Rensselaer Polytechnic Institute

ECSE-4760
Real-Time Applications in Control & Communications

HYBRID SIMULATION OF A CONTROLLED SYSTEM

Number of Sessions – 4 [1]

INTRODUCTION

Analog control systems have been used in industrial environments for many years. An average plant uses many controllers, each controlling one small segment, or control loop, of the entire plant. These controllers are independent and used to be set and monitored by human operators. Digital control systems are now popular in control applications. In Direct Digital Control (DDC) systems, the computer replaces the analog controllers altogether. A typical DDC application is shown in FIG. 1.

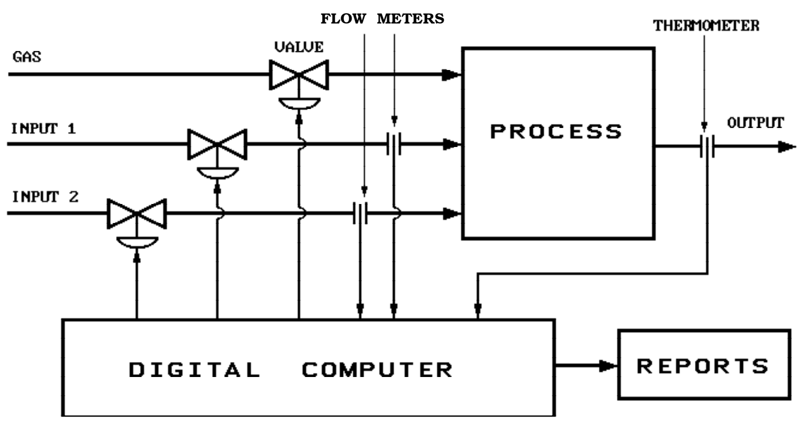


FIGURE 1. Typical direct digital control application.

The main advantage of a DDC system is its flexibility. Changing a few operating parameters or implementing a complete new control strategy in most cases is just a matter of recompiling a program module, whereas with hardware based controllers, delicate and expensive hardware must be modified or rendered unusable. The computer is reprogrammable; several control programs can be stored and may be switched quickly to provide a variety of control operations. Typical operations of the DDC programs are:

1) Co-ordination of several control loops

[1] Four sessions for 2 of the 3 PID controllers, an FST controller (with or without delay), and an analog or digital State Feedback controller.

2) Adjusting to changes in the dynamics of the process
3) Providing a fast and orderly system shut-down and restart when needed
4) Providing preventative maintenance and test services
5) Providing printed data and reports for plant personnel.

Microprocessor based systems with their relatively low cost and high computational speeds can provide these services for almost any size process. Minicomputers that used to handle applications in process control for large systems, rapidly give way to fast 32bit single board computers. Such is the power of today's microprocessor based control systems, that specialized interrupt driven real time operating systems, or real time extensions to industry standard operating systems (UNIX) exist, and control engineers do their development in high level languages assisted by powerful software and hardware debugging tools. Typical examples of microprocessor based control systems range from car fuel controllers, to the advanced F-18 fighter plane whose complete flight controls are implemented in high level software (ADA). Computer control systems combine the disciplines of Automatic control and Software Engineering as well as analog devices and digital hardware to give Electrical and Computer Systems Engineers many opportunities for challenging and creative design work.

This experiment will involve the use of a digital computer to implement the closed loop control. Different control schemes will be studied along with the effects of sampling time and time delays. The student is encouraged to explore as many of the topics presented as time permits. The experiment gives an excellent chance to apply control theory in an interactive way to gain a better understanding of hybrid simulation techniques and control theory in general.

EXPERIMENTAL OBJECTIVES

Several aspects of digital control will be explored in this experiment. Four digital implementations of analog controllers and two types of direct digital controllers will be used. Different controller design schemes will be compared and the effects of sampling rate and time delays will be explored. It should be noted here that all process time delays will be implemented on the PC. This is only done to simulate a process with a delay with relative ease[2].

A single control loop will be used in this experiment. Both the general time and *s*-plane (**Laplace Transform**) representations of a typical closed loop system are shown on FIG. 2. More analytically the process $G_p(s)$ gives the response $y(t)$ ($Y(s)$), to a control input $u(t)$ ($U(s)$). The controller $G_c(s)$ calculates the control signal using the error signal $e(t)$ ($E(s)$), that is the difference between the desired output (reference input) $r(t)$ ($R(s)$) and the process output. The transducer is used for scaling and unit of measure conversions and will be assumed unity here. Furthermore it will be assumed that no disturbance or other type of noise enters the process.

[2] In practice, process time delays are harmful and affect the stability of the controlled system. A controller should never purposely insert a delay. Unavoidable delays could be caused by slow processes, converters, or remote sensing devices (e.g. Earth-Mars remote control signals).

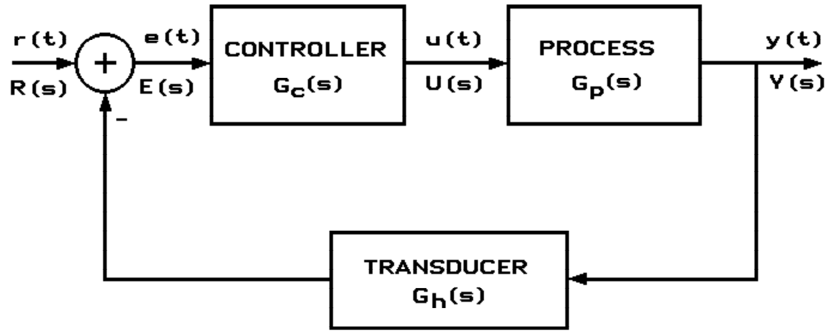


FIGURE 2. *t* and *s* Domain block diagram of a closed loop system.

CONTROLLER CALCULATIONS

Three types of digital controllers are investigated in this experiment. The discrete **PID** controller, the **FST** controller, and the **S**tate **F**eedback controller. The discrete **Proportional + Integral + Derivative** (**PID**) controller to be used is a digital approximation of the continuous **PID** controller. Three design schemes with different assumptions about the process dynamics model are investigated. The **Finite Settling Time** (**FST**) controller is designed specifically for digital control, and one design scheme is presented. The **State Feedback** controller uses a **Pole Placement** design scheme and is implemented in both the continuous and the discrete cases.

PID CONTROLLER

The continuous PID controller has been widely used in industry for many years. It is easy to implement, gives a wide range of control characteristics, and is easily adapted to meet changes in control requirements. Its complete transfer function is very intuitive to conceive, containing a term proportional to the present error (K_p), a term representing the error history (K_i), and an error anticipatory term (K_d) and is given in equation (1):

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \qquad (1)$$

where:

K_p is the proportional gain
K_i is the integral gain
K_d is the derivative gain.

Depending on what terms are actually present when a PID controller is implemented, there are several variations to basic transfer function:

1) Proportional (**P)** controller: the K_i and K_d terms are zero.
2) Proportional + Integral (**PI**) controller: the K_d term is zero.
3) Proportional + Derivative (**PD**) controller: the K_i term is zero.
4) Proportional + Integral + Derivative (**PID**) controller: all terms are non zero.

The digital PID controller is a discrete approximation formed from the above transfer function.

Its derivation stems from the approximation of the derivative with a difference as shown in equation (2). Thus the integral term is replaced with a sum and the derivative term is replaced with a simple difference. A more thorough discussion of the digital PID controller can be found in Cadzow and Martens[1] p. 100.

$$\dot{x}(t)=\frac{dx}{dt}=\lim_{T\to 0}\frac{x(t+T)-x(t)}{T}\approx\frac{x(t+T)-x(t)}{T} \tag{2}$$

The difference equation for the digital PID controller is:

$$u(k) = K_p e_1(k) + K_i e_{22}(k) + K_d e_{23}(k) \tag{3}$$

where:

$U(k)$ is the control at the k^{th} sample instant,
$e_1(k) = r(k) - y(k)$, the error term,
$e_{22}(k) = e_{22}(k - 1) + Te_1(k)$ the integral approximation,
$e_{23}(k) = [e_1(k) - e_1(k - 1)]/T$ the derivative term,
T is the sampling time in *seconds*, $r(k)$ is the reference signal and $y(k)$ is the output signal.

The digital controller is a good approximation of the continuous controller when the sampling time T is short relative to the time constants of the system.

Many design schemes exist for finding the PID controller gains. Schemes using the root-locus or frequency response calculations are required when design specifications are rigid. In other applications, simpler design schemes may be used. Three such schemes will be considered, namely the methods of:

1) Ziegler-Nichols.
2) Gallier-Otto.
3) Graham-Lathrop.

In the first two schemes, parameters are used to characterize the process. The controller gains are defined in terms of these parameters. The third scheme is based on the characteristic equation of the closed loop system; the controller gains are chosen such that the characteristic equation matches one of a set of standard forms.

A. Ziegler-Nichols Design

The process is characterized by two parameters derived from the unit step response of the process. These parameters, R and L, represent the speed and the reaction time of the process. They are found from a plot of the unit step response as shown in FIG. 3. Note that for systems with unity gain (output settles to the same value as the input), FIG. 3 would have a y-axis scale value of 1.0 for the steady state value of output to the unit step.

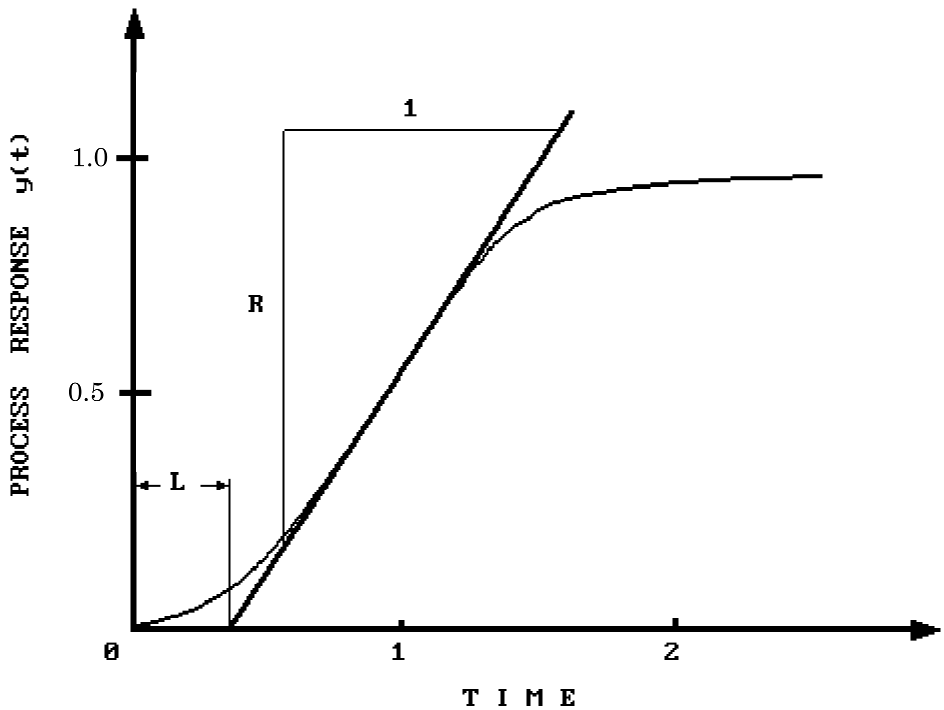


FIGURE 3. Example of a unit step response.

