1993

Design of Control Systems with Respect to Constrained Actuators

Matthew Aitkenhead
University of Rhode Island

DESIGN OF CONTROL SYSTEMS WITH RESPECT TO CONSTRAINED ACTUATORS

BY

MATTHEW AITKENHEAD

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING AND APPLIED MECHANICS

UNIVERSITY OF RHODE ISLAND

1993

MASTER OF SCIENCE THESIS

OF

MATTHEW WILLIAMS AITKENHEAD

APPROVED:

Thesis Committee

Major Professor William J. Palm

Charles D. Nash

R. J. Vaccaro

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

1993

ABSTRACT

This paper presents a method for the design of control systems such that actuator performance limits are not exceeded. The maximum energy delivery concept and root locus analysis methods were used to find the gains for a pseudo-derivative feedback controller for a second order system with zero or first order numerator dynamics. The method has been implemented in a computer program which determines the gains and simulates response characteristics.

ACKNOWLEDGEMENTS

I would like to express my appreciation to Dr. Bill Palm whose careful attention and patience continually motivated me to complete this research. Also, I wish to thank my beloved wife, Margie, and my parents, for their limitless support.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Statement of Problem

In practice, control system design often involves expensive trial and error testing in order to design a controller which satisfies a certain set of criteria. Control system performance is ultimately dependent upon the physical limitations of the controlled system, a concept overlooked by most academic approaches to control system design. This paper presents a method to design control systems considering the physical constraints of the final control elements (actuator) of the system to be controlled. The method has been implemented in the form of a personal computer program which determines the gains for a Pseudo Derivative Feedback (PDF) controller for an adjustable configuration physical system. The system must be modeled as a transfer function with an order of two in the denominator and an order not exceeding one in the numerator. (The gains for the first order denominator configuration can be solved for analytically by direct substitution into equations derived by Phelan (Phelan [1], pp. 152-155).

1.2 Motivation

Few authors are concerned with the design of controllers for systems with actuator limits. Of the authors who do consider this limitation, few provide a complete solution in a form useful for a modern control system designer. Phelan describes the actuator limit concept as the single most important point in control system design

(Phelan p.11). The general method presented by Phelan for solving such problems provides an analytical solution for a first order control system and a suggested trial and error procedure for second order systems. Other constraints on the configuration of the problem examined by Phelan include only an inertia term in the plant transfer function and no numerator dynamics (no derivatives in the transfer function numerator). The motivation of this investigation was therefore to expand the complexity of the problems to which Phelan's methods (or variations of) could be applied. As a consequence of the incorporation of the methods in a personal computer program, a tool has been developed which is convenient for a control designer to use.

1.3 Terminology

The following terms and definitions relate to the block diagram shown in Figure 1-1.

Control system: Any system that controls a supply of energy.

Feedback control system: A control system which uses measurement of the output or controlled variable to help adjust the supply of energy in the system.

Controller: The portion of the control system which encompasses the adjustable parameters which influence how the system responds.

Fixed Elements: The portion of the control system which is not adjustable. The two subsets of the fixed elements are:

> Actuator: Accepts a low power level command from the controller and converts it to a high power level.

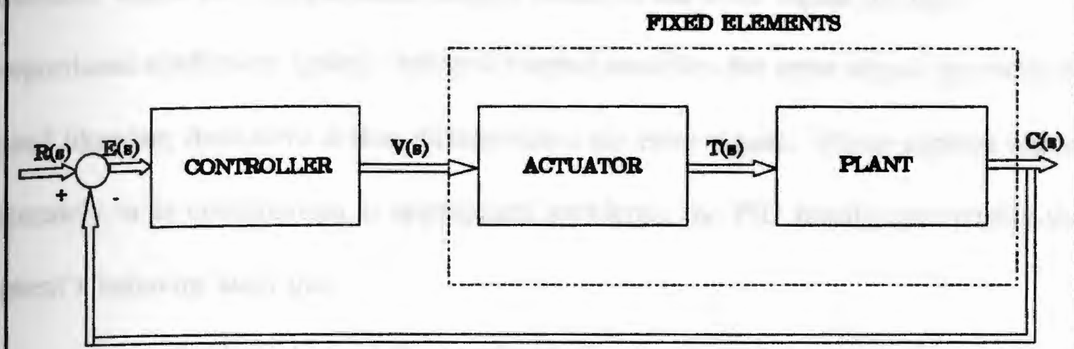

Figure 1-1 Typical Feedback Control System

Plant: The object to be controlled, such as the mechanical load attached to a D.C. electric motor.

Command (Reference) Input, R: The signal or action that is requested of the control system. Often the most severe command input that can be requested of a system is a step input, which is an instantaneous change from no energy to some maximum value.

Output, C: The desired signal or action as a result of the control process.

Error Signal, E: The difference between the command signal and the output signal.

Controller Signal, V: The signal following the controller in the block diagram.

Actuator Signal, T: The signal following the actuator in the block diagram.

