receive information from input instruments in the process (or field) and transmit instructions to the output instruments in the field. Computer buses or electrical buses connect the processor and modules through multiplexer or demultiplexers. Buses also connect the distributed controllers with the central controller and finally to the Human–machine interface (HMI) or control consoles.

BASIC COMPONENTS OF DCS

FCS (Field control station):

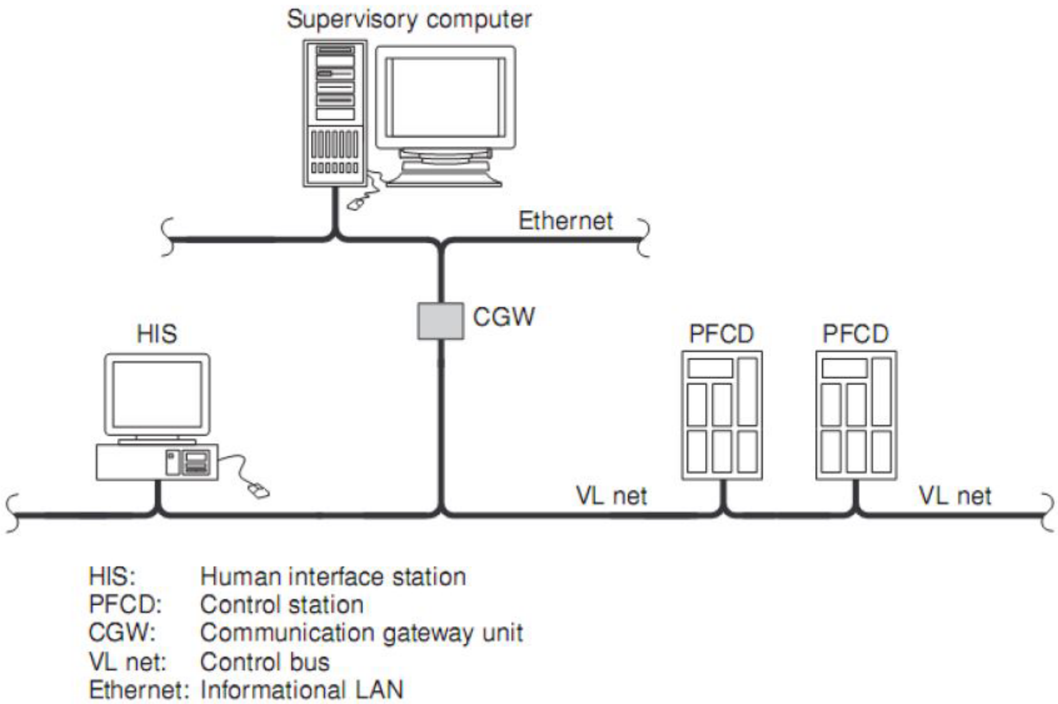
It is used to control the process. All the instruments and interlocks created by the software reside in the memory of the FCS. All the field instruments like transmitters and control valves are wired to the FCS.

OPS(Operator Station):

Used to monitor the process and operate various instruments.

Communication Bus:

Used to communicate between the FCS and the OPS



CENTUM is the generic name of Yokogawa's distributed control systems for small- and medium-scale plants (CENTUM CS 1000), and for large scale plants (CENTUM CS 3000)
The system has one domain and 64 stations. So FCS is FCS0101 (meaning first domain, first station) and HIS0164 (meaning first domain 64 th station).

Features of CS3000

- CENTUM CS 3000 - Major Components
  - FCS (Field Control Station)
    - Reliable controller.
    - Cost-effective and capable I/O subsystem.
  - HIS (Human Interface Station)
    - The operator station based on Windows XP or Windows2000. (Both are selectable.)
    - HIS provides easy & flexible operation.
  - ENG (Engineering Station)
    - Engineering Station is used to do the engineering builder for all the stations like HIS, FCS, CGW, BCV etc. ENG is a PC loaded with Engineering software.
    - The HIS can be loaded with engineering software so that it can be used as HIS as well as ENG.
  - CGW: Communication Gateway Unit used to communicate with supervisory computers.
  - BCV: Bus Converter is used to link two domains.