A line, tangent to the curve, is drawn at the point of maximum slope. This line can be represented by the equation:

$$c(t) = R(t - L) \quad (4)$$

where:

R is the maximum slope of the curve,

L is the time where the tangent line crosses the time axis.

J. G. Ziegler and N. B. Nichols have developed a design scheme using these parameters. This scheme gives controllers with a fast response time but with significant overshoot. For most processes, the system closed loop transfer function will have dominant poles near the rays as shown in FIG. 4. Ziegler and Nichols give the following gain settings for the various PID controller implementations[2]:

For the P control:

$$K_p = 1/RL$$

For the PI control:

$$K_p = .9/RL, \qquad K_i = .3K_p/L$$

For the PID control:

$$K_p = 1.2/RL, \qquad K_i = .5K_p/L, \qquad K_d = .5LK_p$$

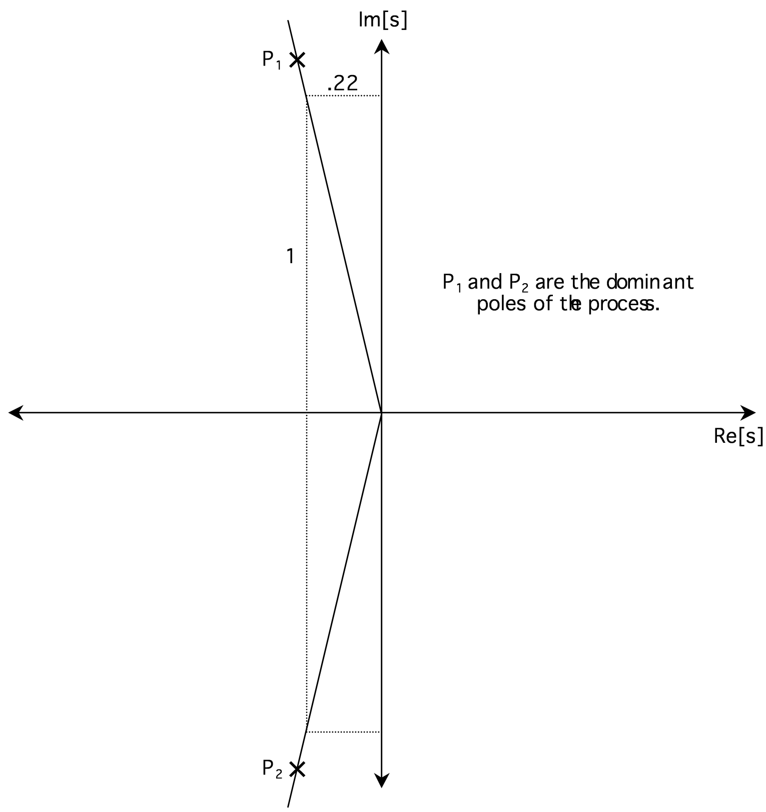


FIGURE 4. Approximate system dominant poles using Ziegler-Nichols gains.

As an example, consider a process with the transfer function:

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

The graph of the unit step response is shown in FIG. 3. The parameter values may be read from the graph as

$$R = 0.25 \text{ and } L = 0.38$$

The Ziegler-Nichols gains are then:
For the P control:

$$K_p = 10.5, \quad K_i = K_d = 0$$

For the PI control:

$$K_p = 9.47, \quad K_i = 7.48, \quad K_d = 0$$

For the PID control:

$$K_p = 12.63, \quad K_i = 16.62, \quad K_d = 2.40$$

B. Gallier-Otto Design

Many processes exhibit behavior like that of a second order time-invariant linear system with time delay. These processes may be approximated by the transfer function

$$G_p(s) = \frac{K e^{-T_D s}}{(T_1 s + 1)(T_2 s + 1)} \tag{6}$$

where:

K is the process gain,
T_1 is the smaller time constant,
T_2 is the larger time constant,
T_D is the time delay.

In practice, this approximation is made by inputting a step or a short pulse to the process and recording the output. The parameters K, T_1, T_2, and T_D are then found by applying regression analysis to fit the transfer function to this response.

A simple transformation of the parameters facilitates the comparison of the different processes. The ratio of the time constants is defined as:

$$\lambda = T_1/T_2$$

Time is normalized using the normalization constant

$$T_n = T_1 + T_2 + T_D$$

The normalized time delay (normalized deadtime) is then

$$t_D = T_D/T_n$$

P. W. Gallier and R. F. Otto[3] have developed a design scheme which uses the transformed parameters λ, T_n, and t_D. The scheme is based on the minimization of the integrated absolute error performance index IAE given by equation (7):

$$IAE = \int_0^\infty |e(t)| dt \tag{7}$$

The optimal values for the controller

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \tag{8}$$

were found for various values of λ and t_D.

Gallier and Otto compiled graphs relating the controller parameters to the process parameters λ and t_D, for both PI and PID control. In particular, these graphs give optimal values for the normalized parameters:

$K_0 = K_p K$, the loop gain of the closed loop system,
$t_i = T_i/T_n$, the normalized reset time,
$t_d = T_d/T_n$, the normalized derivative time.

The gains for the PID controller used in this experiment are then derived in terms of these parameters. Thus, for the PID controller

$$G_c(s) = K_p + K_i/s + K_d s$$

the three coefficients become:

$$K_p = \frac{K_0}{K}$$
$$K_i = \frac{K_p}{T_i} = \frac{K_p}{t_i T_n} \qquad (9)$$
$$K_d = K_p T_d = K_p t_d T_n$$

Continuing with the example transfer function as in equation (5), the process parameters are:

$$K = 1, \quad T_1 = 1, \quad T_2 = 2, \quad T_D = 0$$

Thus, the transformed parameters are:

$$\lambda = 0.5, \quad t_D = 0, \quad T_n = 3$$

Referring to graphs of FIG. 5, the controller gains become:

$$K_p = 4, \quad K_i = 1.12,$$

for PI control and

$$K_p = 7.5, \quad K_i = 2.78, \quad K_d = 4.5$$

for PID control.

C. Graham-Lathrop

D. Graham and R. C. Lathrop[4] have derived standard forms for the characteristic equation of the closed loop system as shown on TABLE 1.

The derivation is based on the minimization of the integrated time absolute error (*ITAE*) given in (10):

$$ITAE = \int_0^\infty t|e(t)|dt \qquad (10)$$

The *ITAE* performance index gives less weight to the initial error signal than the *IAE* index. The resulting system has a more oscillatory response than the one derived from the *IAE* index. The roots of the standard forms for orders 1, 2, 3, and 4 are shown in FIG. 6.

Recall that the characteristic equation of a closed loop system is the denominator of the **C**losed **L**oop **T**ransfer **F**unction (**CLTF**), and for a system like the one shown in FIG. 2 the CLTF is given by (remember $H(\boldsymbol{s})$ is unity):

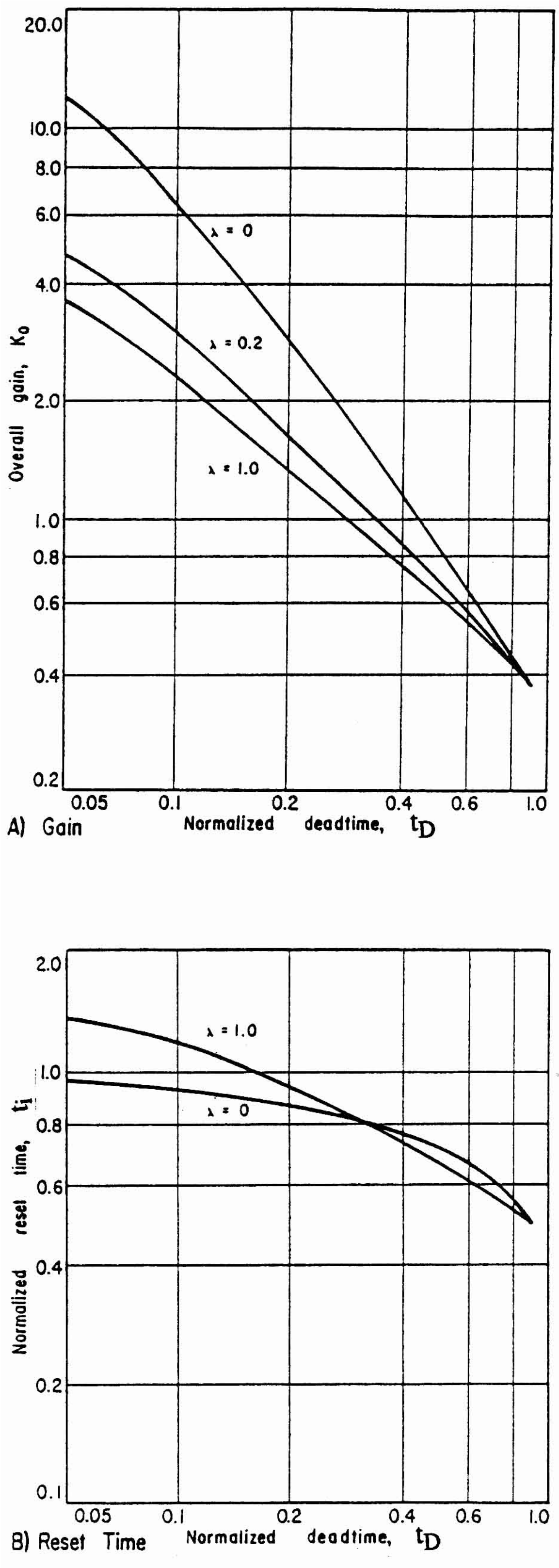


Figure 2. Tuning coefficients for two-mode control with minimum IAE $(\lambda = T_1/T_2)$