1.4 Summary of the PID family of control laws

The most traditional class of controllers are the proportional (P), integral (I) and derivative (D) controllers and various combinations thereof (Phelan p.70). A controller which uses proportional control modifies the error signal, E, by a proportional coefficient (gain). Integral control modifies the error signal by integrating it and likewise, derivative action differentiates the error signal. When applied either separately or in combination to appropriate problems, the PID family can modify the system's behavior such that

a. steady state error (error signal after transient behavior has disappeared) is minimized

b. overall system response time is minimized

c. transient specifications, such as maximum overshoot of the output signal are minimized (Palm, [2] p. 335).

In addition to these desirable characteristics of PID controllers, there are some undesirable ones. Foremost is that the controller which uses some combination of PID (which is more likely than any one action by itself) may simultaneously modify conflicting signals, resulting in possibly un-predicted controller performance. Systems designed with the PID family of controllers should therefore never operate on more than one signal in the forward path of the controller. This concept is sometimes referred to as the principle of one master (Phelan p. 150).

2. DESIGN BASED UPON CONSTRAINED ACTUATORS

Phelan makes the following statement about control system design: "Two kinds of automatic control systems - academic and real exist, and they have almost nothing in common." (Phelan p.11) While many aspects of control system design can be understood using basic controller theory, these methods will only be accurate if the system responds linearly. Unfortunately, linearity is not guaranteed unless the physical limitations of the real actuator are taken into account in the determination of controller gains.

2.1 Non-linearity in Control Systems

The equations of motion describing the dynamics of every real controlled system are non-linear. Since the mathematical analysis of non-linear systems is much more complicated than linear systems, it is advantageous to simplify the equations of motion so that they are linear. Fortunately, the fundamental idea behind a feedback control system - the comparing of the actual output to the desired output, makes real (vs. academic or theoretical) control systems inherently very tolerant of most non-linearities, provided they are designed properly.

2.2 Non-linearity produced by Actuator Saturation

There are many types of controllers, each of which can provide a wide variety of response characteristics to the signal upon which it operates, the error signal. Most academic lessons in control system design discuss these control methods and provide examples of their use with every day problems. There are few examples of actual response data in the literature. There may be situations where the same control strategies are used in a real design problem, and the resulting response does not react as expected.

The problem is that while the actuator is the "muscle" of the control system, it is also the weakest link. The actuator is part of the fixed elements of the control system and therefore is not easily adjustable. Examples of this inflexibility include:

1. A D.C. motor has a limited torque which it can produce - either deliberately so as to prevent damage to the motors components or accidentally, such as due to improper selection of an amplifier.

2. A valve cannot be more than fully open or fully closed in a liquid level controller or pressure control system.

If a control system operates over a wide range of conditions, it is possible that the output of the controller, V, may request more energy from the actuator than it is capable of delivering. When this maximum value is exceeded, the feedback loop is effectively broken because while the control signal is requesting more energy, the actuator will produce only what it is limited to. When the actuator is at its limit, it is said to be saturated. Some other types of non-linearities include dead-zone, bang-

bang, hysteresis and mechanical backlash (Towill, [3] p. 411). In some of these cases, the non-linearity is actually deliberately produced to improve system performance (Towill p. 415).

A common result of this saturation is called reset-windup. This occurs in controllers which use the integral of the error signal to control the process. The value of the control signal for the integral control algorithm is:

$$v(t) = K_1 \int e(t)\,dt \tag{2-1}$$

The value of V is dependent both upon the magnitude of the error signal and the length of time the error exists. For a step input, the integral term increases rapidly until the actuator saturates and the response overshoots its desired level. The saturation would occur even sooner if proportional control of the error signal were also used (PI control) because the error is at its maximum value just after time zero. After the response overshoots the set-point value, and the error changes sign, it takes some time before the error is large enough to cancel out the overshoot. Consequently, the actuator signal can not pull away from its saturation limit and the system behaves non-linearly. The result is that any controller with integral action may have significant overshoot and a longer response time than it would have if the actuator signal did not saturate.

The neglect of the finite energy delivering capability of actuators is the primary reason academic control systems are so different from real ones. Many manufacturers of control equipment use academic methods on real-world systems. As a result their equipment falls short of expected performance which would then require a set of tuning procedures to bring the performance in line. (Phelan pp. 66-67).

There are several methods that have been developed which consider actuator saturation and its effect on overall system performance. A controller which uses Anti-Windup (Astrom [4], p.12) has an extra feedback path which measures the actuator signal as a means to prevent saturation. More recently, a numerical method was developed which determines linear controller designs based upon convex optimization techniques (Boyd, et al [5]). The maximum energy delivery concept was developed by Phelan, and is described further in Section 2.3.

2.3 The General Method

Actuator saturation and non-linear response can be prevented by simply designing the controller (that is select the control gains for the control scheme) such that the control signal never requires the actuator to saturate. This will require the designer to know three types of information about the system to be controlled:

1. The coefficients of the parameters of the fixed elements of the system. For a second order actuator/plant pair, this would be inertia, damping and restoring terms.