- CENTUM CS 3000 - Networks
    - V-Net (Communication Bus)
        - Real-time control bus.
        - V-NET is a used for communication between HIS, FCS, BCV & CGW.
        - Maximum 64 Stations can be connected on the V-net.
    - ETHERNET (Communication Bus)
        - Ethernet is a standard network in CS3000 to connect HIS, ENG and supervisory computers.
        - Transmission speed: 10 MBPS
- FIO
    - FIO means Field network I/O.
    - FIO is Process I/O modules.
    - A kind of compact, cost-effective, reliable I/O devices, targeted as the industrial standard I/O of next-generation.
    - FIO includes the latest network technologies and field experience.
- FIO System Specification for FFCS (Compact Field Control Station)
    - The bus among FFCS and local nodes (ESB bus)
        - Dedicated Internal Bus
        - Speed : 128 Mbps
        - Distance in Total: Max. 10m
    - Remote I/O bus (ER bus)
        - Based on Ethernet
        - Speed : 10 Mbps
        - Distance in total: 10base2 -> max. 185m / 10base5 -> max. 500m max. 2 km with repeater (Standard of Ethernet)
        - Up to 3 remote nodes can be installed on a FFCS.
- Pair & Spare CPU Concept

Input /Output Modules

- **Analog I/O Modules**
    - AAB841-S (8Ch Voltage input/8Ch Current output 1-5V input/4-20mA output MAC2 compatible)
- **Digital I/O Modules**
    - ADV151 32 Ch24VDC input, Common minus side every 16-channel
    - ADV551 32 Ch24VDC,0.1A, Common minus side every 16-channel
- **Communication Modules**
    - Serial Communication Module
        - ALR111 RS232C
            - 2 ports, 1200bps to 115.2k bps
        - ALR121 RS422/RS485
            - 2 ports, 1200bps to 115.2k bps
    - Ethernet Communication Module
        - ALE111 Ethernet Communication
        - Installable both on Local and Remote Node

- Subsystem Packages List
    - RS Communication (ALR111/ALR121)
        - YS Communication
        - YS Directly Communication
        - FA-M3
        - Modbus
        - SLC500/PLC5
        - MELSEC
    - Ethernet Communication (ALE111)
        - FA-M3
        - Modbus
        - SLC500/PLC5
        - Control Logix
        - MELSEC
- Foundation Field bus Module (ALF111)
    - FF-H1 interface card
    - Redundancy
    - Installable both on Local and Remote nodes
    - VCR (Virtual Communications Relationship): 105 per port (one segment)
    - Both pressure clamp and terminal board are available.
    - Link Master