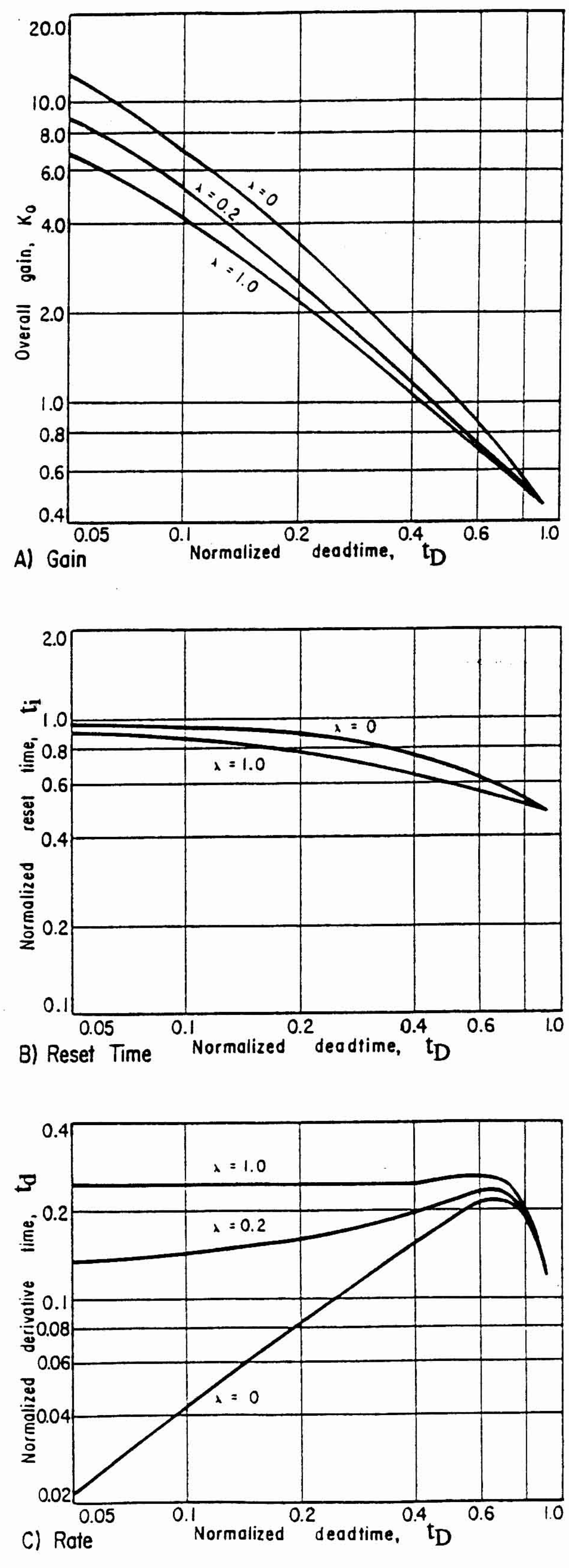


Figure 3. Tuning coefficients for three-mode control with minimum IAE $(\lambda = T_1/T_2)$

FIGURE 5. Gallier-Otto gains for PI and PID control.

	Closed Loop Characteristic Equations as derived by Graham & Lathrop
1	$s + \omega_0$
2	$s^2 + 1.4\omega_0 s + \omega_0^2$
3	$s^3 + 1.75\omega_0 s^2 + 2.15\omega_0^2 s + \omega_0^3$
4	$s^4 + 2.1\omega_0 s^3 + 3.4\omega_0^2 s^2 + 2.7\omega_0^3 s + \omega_0^4$
5	$s^5 + 2.8\omega_0 s^4 + 5.0\omega_0^2 s^3 + 5.5\omega_0^3 s^2 + 3.4\omega_0^4 s + \omega_0^5$
6	$s^6 + 3.25\omega_0 s^5 + 6.6\omega_0^2 s^4 + 8.6\omega_0^3 s^3 + 7.45\omega_0^4 s^2 + 3.95\omega_0^5 s + \omega_0^6$
7	$s^7 + 4.47\omega_0 s^6 + 10.42\omega_0^2 s^5 + 15.08\omega_0^3 s^4 + 15.54\omega_0^4 s^3 + 10.64\omega_0^5 s^2 + 4.58\omega_0^6 s + \omega_0^7$
8	$s^8 + 5.2\omega_0 s^7 + 12.8\omega_0^2 s^6 + 21.6\omega_0^3 s^5 + 25.75\omega_0^4 s^4 + 22.22\omega_0^5 s^3 + 13.3\omega_0^6 s^2 + 5.15\omega_0^7 s + \omega_0^8$

TABLE 1. Graham-Lathrop standard forms.

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \tag{11}$$

According to their scheme the closed loop system characteristic equation is first normalized by dividing through by the coefficient of the highest order term. It is then compared with the standard form of the same order from TABLE 1, and the coefficients of each order of ***s*** are equated. The resulting equations are then solved for the controller gains.

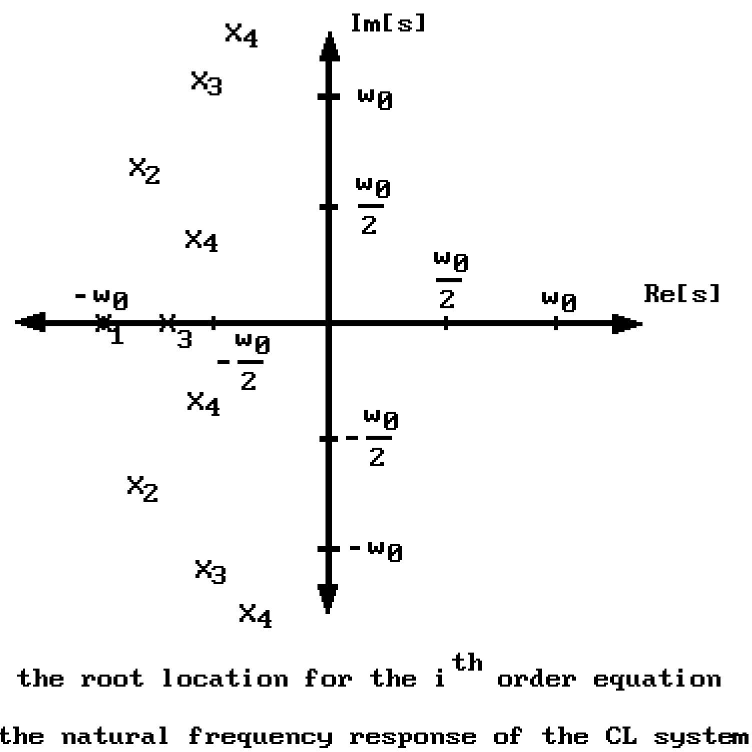


FIGURE 6. Roots for 1st, 2nd, 3rd, and 4th order Graham-Lathrop forms.

Continuing with the original example, the normalized characteristic equation for the closed loop system is:

$$s^3 + \frac{3 + K_d}{2} s^2 + \frac{1 + K_p}{2} s + \frac{K_i}{2} = 0 \tag{12}$$

• For **P** control, equation (12) is reduced to

$$s^2 + 1.5s + .5(1 + K_p) = 0$$

The second order standard form, from TABLE 1 row 2, is

$$s^2 + 1.4\omega_0 s + \omega_0^2 = 0$$

Thus the coefficient matching results in

$$1.4\omega_0 = 1.5, \qquad \omega_0^2 = .5(1 + K_p)$$

Solving for the unknowns yields the controller gain:

$$\omega_0 = 1.07, \qquad K_p = 1.296$$

• For the **PI** control the characteristic equation (12) becomes

$$s^3 + 1.5s^2 + .5(1 + K_p)s + .5K_i = 0$$

The third order standard form is

$$s^3 + 1.75\omega_0 s^2 + 2.15\omega_0^2 s + \omega_0^3$$

Thus,

$$1.75\omega_0 = 1.5, \qquad 2.15\omega_0^2 = .5(1 + K_p), \qquad \omega_0^3 = .5K_i$$

Solving for the gains:

$$\omega_0 = 0.857, \qquad K_p = 2.159, \qquad K_i = 1.252$$

• For **PID** control the standard form is still the TABLE 1 row 3 equation and we end up having three equations with four unknowns (one degree of freedom). Values of 1.0 and 1.5 will be chosen for ω_0.

$\omega_0 = 1.0$ yields:

$$K_p = 3.30, \qquad K_i = 2.00, \qquad K_d = 0.50$$

$\omega_0 = 1.5$ yields:

$$K_p = 8.68, \qquad K_i = 6.75, \qquad K_d = 2.25$$

As can be verified from FIG. 6, the larger the ω_0 is, the faster the response becomes.

FINITE SETTLING TIME CONTROLLER

A. FST Controller for a 2nd Order Process without Time Delay

At low sampling rates, controllers designed using discrete time theory must be used because digital approximations to continuous controllers result in poor performance. One such controller is the Finite Settling Time (**FST**) ripple free controller. This controller is designed to drive the process to a set point in a small finite number of sample periods. FIG. 7 shows the sampled data version of the same closed loop control system as in FIG. 2. The error is the input to the controller at the sample instants $t = kT$, $k = 0, 1, ...,$ where T is the sampling period. The controller sends an output at each sample instant to a Zero Order Hold, which sends it to the process for the entire sample period. FIG. 8 shows the **z-transform** block diagram and the signal flow.

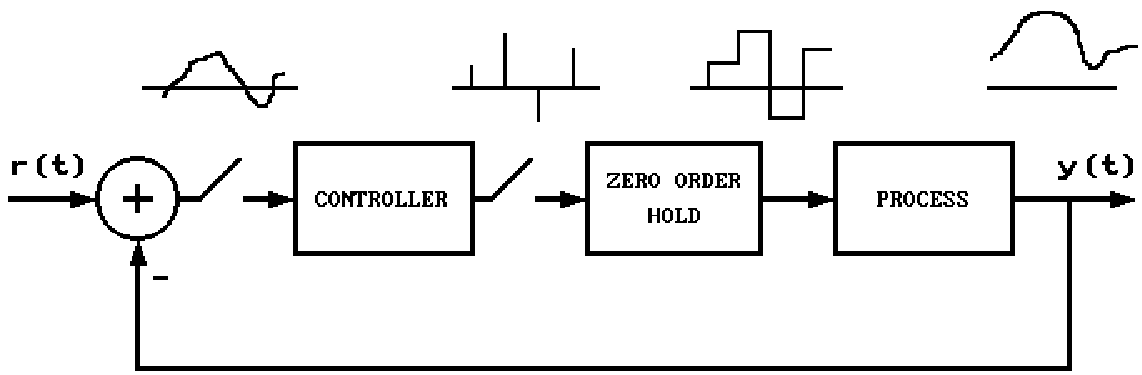


FIGURE 7. Sampled data version of the continuous system of FIG. 2.