Human Interface station

The Station for Real time Plant Monitoring/Operation

Operation Windows

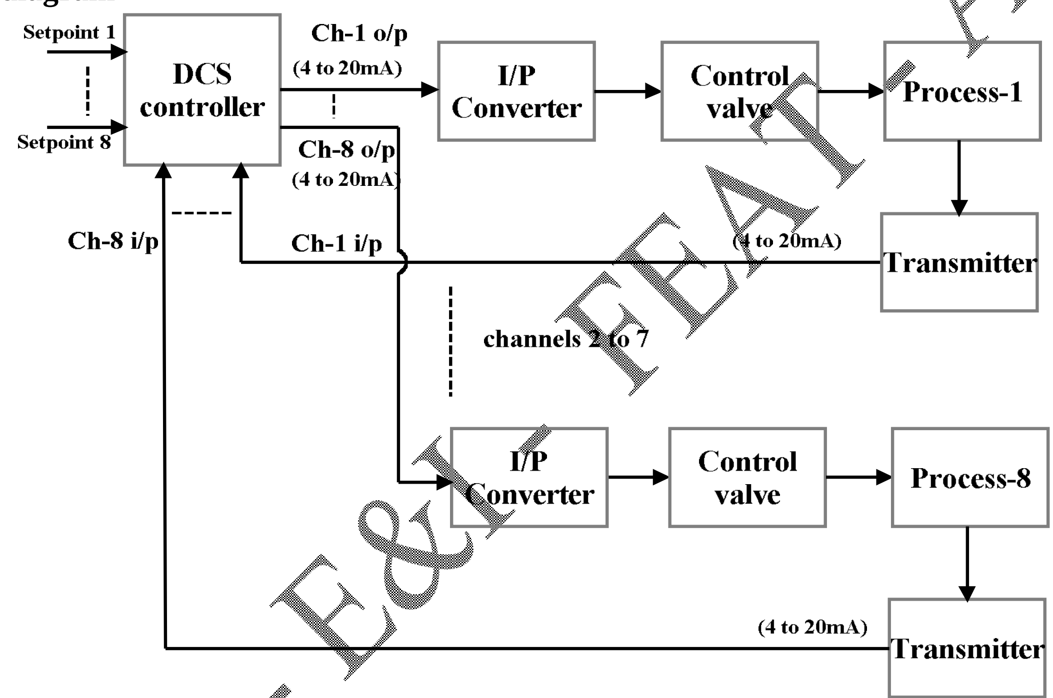
Information regarding the process is gathered as well as monitored by the following Standard Operation windows on the HIS.

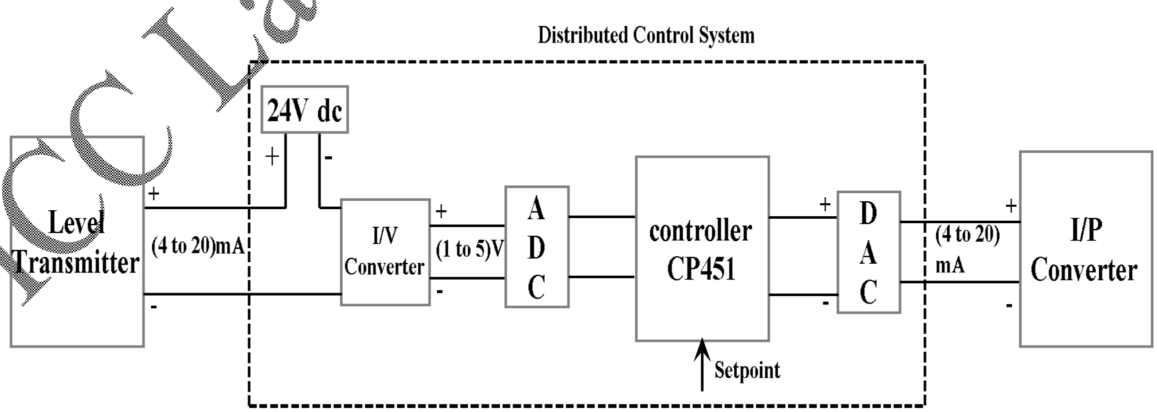
- Tuning Window
- Control Group Window
- Trend Window
- Process Alarm Window
- Operator guide Message Window
- Graphic Window
- Overview Window
- Process Report Window
- Historical Report Window

- Overview Window
    - Overview Window displays the overview of the current process status.
    - Information regarding the process is distributed among the various display blocks.
    - 32 Display Blocks per Overview Window.
    - Each block gives dynamic information regarding the process.
    - Double click on the display block for more details.
    - 3 Types of Display Blocks
    - Single Tag Block
    - Window Display Block
    - Comment Block

- Process Report Window
- Historical Message Report Window
- Sequence Tables
- Logic Charts
- System Status Window
- System Alarm Window
- SYSTEM ALARM WINDOW displays the latest 200 system alarms. Alarms can be acknowledged either as a Group or as Individual alarm.
- Navigator Window

Block diagram





Procedure

1. Power on the system
2. Goto AUEIE login
3. Goto Start – Programs - Yokogawa Centum - System view [system view screen will open]