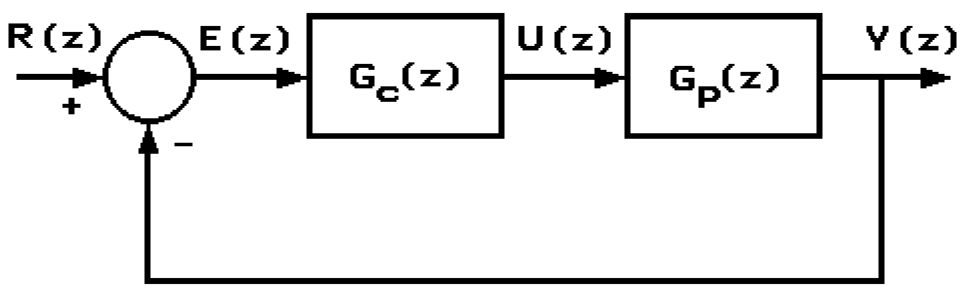


FIGURE 8. Typical *z*-plane block diagram.

To illustrate the rules that apply to ripple free design, consider the second order process given in a Laplace form as:

$$G_p(s) = \frac{1}{(T_1 s + 1)(T_2 s + 1)} \tag{13}$$

The process transfer function with a zero order hold on it's input is given by:

$$G_p(s) = \frac{1 - e^{-Ts}}{s(T_1 s + 1)(T_2 s + 1)} \tag{14}$$

The process pulse transfer function is given by:

$$G_p(z) = \frac{[\,1 - (1 + d_1)p_1 - (1 + d_2)p_2\,]z^{-1} + [\,p_1p_2 + d_1p_1 + d_2p_2\,]z^{-2}}{(1 - p_1z^{-1})(1 - p_2z^{-1})} \tag{15}$$

where:

$$\begin{aligned} p_1 &= e^{-T/T_1} \\ p_2 &= e^{-T/T_2} \\ d_1 &= \frac{-T_2}{T_2 - T_1} \\ d_2 &= \frac{T_1}{T_2 - T_1} \end{aligned} \tag{16}$$

The process pulse transfer function can be transformed into the general form:

$$G_p(z) = \frac{c_1z^{-1} + c_2z^{-2}}{(1 - p_1z^{-1})(1 - p_2z^{-1})} \tag{17}$$

where:

$$\begin{aligned} c_1 &= 1 - (1 + d_1)p_1 - (1 + d_2)p_2 \\ c_2 &= p_1p_2 + d_1p_1 + d_2p_2 \end{aligned} \tag{18}$$

The general FST controller has the form

$$u(k) = \sum_{i=0}^{N_e} K_{ei}e(k - i) + \sum_{i=1}^{N_u} K_{ui}u(k - i) \tag{19}$$

where:

$e(k - i)$ is the error term at the k^{th} sample instant
$u(k - i)$ is the control term at the k^{th} sample instant
K_e is the vector of gains for the error terms
K_u is the vector of gains for the control terms
N_e is the order of the error sum
N_u is the order of the control sum

The design procedure developed by Ragazzini and Franklin[5] was applied to the above transfer function for a unit step input. The resulting parameters are:

$$\begin{aligned} N_e &= 2 \\ N_u &= 2 \\ K_{e0} &= \frac{1}{c_1 + c_2} \\ K_{e1} &= \frac{-p_1 - p_2}{c_1 + c_2} \\ K_{e2} &= \frac{p_1p_2}{c_1 + c_2} \end{aligned} \tag{20}$$

$$K_{u1} = \frac{c_1}{c_1 + c_2}$$

$$K_{u2} = \frac{c_2}{c_1 + c_2}$$

The resulting controller has the following difference equation:

$$u(k) = K_{e0}e(k) + K_{e1}e(k - 1) + K_{e2}e(k - 2) + K_{u1}u(k - 1) + K_{u2}u(k - 2) \qquad (21)$$

Note that as T gets smaller, the K_e terms increase. For T < **1 second** the control values which are proportional to the K_e terms will need to exceed their maximum value of **10 Volts** to properly control the system, unless the input step size is very small. For the purposes of this experiment, sampling time T for the FST controller should be greater than 1 second.

B. FST Controller for a 2nd Order Process with Time Delay

The process with time delay may be approximated by the transfer function

$$G_p(s) = \frac{e^{-T_D s}}{(T_1 s + 1)(T_2 s + 1)} \qquad (22)$$

where

T_1 and T_2 are the process time constants.

T_D is the time delay implemented as an integral number of sample periods ($T_D = MT$).

The process transfer function with zero order hold is given by:

$$G_p(s) = \frac{(1 - e^{-Ts})e^{-MTs}}{s(T_1 s + 1)(T_2 s + 1)} \qquad (23)$$

where

M is the number of sample periods of time delays.

T is the sampling period.

The process pulse transfer function[3] is

$$G_p(z) = \frac{(c_1 z^{-1} + c_2 z^{-2})z^{-M}}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})} \qquad (24)$$

where c_1, c_2, p_1, and p_2 were defined in the previous section. Applying the same procedure as before (system without time delay), one can find that the controller equation given in equation (25) is exactly the same as the equation (21), except for the fact that the control history is delayed by M samples.

$$u(k) = K_{e0}e(k) + K_{e1}e(k - 1) + K_{e2}e(k - 2) + K_{u1}u(k - 1 - M) + K_{u2}u(k - 2 - M) \qquad (25)$$

where all the coefficients are given by equations (20).

---

3 Remember $e^{Ts} = z$

STATE FEEDBACK CONTROLLER USING THE POLE PLACEMENT METHOD

A. Introduction to State Variable and Pole Placement

In all previous cases the Input/Output representation is used as the plant model. Under this representation the plant always starts from rest (all initial conditions assumed zero) and more importantly it is assumed to be a "closed black box", accessible only though the input and the output ports. The controller is an entirely different unit cascaded in front of the plant, with possibly a transducer measuring, unit-converting and feeding back the output. When the controller is fast enough to be assumed continuous, then the analysis is done in the ***s*** domain using the **Laplace transforms** of the controller and plant transfer functions $G_c(s)$, $G_p(s)$ respectively. When sampling rate is too slow, or other factors are forcing the whole process to be assumed discrete, then the analysis is done in the ***z*** domain using the **z-transforms** of the transfer functions $G_c(z)$, $G_p(z)$.

Even though this representation is still very popular among control engineers, the resulting controller isn't always as successful as would be expected when initially designed, or, in rare cases, it may completely fail to control the plant. These failures can be attributed to unexpected initial conditions, unsuccessful pole cancellation or excitation of hidden oscillatory modes. Further discussion of these cases is beyond the scope of this experiment, yet the student can find more examples and explanation in[6] and [7].

As a remedy to the above problems the state variable representation is used extensively. The plant is modeled using the state and output equations[8] , thus all the information about oscillatory modes or initial conditions is readily available. A typical second order, time invariant plant with two state variables x_1 and x_2 is shown in FIG. 9.

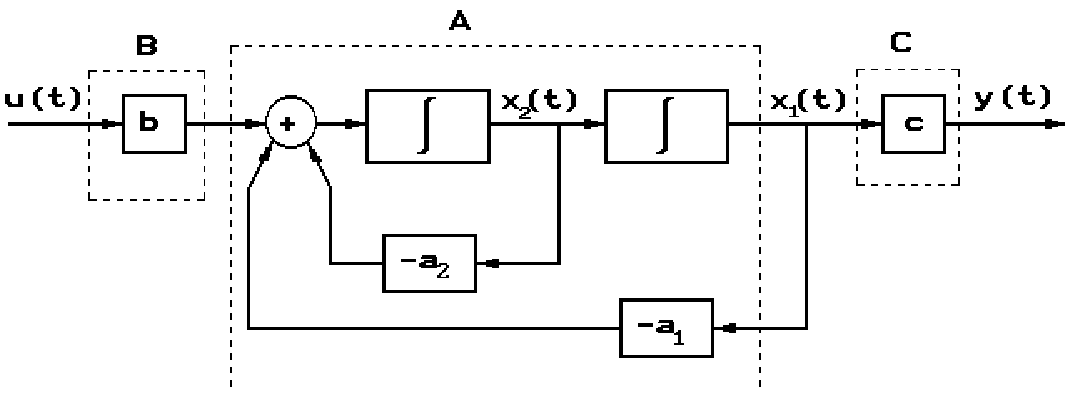


FIGURE 9. Block diagram of a typical 2nd order plant.

The plant is represented in state variable form (assuming $t \geq 0$) by a set of differential equations as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) & x_1(0) &= x_{10} \\ \dot{x}_2(t) &= -a_1x_1(t) - a_2x_2(t) + bu(t) & x_2(0) &= x_{20} \\ y(t) &= cx_1(t) \end{aligned} \tag{26}$$

or in matrix form:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A_c x}(t) + \boldsymbol{B_c u}(t) \ , & \boldsymbol{x}(0) &= \boldsymbol{x}_0 \\ \boldsymbol{y}(t) &= \boldsymbol{C_c x}(t) \end{aligned} \tag{27}$$

where:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad \boldsymbol{A_c} = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix}, \quad \boldsymbol{B_c} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \quad \boldsymbol{C_c} = [\, c \quad 0 \,], \quad \boldsymbol{x}_0 = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

To return to the familiar s plane input/output representation, we must assume that the initial conditions $\boldsymbol{x}_0$ are zero, take the Laplace transform of equations (26) and eliminate the state variables x_1 and x_2 from them. The student can verify that the resulting transfer function $G_p(s)$, and the system characteristic equation are given by equations (28):

$$\frac{Y(s)}{U(s)} = H(s) = \frac{bc}{s^2 + a_2 s + a_1}$$

$$\det [\, s\boldsymbol{I} - \boldsymbol{A_c} \,] = s^2 + a_2 s + a_1 = 0 \qquad (28)$$

As mentioned before, when we are using the state variable representation the classic notion of a separate controller with its own transfer function, being cascaded with the plant (as in FIG. 2) is no longer valid. Instead we achieve the desired output $y(t)$ by feeding back[4] the system states amplified by properly selected gains, thus in essence by modifying the state equations.