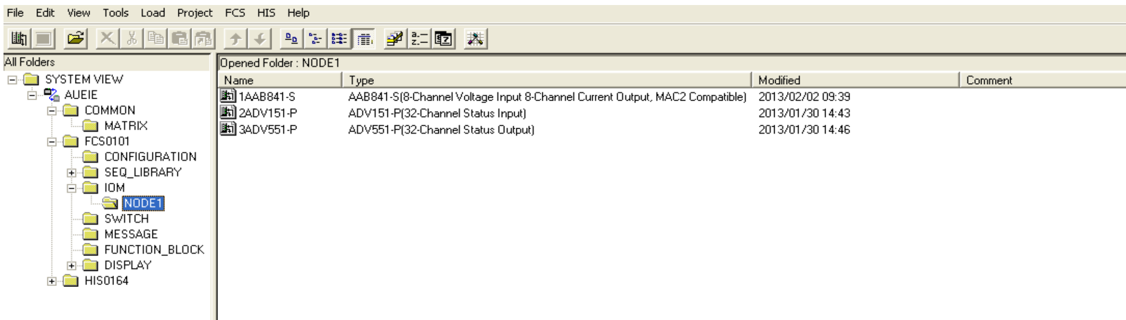


Fig.1

Step1: IOM builder

Goto AUEIE - FCS0101 – IOM - node 1

1. Click on node 1.
2. The analog I/O and digital I/O modules will appear in the right side [Fig.1]

   Open AAB841-S by double clicking over it. Press **Show/hidden detailed setting item** Icon to see the **Label** column

3. Configure the required channel with nomenclature as %%IXXX for input and %%OXXX for output in the Label column [Refer Fig.2]

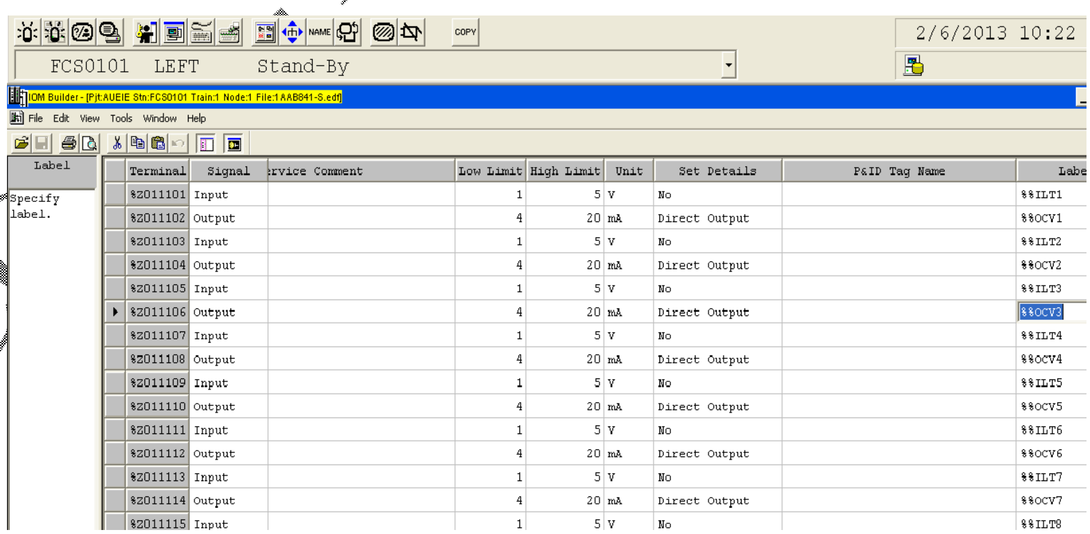


Fig.2

4. After configuration, goto File and download.
5. Give Ok in Downloading confirmation dialog, so that **Equalize completed successfully message** will appear.

Step 2: Control drawing

Goto AUEIE - FCS0101 – Function block [drawing pages will open]

Note: 200 drawing pages were there.