Depending upon what the overall design specifications for the system behavior are, there are several methodologies that can be applied to yield a set of state feedback gains. The **Pole Placement Method**[9][10] to be employed here, requires that such feedback gains $\boldsymbol{K_c}$ be found, so that the **poles** of the characteristic equation of the new system match a set of preselected (desired) values. The gains are found following the steps below:

- a desired polynomial is calculated that has as its roots the desired set of poles;
- the new (augmented) system characteristic equation is found as an expression of the unknown gains;
- by matching the coefficients of the above two polynomials, a set of simultaneous equations is constructed;
- solving the above equation set yields the feedback gains (if a unique set can be found).

B. Pole Placement for the Continuous Case

We'll be using the same general plant introduced in FIG. 9 and require that the closed loop system has p_1, and p_2 as its poles. Thus the desired characteristic polynomial becomes:

$$(s - p_1)(s - p_2) = s^2 + (-p_1 - p_2)s + (p_1 p_2) = 0$$

$$s^2 + ms + n = 0, \quad -m = p_1 + p_2, \quad n = p_1 p_2 \qquad (29)$$

The augmented system with the unknown feedback gains k_{c1}, k_{c2} and a reference input signal $r(t)$ all present, is shown in FIG. 10. For this type of system-control combination it is necessary to amplify the reference signal with a feedforward gain g (to be found), if we want zero steady state error (output matching the input) under a step input.

---

[4] This implies that we have full access to all states, introducing the notions of **controllability** and **observability** that go beyond the scope of this experiment.

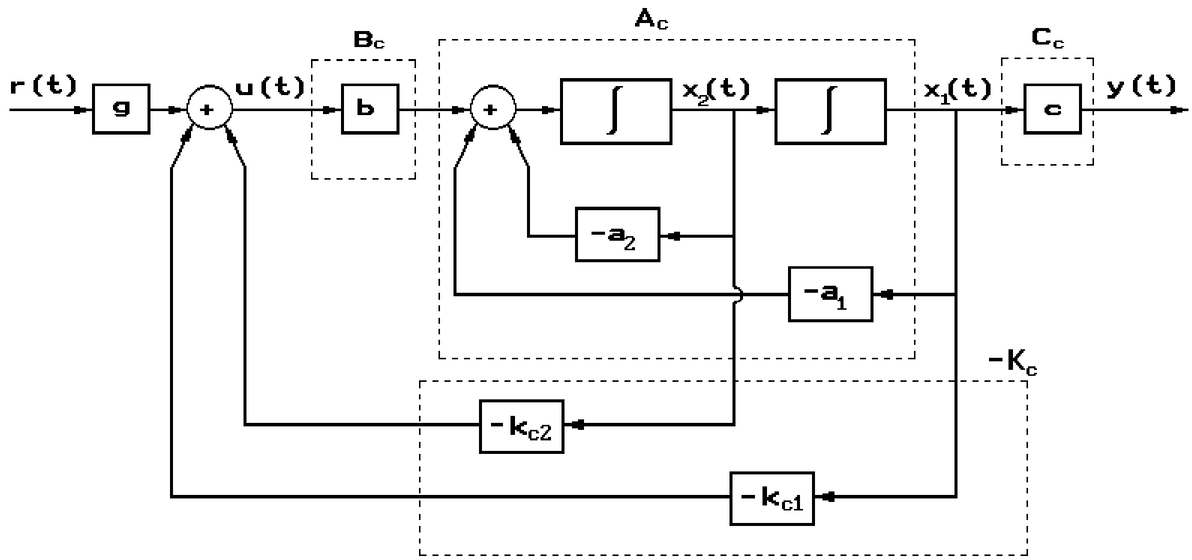


FIGURE 10. Augmented system with feedback controller and reference.

Assuming zero initial conditions as before, the new state equations are:

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -a_1x_1(t) - a_2x_2(t) - k_{c1}bx_1(t) - k_{c2}bx_2(t) + gbr(t) \qquad (30)$$
$$y(t) = cx_1(t)$$

From direct comparison between equations (30) and (26) it's easy to note that the control law $u(t)$ is given by:

$$u(t) = gr(t) - [k_{c1}x_1(t) + k_{c2}x_2(t)] = gr(t) - \boldsymbol{K_c}\boldsymbol{x}(t) \qquad (31)$$

with $\boldsymbol{K_c} = [\,k_{c1} \quad k_{c2}\,]$ which precisely is the formula for the state feedback control. Equations (30) can be written in matrix form as:

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A_c} - \boldsymbol{B_c}\boldsymbol{K_c})\boldsymbol{x}(t) + \boldsymbol{B_c}gr(t)\ , \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0$$
$$\boldsymbol{y}(t) = \boldsymbol{C_c}\boldsymbol{x}(t) \qquad (32)$$

As in the open loop case (equations (28)), the overall transfer function and the characteristic equation of the closed loop system are:

$$\frac{Y(s)}{U(s)} = H(s) = \frac{gbc}{s^2 + (a_2 + bk_{c2})s + (a_1 + bk_{c1})}$$
$$\det[\,s\boldsymbol{I} - \boldsymbol{A_c}\,] = s^2 + (a_2 + bk_{c2})s + (a_1 + bk_{c1}) = 0 \qquad (33)$$

and by coefficient matching with equation (29) the gains are found. With no loss of generality from now on the c coefficient in equations (33) is assumed to be equal to 1.

The unknown feedforward gain g is calculated as follows: The error between the reference signal $r(t)$ and the output $y(t)$ is: $e(t) = r(t) - y(t)$, and since our reference input is a **step** function of size R, the error becomes $e(t) = R - y(t)$. Thus the steady state error e_{ss} becomes:

$$e_{ss} = \lim_{t \to \infty} (R - y(t)) = R - y_{ss} \tag{34}$$

Applying the **Final Value** theorem to equation (32a) and noting that $R(s) = R/s$, we calculate y_{ss}

$$y_{ss} = \lim_{t \to \infty} y(t) = \lim_{s \to 0} sY(s) = \lim_{t \to \infty} s \frac{gb\frac{R}{s}}{s^2 + (a_2 + bk_{c2})s + (a_1 + bk_{c1})} = \frac{gbR}{a_1 + bk_{c1}}$$

We want zero steady state error hence y_{ss} must be equal to R, which identifies g as a function of the gains $\boldsymbol{K_c}$:

$$\mathrm{g} = \frac{a_1}{b} + k_{c1} \tag{35}$$

Assuming that the gain matrix $\boldsymbol{K_c}$ is already found by polynomial coefficient matching in the pole placement part, the feedforward gain g is also known.

C. Pole Placement for the Discrete Case

When the sampling period T becomes very large the analysis should be done in the z (discrete) domain in order to ensure accurate control. Thus the desired closed loop poles specified in the s domain are converted to the z domain by using the relationship:

$$z_i = \mathrm{e}^{s_i T} \tag{36}$$

where T is the sampling time.

As in the continuous case the discrete desired second order polynomial is

$$z^2 + qz + p = 0, \quad \text{with} \quad -q = z_1 + z_2, \quad \mathrm{p} = z_1 z_2 \tag{37}$$

Furthermore the equivalent discrete-time system is derived from the continuous [11] as:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A_d}\boldsymbol{x}(k) + \boldsymbol{B_d}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C_d}\boldsymbol{x}(k) \end{aligned} \tag{38}$$

where $\boldsymbol{A_d}$, $\boldsymbol{B_d}$, and $\boldsymbol{C_d}$ are the discrete counterparts of $\boldsymbol{A_c}$, $\boldsymbol{B_c}$, and $\boldsymbol{C_c}$ determined by

$$\boldsymbol{A_d} = \mathrm{e}^{\boldsymbol{A_c}T}, \quad \boldsymbol{B_d} = \int_0^T \mathrm{e}^{\boldsymbol{A_c}\mu} \boldsymbol{B_c} \, d\mu = (\boldsymbol{A_d} - \boldsymbol{I})(\boldsymbol{A_c})^{-1}\boldsymbol{B_c}, \quad \boldsymbol{C_d} = \boldsymbol{C_c} \tag{39}$$

The key element in equations (39) is the **State Transition Matrix** $\mathrm{e}^{\boldsymbol{A_c}T}$. This is computed using equation[12] (40) either analytically as the inverse Laplace transform of the matrix $[(s\boldsymbol{I} - \boldsymbol{A_c})^{-1}]$, or numerically from the infinite sum:

$$\mathrm{e}^{\boldsymbol{A_c}t} \doteq \mathrm{L}^{-1}\{(s\boldsymbol{I} - \boldsymbol{A_c})^{-1}\} \doteq \boldsymbol{I} + \boldsymbol{A_c}\,t + \boldsymbol{A_c}^2 \frac{t^2}{2!} + \boldsymbol{A_c}^3 \frac{t^3}{3!} + \ldots \tag{40}$$

In order to compute the $\boldsymbol{K_d}$ gains the desired polynomial (37) must be coefficient matched with the discrete system closed loop characteristic equation:

$$\det [\, z\boldsymbol{I} - (\boldsymbol{A_d} - \boldsymbol{B_d}\boldsymbol{K_d}) \,] = 0 \qquad (41)$$

Again the discrete control $u(k)$ has the same form as its continuous counterpart in (31):

$$u(k) = gr(k) - [\, k_{d1}x_1(k) + k_{d2}x_2(k) \,] \qquad (42)$$

with $r(k)$, $x_1(k)$, $x_2(k)$ the discrete equivalent of the reference and state signals.

The discrete closed loop transfer function is given by equation (43):

$$H(z) = \frac{Y(z)}{R(z)} = \boldsymbol{C_d}[\, z\boldsymbol{I} - (\boldsymbol{A_d} - \boldsymbol{B_d}\boldsymbol{K_d}) \,]^{-1}\boldsymbol{B_d} \qquad (43)$$

The feedforward gain is calculated as before, using the discrete Final Value Theorem, the output $Y(z) = R(z)H(z)$, the transfer function $H(z)$ from (43), and the unit step $R(z) = \dfrac{R}{(1 - z^{-1})}$

$$y_{ss} = \lim_{k \to \infty} y(k) = \lim_{z \to 1} (1 - z^{-1})\, Y(z) = \lim_{z \to 1} (1 - z^{-1})\, R(z)H(z) = \lim_{z \to 1} (1 - z^{-1}) \frac{R}{(1 - z^{-1})} H(z) = RH(1)$$

The steady state error $e_{ss} = R - y_{ss}$ is required to be zero, thus $y_{ss} = R$, and from the above formulas we conclude that $H(1) = 1$, which determines the gain g. Remember that $H(1)$ is the $H(z)$ from (43), evaluated at $z = 1$, and $\boldsymbol{C_d} = \boldsymbol{C_c} = 1$.