1. Select a page for control drawing
2. Goto Insert –Function block-Regulatory Control Blocks- Controllers-PID
3. Place the PID block in the drawing page and Give suitable Tag name e.g LIC1101
4. Again Goto Insert- Function Block- Link block-PIO
5. Place the PIO block in the drawing page and Give Tag name as configured in IOM. E.g %%ILT1 [Input from the field]
6. Place another PIO block in the drawing page and Give Tag name as configured in IOM. E.g %%OCV1 [Output to the field]
7. Choose wiring Icon [or Goto Insert- wiring] and connect as shown in Fig.3

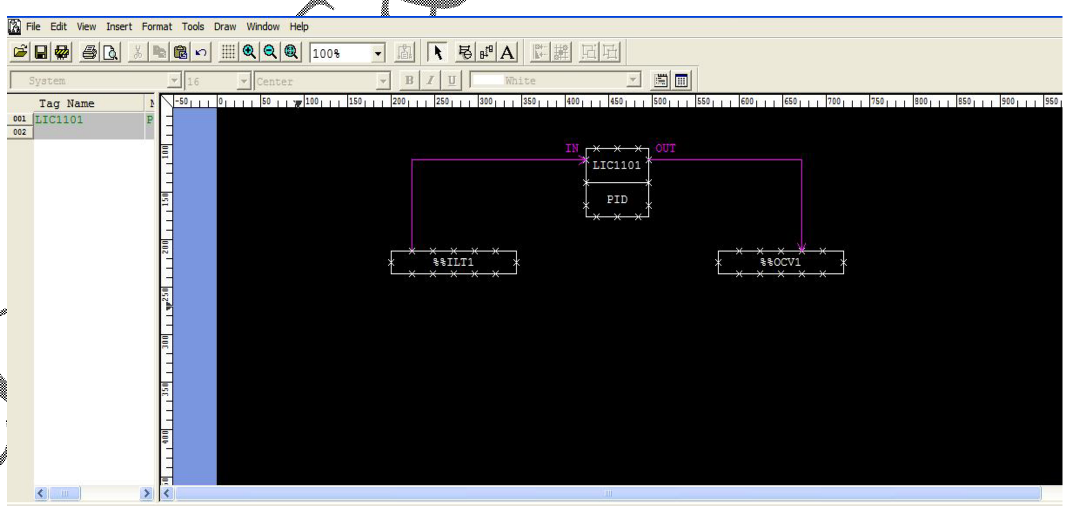


Fig.3

8. After drawing, goto File and download.
9. Give Ok in Downloading confirmation dialog, so that **Equalize completed successfully message** will appear.

Step 3: PID tuning:

In the control drawing page, goto **Name** Icon and give the Tag name of the controller. E.g LIC1101 in **Input Window name.**