EXPERIMENTAL PROCEDURE

PROBLEM FORMULATION

The objective of the experiment is to investigate the control of a process with transfer function:

$$G_p(s) = \frac{e^{-T_D s}}{(10s + 1)(2s + 1)}$$

in the presence of a step input of size R (default value $R = 1$ volt).

The experiment is run on a Comdyna analog computer serving as the process simulator, and a PC serving as the digital controller. The analog computer's ability to produce an exact replica of a system's dynamic model, running in real time, combined with the digital computer's ability to perform complex calculations and make diverse decisions with speed and accuracy, result in a highly flexible hybrid system that can handle a wide variety of problems with minimum programming effort.

The process will be implemented on the Comdyna analog computer using the analog computer simulation diagram of FIG. 11 for the PID and FST part, and FIG. 12 for the State Feedback part.

The student is urged to verify that the two plant analog simulation diagrams represent the same transfer function. There are 2 separate analog computers available for the lab. One is configured for Parts A, B, and C while the second is configured for Part D. Before dismantling a wired Comdyna, check to see if a correctly wired one is available first to save yourself a lot of time.

The PID, FST and State Feedback controllers will be designed using the methods presented previously, and implemented using the PC program. The experiment will include investigation of the following topics:

1) Comparison of the PID controllers designed by the various schemes.
2) Study of the effect of increased sample time on control.
3) Study of the effects of time delay.
4) Study of the FST controller especially when using very long sampling times.
5) Study of the State Feedback controller both in the continuous and the discrete domain.
6) Comparison between the PID, FST and State Feedback controllers both at the transient and at the steady state level, with long sample times, and with time delays.
7) Steady state error[5] analysis.

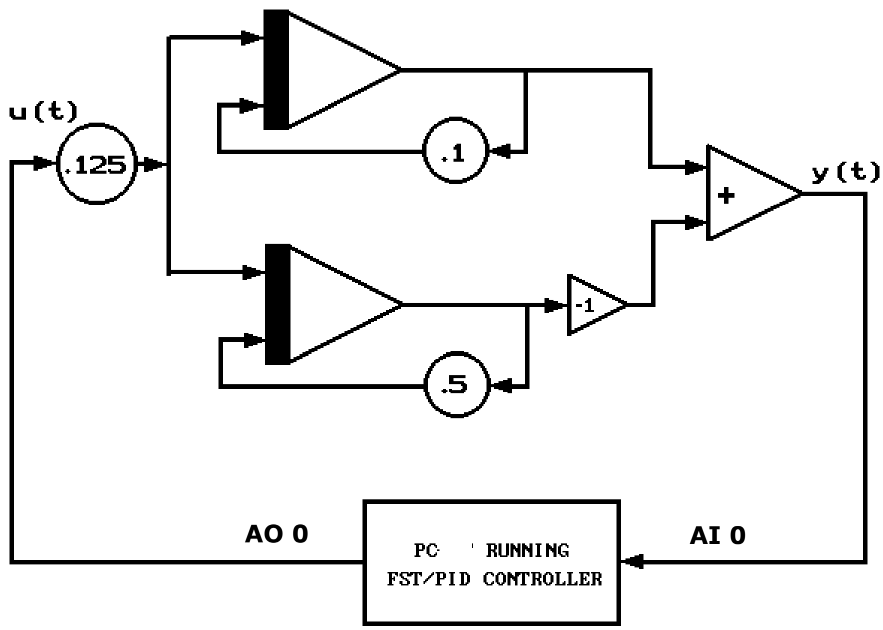


FIGURE 11. Analog process simulation for PID and FST controllers.

The theory covered in the previous topics is so broad that the instructions have to be as general as possible. It is hoped that initiative will be used in the study of the various controllers. If an area seems interesting, spend time exploring it.

HARDWARE SETUP – COMPUTER USAGE

The hardware to be used consists of the Comdyna analog computer implementing the 2[nd] order process, the PC running the **Hybrid Controller DAQmx+.vi** feedback control program, a digital

[5] For the closed loop system shown in FIG. 2 the transient and steady state error are:

$$e(s) = \frac{R(s)}{1 + G_p(s)G_c(s)}, \quad e_{ss} = \lim_{s \to 0} \frac{sR(s)}{1 + G_p(s)G_c(s)}$$

storage oscilloscope (DSO) to record the control signal and the states or the process response, and a set of cables.

The parallel realization of FIG. 11 is to be built for the PID and FST cases. The process output y is to be connected to the PC by BNC-2110 input port **AI 0**, and the process input u to the PC BNC-2110 output port **AO 0**. The cascaded realization of FIG. 12 is to be built for the State Feedback part. State x_1 must be connected to the PC input port **AI 0**, state x_2 to the PC input port **AI 1**, and the system input u to the PC output port **AO 0**. In either case all unused ports are to be left free. It is suggested that the Comdyna manual is at least browsed before starting the experiment. Special care must be taken when implementing the gains because of the sign inversions at the output of the amplifiers. The Comdyna dial must be on the **Pot Set** position during setup; during operation the pushbuttons must be switched back and forth between the **OP** (operate) and the **IC** (reset initial conditions) states, at the start and end of each run. The DSO must be thoroughly understood and its various input scaling options mastered, before any useful work can be done. One of the DSO channels must be connected to the control output (**AO 0**) and the other to the process output (**AI 0**). Use the maximum voltage range possible for more detailed results. Scale settings on the DSO frequently should be checked periodically.

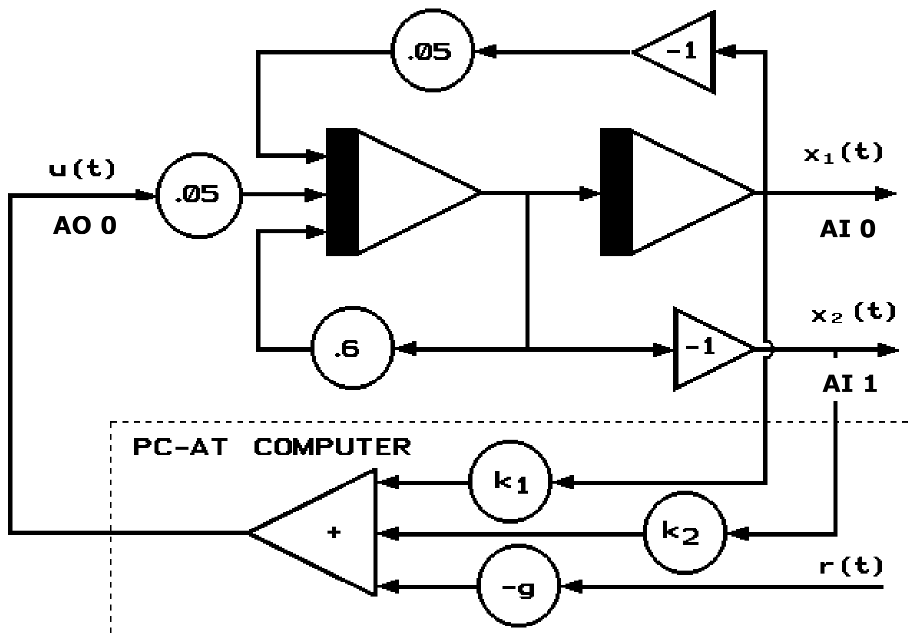


FIGURE 12. Analog Process simulation for pole placement controller.

To access the LabVIEW program do the following:

- Turn the PC on (if off) and go the **HYBRID** subdirectory (**My Computer\Local Disk**): **C:\CStudio\RTA_lab\Hybrid**.
- Double click **Hybrid Controller DAQmx+.vi** to load the program. A LabVIEW program will open with six different tabs at the top of the screen. When moving through the parameter fields, the tab button will not work. You must use the mouse to manually select them.
- Be sure to push the operate button (**OP**) on the analog computer before starting the VI program, otherwise the controller will not behave properly. Push **IC** when done to reset the plant for the next run.
- Press the right arrow button at the top left-hand corner of the screen in order to start execution. In order to pause; press the **STOP** button on the screen, **not the stop sign at the top-left of the page (this will help to calculate actual sampling times, and reset the output to 0**

Volts). It may take a few seconds for the program to respond.

- Although some values can be changed during execution by user input, it is important, in order to properly ensure correct measurement, to stop the controller (by using the **STOP** button) before altering parameters values as necessary.
- If you notice the controller isn't working properly:
  - o Press the stop button then run the LabVIEW program. This should reset the program and enable you to start from scratch.
  - o Wiggle the T-connectors; make sure the connection is good. If when wiggling you notice a difference in the response, change connectors. Occasionally the PC may need to be restarted.

In almost all the cases data entry consists of the sampling period T in ms, the reference step size R and the precalculated coefficients/gains that are usually a function of T. Even though a software implemented **clipper** prevents the control output from "wrapping" around by forcing it to stay within its respective **max/min** values, there are cases when the signal remains at these levels very long (**saturated**) the results will be erroneous. It is suggested that R be reduced until the maximum non-saturating value is found.

A suggested reference input R value is 1 volt, even though other acceptable values can be used. Sampling time T depends on the algorithm used to derive the gains, and for the continuous implementations the general notion is the faster the sampling, the closer the computer controller resembles the analog model (10 ms or less).

The time delay is $T_D = MT$ where T is the sampling time and M is the number of periods.

IMPLEMENTATION – INVESTIGATION

A. Step Response Mode (Uncontrolled Open Loop)

This mode is used to observe the open loop step response of a process under the "**Apply Step Inpu**t" tab. The computer outputs -on **AO 0**- a step of magnitude R defined by the user. It must be run first, as it is used for the calculating the L and R coefficients for the Ziegler-Nichols PID controller. The sampling time T is currently of no use (placeholder for the future) even though a value must be entered. Several runs are suggested, using different R and settings on the time and voltage ranges, until the most detailed curve is produced.

B. PID Control Mode

This mode implements the PID controller given in equation (3). The "**PID Controller**" tab contains fields for Kp (proportional gain), Ki (integral gain), Kd (derivative gain), R (reference input) and T (sampling time in ms). The computer outputs -on **AO 0**- a step of magnitude R defined by the user. The computer inputs -on **AI 0** (y(t))-, the response of the system. Remember to press the operate mode on the Analog Computer before beginning execution of the Step Response. It is also important to remember to end execution of the program by pressing the **STOP** button on the screen. Since all variations of the controller are to be investigated, it's strongly advised that the sets of coefficients be calculated ahead of time for various sampling and delay times (wherever applicable). More specifically the equations between (4) and (5) will be used for the Ziegler-Nichols; equations (6), (8), and (9) for the Gallier-Otto method (use several delay times with values comparable to the process time constant); and TABLE 1 will be used for the Graham-Lathrop forms.