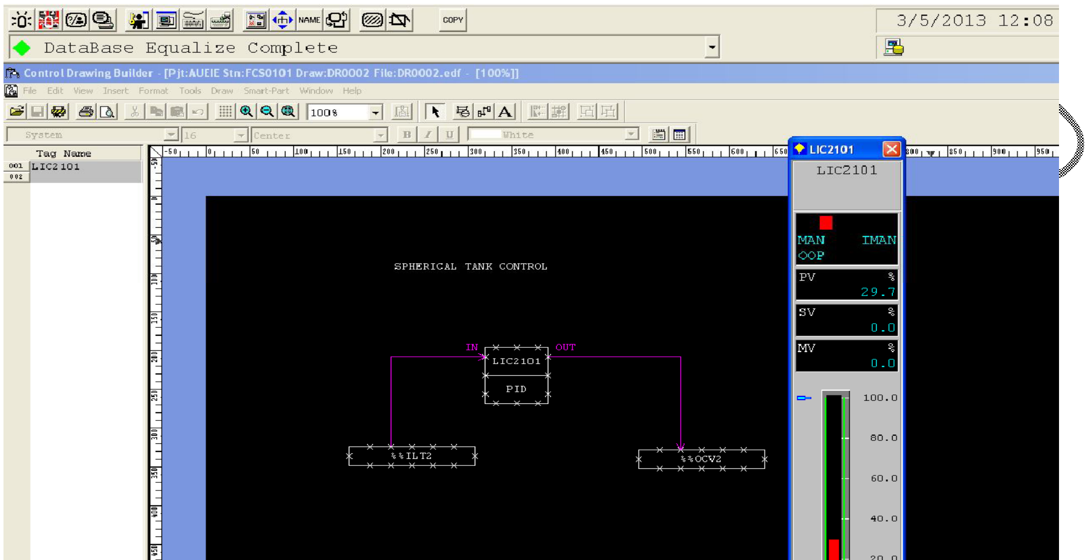


Fig.4

The face plate of PID will appear [Refer Fig.4]. In the face plate, to change the displayed parameter value, click over it. The data entry box will appear. Type the new value and press enter key. The new data value will be displayed.

Block Mode Change operation:

Clicking the block mode display area with mouse will display the block mode change operation dialog box to change the block mode. There are three basic block modes for a PID controller, namely

MAN: Manual mode
AUT: Automatic mode
CAS : cascade mode.

The block mode can be changed by operating the dialog box. If security level doesn't permit to change the mode, Goto **User-In** dialog box and switch to engineering user [**ENGUSER**] to change the PID parameters, alarms etc.

Step 4: Trend Window: To see the trend

Trend recording function of HIS acquires data from the field control station and displays changes in the acquired data in a graphical format of parameter versus time.

Trend Structure: Trend recording function has a three layered structure as **Trend Block, Trend group/window** and **Trend point/pen window**. There are 50 Trend Blocks [Note: 8 Blocks allotted for Real Time Trend], each Block has 16 Trend Groups and each group can display 8 data graphically.

To assign the variables for **Trend point/pen window**, Goto HIS0164 and click configuration. The trend Blocks appear in the right as TR0001, TR0002 like wise. Suppose we are selecting **TR0001** [i.e., first block] and Group 1 in that for assigning pen variables, to see the Trend, call the **Name** Icon and type **TG0101**, meaning Trend Group referring first block-**01** and first group-**01**. The variables are assigned as Tagname.variable name [For e.g., LIC1101.PV for process variable display of PID controller with Tagname LIC1101] {Refer Fig.5 & Fig.6}

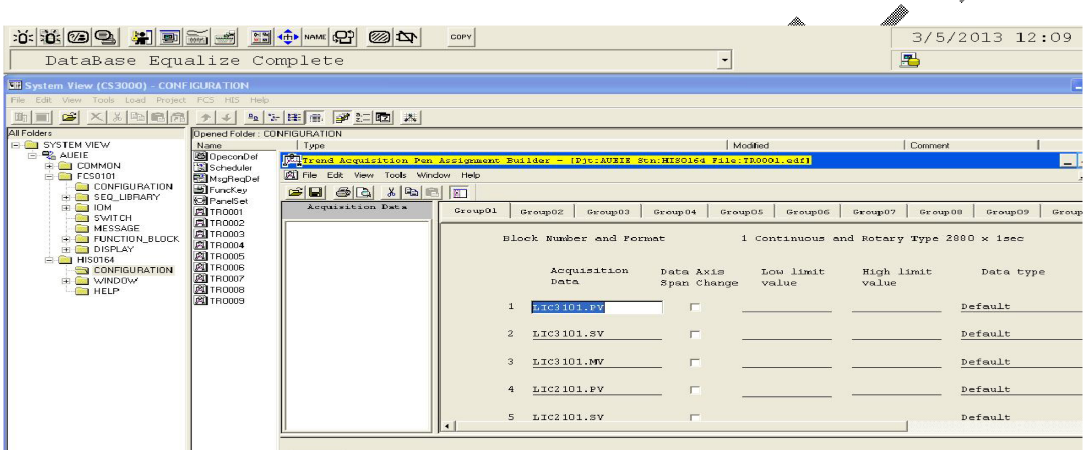


Fig.5

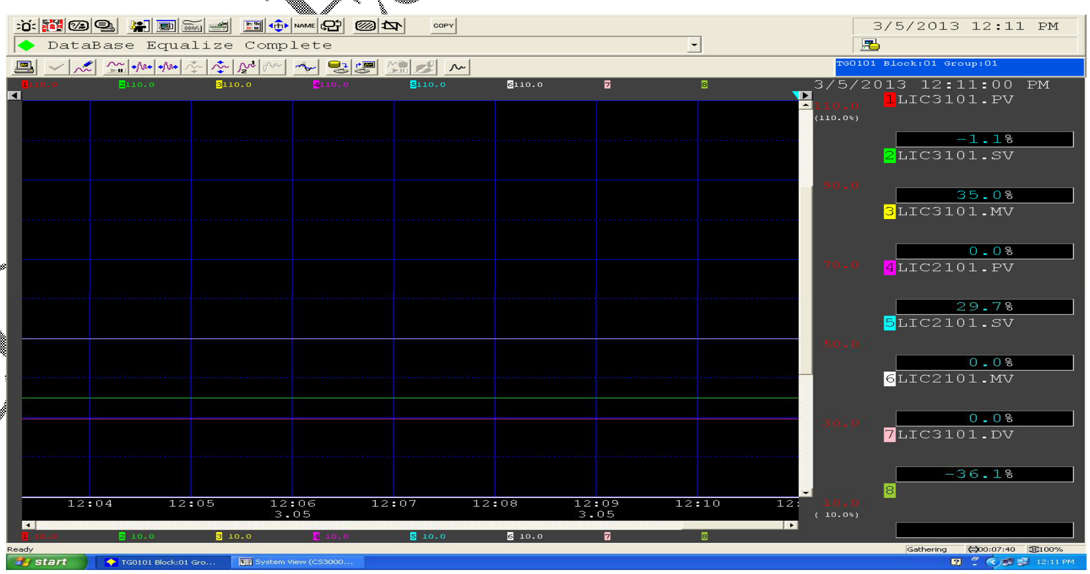


Fig.6

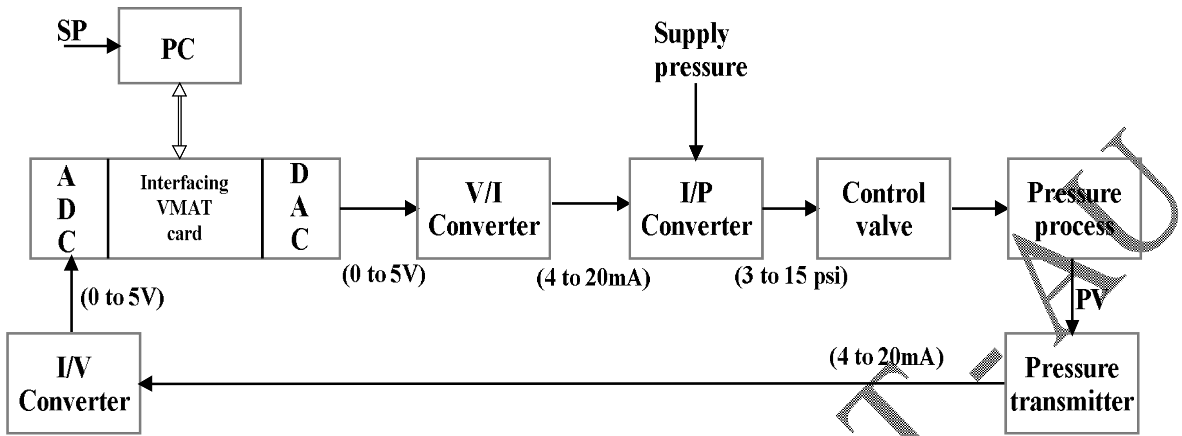
Result: Thus Level process control is studied through DCS.

Pin Assignments for the 68-pin I/O connector on the PCI-6023E, PCI-6024E, and DAQ Card-6024E.