It is suggested that the pure proportional (P) control be implemented first, so that a feel for the scaling and the process response is acquired. After several runs using different sampling times, vary the controller gain and observe the results in the control effectiveness. It is very important to have precalculated the Routh-Hurwitz[13] test stability test for the closed loop characteristic equation, so that the stability limits of the system can be investigated.

Progressively start implementing the other PID variations using a standard set of sampling periods. Compare the various PID designs (Ziegler-Nichols, Gallier-Otto, and Graham-Lathrop) for the system with no time delays. The Graham-Lathrop PI and PID controllers should be studied for several values of the natural frequency ω_0 and the best controller should be selected. Comparisons should be based on

1) Rise time.
2) Overshoot.
3) Settling time.
4) Steady state error.

Select several of the controller designs that give good control. Increase the sample time and observe the effects in the control. Using a "good" set for the controller gains as obtained from the previous steps, start adding time delays to the system and observe it's effect on the control quality. If during any run the controller output seems to saturate at either of the extreme points (±10 volts), rerun the case using a smaller step input. Note that the derivative term of the control signal is calculated by dividing by T, the sampling period. For small T (~1 ms) 1/T is 1000 and multiplying the quantized slightly noisy voltages by such a large value results in a control signal that may rapidly jump above and below an average value by a volt or more. This will be visible on the scope. It may be reduced by increasing T, thus reducing the large multiplying factor.

C. FST Control Mode

This mode implements the FST controller given in equation (25), using the definitions of equation (20). The computer outputs -on **AO 0 (u(t))-** a step of magnitude R defined by the user. The computer inputs -on **AI 0 (y(t))-** the response of the system. The "**FST Controller**" tab contains fields for R (reference input) and T (sampling time in ms), control of error terms Ke0, Ke1, Ke2, the coefficients for the control history terms Ku1, Ku2, and the delay factor M. The routine assumes that a sampled data version of the process is used, and the sampling time is sufficiently large (> 500 ms) to ensure non-saturated control. The FST controller with delay tab has provision for long delays as an integral number of sample periods and can be disabled by entering zero in the relevant field. The FST controller (without delay) is wired to the analog computer the same way as the PID controller with the control output from **AO 0** goes to the analog plant input and the plant output going to **AI 0**. **AO 0** and **AI 0** signals are to be displayed on the scope and recorded. Due to the way the delay is simulated in LabVIEW the wiring must be modified to observe the responses for the FST controllers with plant delays. For these runs, **AO 0** and **AI 0** are still connected to the plant input and output respectively, but the control signal to be observed and recorded on the scope comes from **AO 1**, which is compared to the same plant output on **AI 0**.

The response of the process to FST control for several sample periods should be recorded and analyzed. The effects of time delay should be investigated. The above responses should be compared to the responses of similar systems with PID control. How does the steady state error in the FST case compare to those of the PID variations?

D. State Feedback Control Mode

This mode implements the State feedback controller given in equations (31) and (42). The "**Pole Placement Controller**" tab contains fields for R (reference input) and T (sampling time in ms), the state feedback k1, k2, (both continuous and discrete cases) and the feedforward gain g. The computer outputs -on **AO 0 (u(t))-** a step of magnitude R defined by the user. LabVIEW reads from **AI 0 (x_1(t)),** and **AI 1 (x_2(t))** the different states of the system. In this part of the experiment, the student will calculate gains of the state feedback controller for the second order plant given, by using the pole placement method. The closed loop process response requirements are an underdamped unit step response, with damping ratio $\zeta = 0.707$, and settling time $T_s \leq 4$ seconds. The closed loop poles are to be calculated using the above specifications.

For small sampling time the continuous domain equations should be used. The desired polynomial is to be constructed according to (29). Before proceeding the $\boldsymbol{A_c}$, $\boldsymbol{B_c}$ and $\boldsymbol{C_c}$ must be correctly identified. The student is urged to spend ample time matching the block diagram of FIG. 10 with the simulation diagram of FIG. 12 and the process transfer function given. When done, the closed loop characteristic equation is to be evaluated from (33) and the gains $\boldsymbol{K_c}$ found as explained on the relevant section. Equation (35) is to be used for the feedforward gain g. For the actual runs start with very small sampling time (≤ 10 ms) and progressively increase it, recording the effects in the system response.

For large values of sampling time the discrete analysis should be used. First evaluate the desired characteristic polynomial from equations (36) and (37). The state transition matrix must then be evaluated either analytically or through your favorite math package (e.g. MATLAB). Apply equations (39) to calculate $\boldsymbol{A_d}$, $\boldsymbol{B_d}$ and $\boldsymbol{C_d}$. Again the closed loop characteristic equation is to be found from (41), and by coefficient matching, the gains $\boldsymbol{K_d}$. The discrete transfer function (43) must next be found and evaluated at $z = 1$, thus specifying the feedforward gain g. Since most of the above variables depend on T (used in seconds here rather than milliseconds) and several runs with different sampling periods are expected, it is suggested that their values be computed using a spreadsheet program or any other similar package.

Be warned that since the control algorithm is exactly the same in both the continuous and the discrete cases, it is the students' responsibility to calculate the gains using the formulas suitable for each case. In general the scaling factor W will be set at 1. Also try running both cases with $g = 1$ (unity feedforward gain), and compare the experimental steady state error with the theoretical one.

E. Further Studies

As an extension to the previously described experiments, (time permitting) you may try the following:

1) Study the effects of a disturbance input to the system. A disturbance input can be simulated by adding a small voltage to some point in the analog computer simulation diagram.
2) Change the transfer function of the process. Investigate poles close to the imaginary axis. Use your initiative to try as many combinations and pole placements as time may permit.
3) Most controllers are not designed with time delays in mind. Could you suggest any modifications that would add this capability into their formulas.

WRITE UP ANALYSIS – CONCLUSIONS

The write-up should include the following:

1) Analytical calculations for the various PID controllers. Printouts of all software-calculated variables.
2) For the FST controller, the difference equations for the process should be derived and one set of controller gains calculated.
3) For the Pole Placement controllers, detailed calculations of the transfer functions, characteristic equations and gains. For the discrete part tables with the values of all variables as functions of the sampling T.
3) A list of all the runs, including comments on the individual runs. All saved DSO screen shots must contain scaling info and be labeled with the controller name and coefficient set. A subject

numbering scheme (e.g. P 1, G-O PID 1, etc) can prove to be very helpful.

4) An analysis of the assigned areas of study, and any other observations that may seem interesting. DSO screen shots should be used to illustrate your results at every stage.

5) A net conclusion giving in a table the *pros* and *cons* of each design. This should serve as a selection guide to a "design engineer" as to what controller he should use under his specs for sampling, delays, overshooting, etc.

REFERENCES

In addition to context specific references listed below with their relevant numbers on the left, the first two entries contain vital general theory covering to the whole area of this experiment.

DeRusso, P. M., Roy, R. J., and Close, C. M., *State Variables for Engineers*, John Wiley and Sons, New York 1965. Chapter 3 for z-transforms and Chapter 6 for discrete time systems.

Dorf, R. C., *Modern Control Systems*, Addison Wesley, Reading, Mass. 1974. Great for *s*-domain feedback design applications.

[1] **Cadzow, J. A. and Martens, H.R.**, *Discrete Time and Computer Control Systems*, Prentice Hall, 1970. Covers theory and techniques of discrete time systems, computer control systems, computer simulation and digital data systems.

[2] **Takahashi, Y., Rabins, M. J., and Auslander, D. M.**, *Control and Dynamic Systems*, Addison Wesley, 1970, pp. 343-344.

[3] **Gallier, P. W. and Otto, R. E.**, *Self tuning Computer Adopts DDC Algorithms*; Progress in Direct Digital Control, ed. T. J. Williams, F. M. Ryan, ISA, Pittsburg, 1969.

[4] **Graham, D. and Lathrop, R. C.**, *The Synthesis of Optimum Transient Response: Criteria and Standard Forms*, Trans. AIEE, Nov. 1953, pp. 273-288.

[5] **Ragazzini, J. R. and Franklin, G. F.**, *Sampled Data Control System*, McGraw Hill, 1958, pp. 145-198

[6] **Kailath, T.**, *Linear Systems*, Prentice Hall, NJ, 1980, pp. 31-35.

[7] **Frederick, D. K. and Carlson, A. B.**, *Linear Systems in Communication and Control*, John Wiley & Sons, 1971, pp. 83-85.

[8] **Frederick, D. K. and Carlson, A. B.**, *Linear Systems in Communication and Control*, John Wiley & Sons, 1971, pp. 79-88.

[9] **Kailath, T.**, *Linear Systems*, Prentice Hall, NJ, 1980, pp. 197-198.