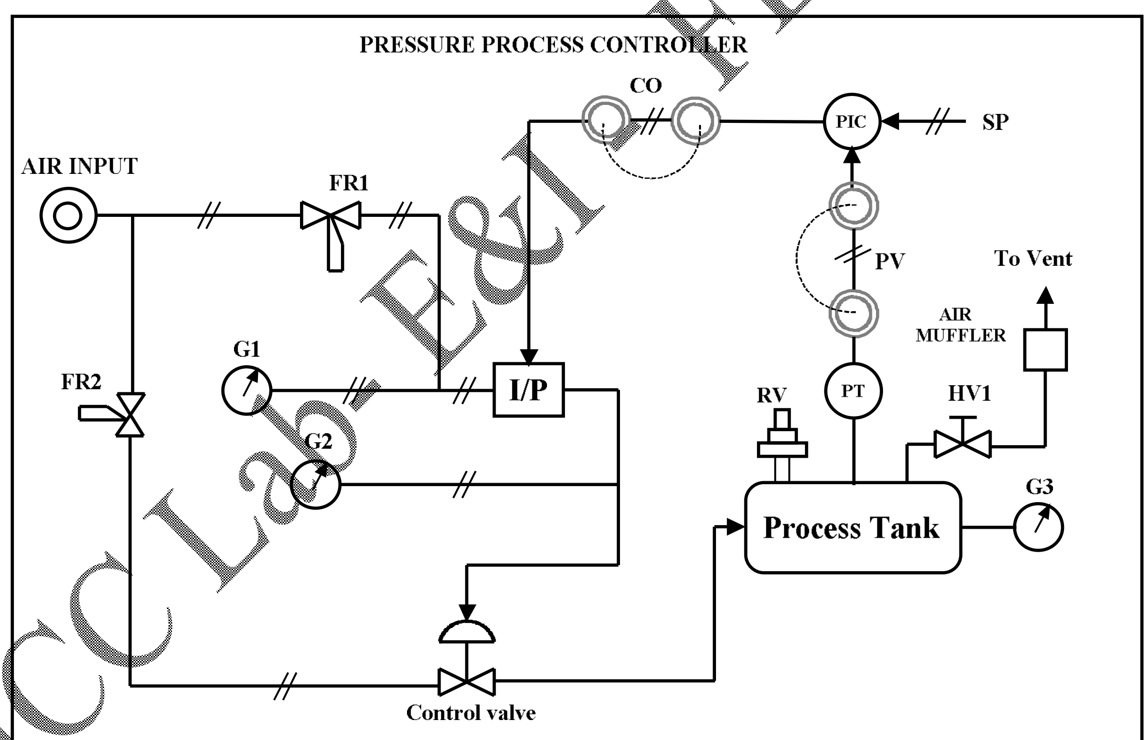
Signal	Pin	Pin	Signal
ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7
DAC0OUT[1]	22	56	AIGND
DAC1OUT[1]	21	55	AOGND
RESERVED	20	54	AOGND
DIO4	19	53	DGND
DGND	18	52	DIO0
DIO1	17	51	DIO5
DIO6	16	50	DGND
DGND	15	49	DIO2
+5 V	14	48	DIO7
DGND	13	47	DIO3
DGND	12	46	SCANCLK
PFI0/TRIG1	11	45	EXTSTROBE*
PFI1/TRIG2	10	44	DGND
DGND	9	43	PFI2/CONVERT*
+5 V	8	42	PFI3/GPCTR1_SOURCE
DGND	7	41	PFI4/GPCTR1_GATE
PFI5/UPDATE*	6	40	GPCTR1_OUT
PFI6/WFTRIG	5	39	DGND
DGND	4	38	PFI7/STARTSCAN
PFI9/GPCTR0_GATE	3	37	PFI8/GPCTR0_SOURCE
GPCTR0_OUT	2	36	DGND
FREQ_OUT	1	35	DGND

[1] Not available on the 6023E

General Block diagram



P&ID of Pressure Process



Technical Specifications of Pressure Process

Pressure tank capacity:	max 70 psi
Control valve:	Air to Open
Pressure sensor:	Piezo-electric type