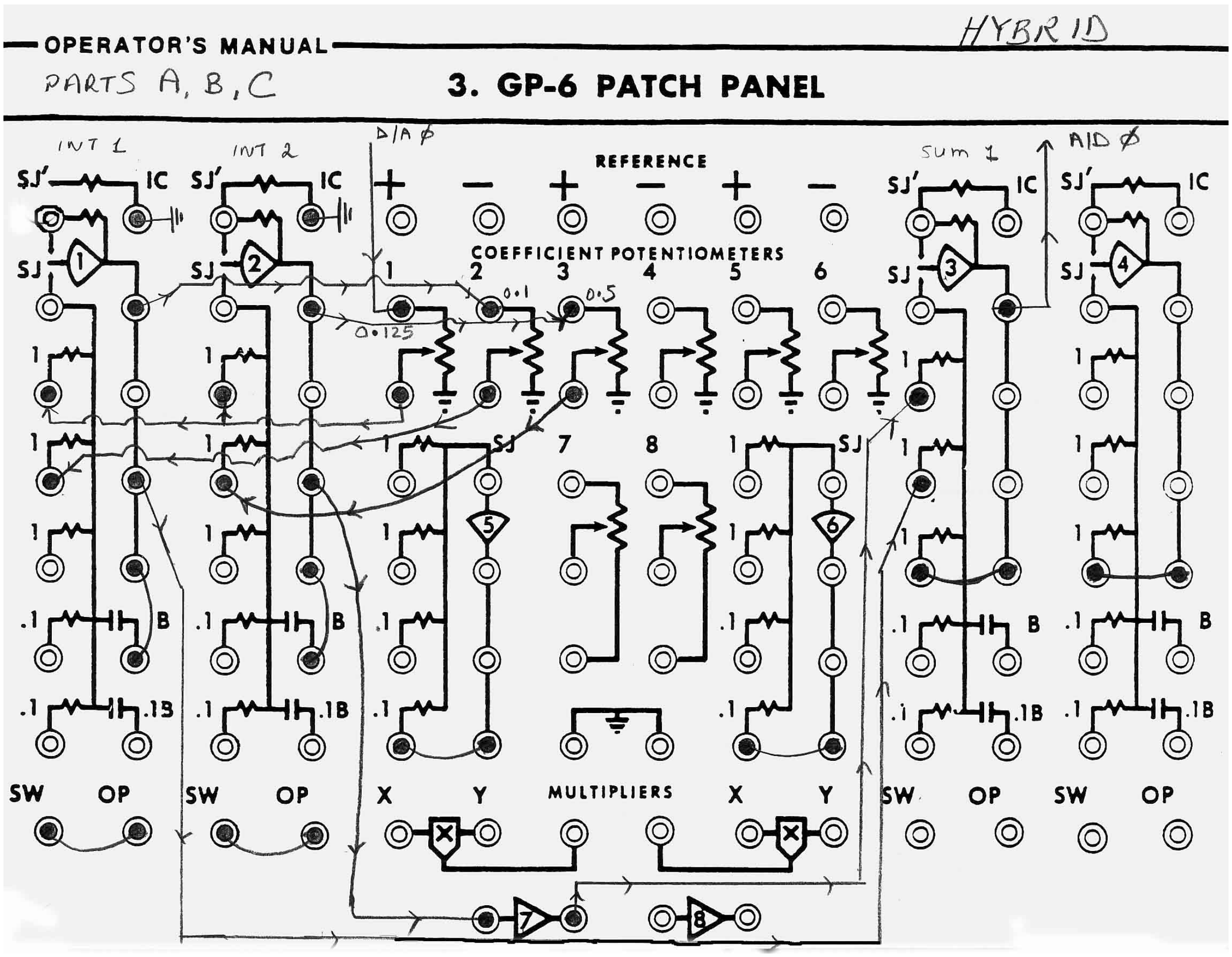
[10] **Swisher, J. M.**, *Introduction to Linear Systems Analysis*, Matrix Publishers Inc., Champaign, Il., Ch 10.

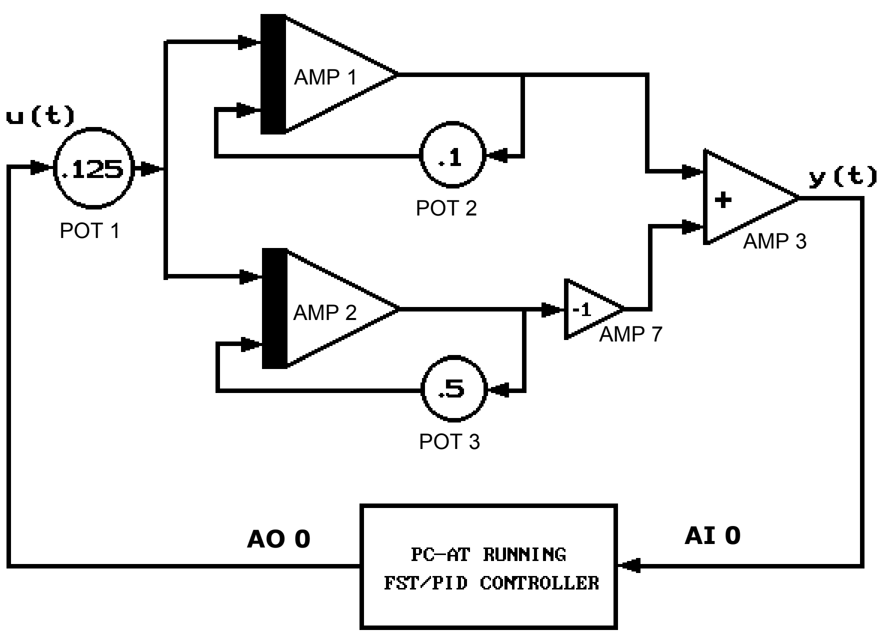
[11] **Kailath, T.**, *Linear Systems*, Prentice Hall, NJ, 1980, p 174

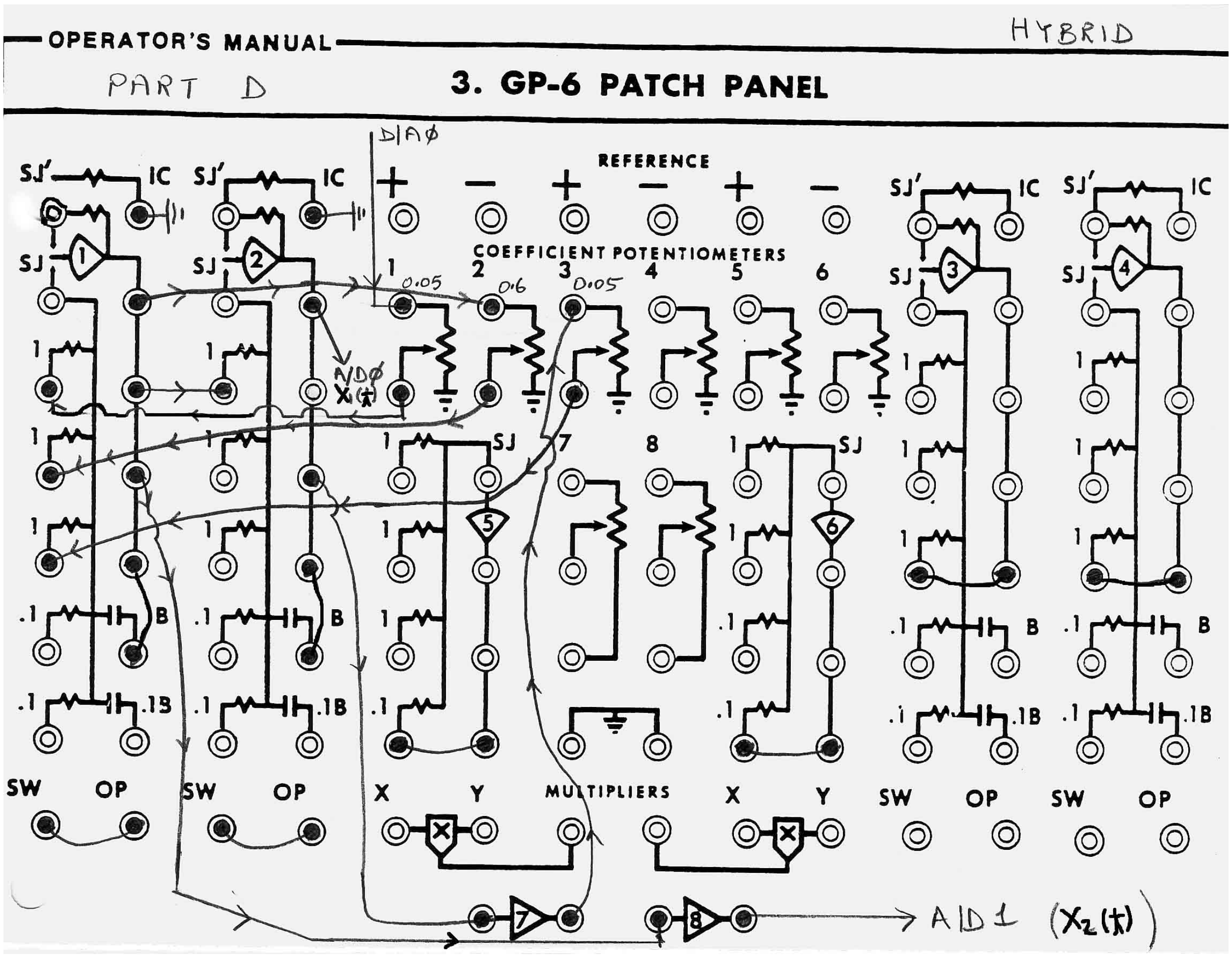
[12] **Frederick, D. K. and Carlson, A. B.**, *Linear Systems in Communication and Control*, John Wiley & Sons, 1971, pp. 295-298.

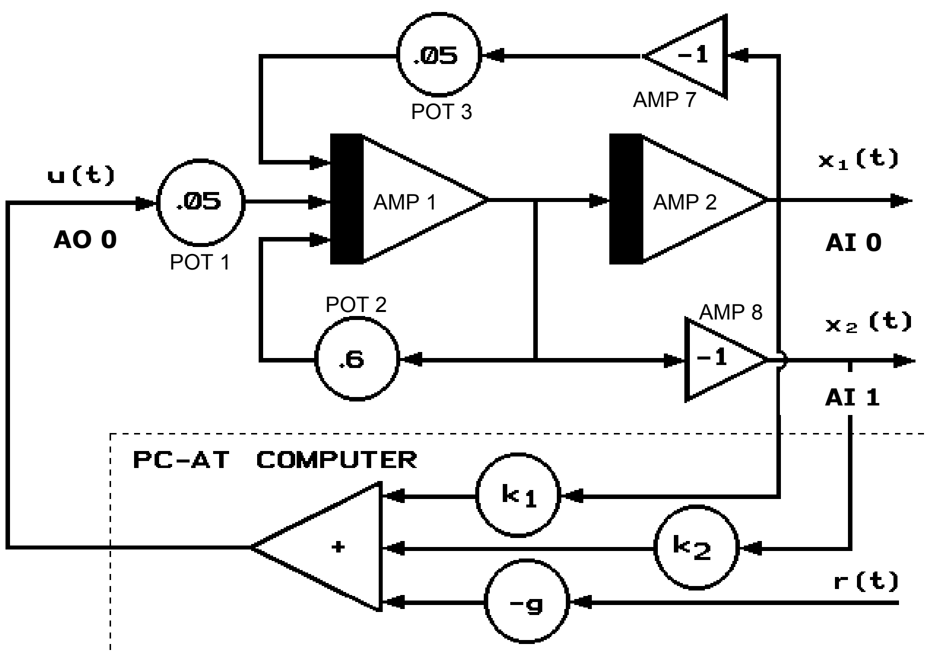
[13] **Frederick, D. K. and Carlson, A. B.**, *Linear Systems in Communication and Control*, John Wiley & Sons, 1971, pp. 385-388.

APPENDIX - ANALOG COMPUTER WIRING DIAGRAMS









See OPTIMAL CONTROL experiment analog closed loop wiring diagram for pure **analog controller**.