2. The actuator saturation limit.

3. A maximum operating condition, such as the maximum speed at which a D.C. motor is expected to operate.

The crucial information is item 2, and is also the most difficult to obtain. The difficulty is that so little emphasis has been placed on actuator limits in the past that data is rarely available on this parameter. The designer may be required to derive this limit from some maximum operating condition of the system. The characteristics of an ideal actuator with a finite limit on the output, T, is shown in Figure 2-1.

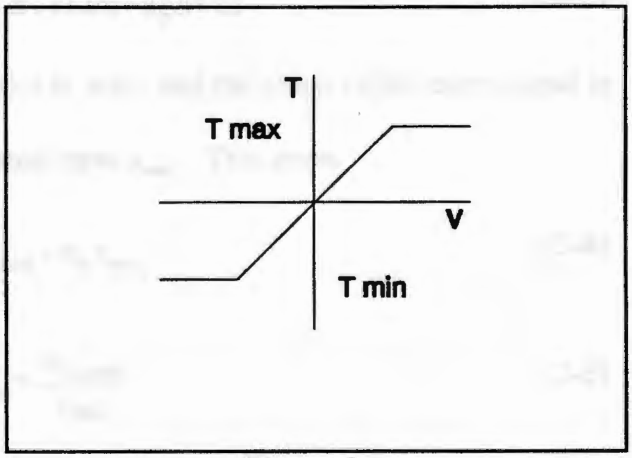

Figure 2-1
Piecewise Linear Actuator Function

While as a whole, Figure 2-1 does not describe a linear function, it is piece-wise linear. That is, for certain ranges of the input V, the function is linear. In terms of the problem to be solved, linear operation occurs when V remains in the region such that

$$T_{min} < T < T_{max} \tag{2-2}$$

The minimum and maximum values of T represent the limits on the actuator and may or may not be equal in magnitude. The values of T_{min} and T_{max} are entirely dependent upon the system requirements and hardware limitations.

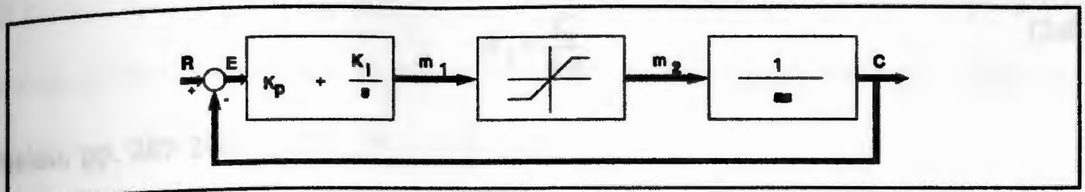

Figure 2-2 PI control of a saturating first order system.

As an example, consider PI control of the first order plant shown in Figure 2-2, where m_1 and m_2 are the inputs and outputs of the piece-wide linear actuator function respectively. When the actuator function is operating in the linear region,

$$m_2 = m_1 = K_i \int e(t)\,dt + K_p e(t) \tag{2-3}$$

At time t=0, the integral of the error signal is zero, and the value of the error signal is at its maximum - the magnitude of the step input r_{max}. This gives

$$m_{2,max} = K_p r_{max} \tag{2-4}$$

$$K_p = \frac{m_{2,max}}{r_{max}} \tag{2-5}$$

where $m_{2,max}$ is the saturation limit of the actuator function. The characteristic equation of the system in Figure 2-2 is:

$$as^2 + K_p s + K_i = 0 \tag{2-6}$$

The standard formula for damping ratio for this second order equation is:

$$\zeta = \frac{K_p}{2\sqrt{aK_i}} \tag{2-7}$$

Assuming the most desirable response characteristics will be achieved when the system is critically damped (ζ=1) K_i can be determined as such:

$$K_i = \frac{K_p^2}{4a} \qquad \text{(2-8)}$$

(Phelan, pp. 267-278).

2.4 Assumptions

2.4.1 Linearity

The differential equation which describes the fixed elements of the control system is assumed to be linear. The transfer functions describing the control system in its entirety are also assumed to be linear, provided that the actuator is prevented from saturating.

2.4.2 Absence of Disturbance Terms

Sometimes random forces and/or deviations (non-linearities) in the parameters of the plant transfer function create a random input preceding the plant in Figure 1. For the purposes of this study, disturbances were neglected because their maximum magnitude is difficult to predict, and thus an estimate can not be made on whether they will produce actuator saturation or not. It is assumed that disturbances are second-order effects that don't cause actuator saturation.

2.4.3 Step Functions

The most severe, and therefore most useful, type of reference input is a step function (Phelan p.96). A step change in command input represents an instantaneous, non-continuous change. No real-world system can respond as such for this would require an infinite amount of energy at time 0. For problems modeled with transfer functions having numerator dynamics, a pure step input is unrealistic. Therefore, a replacement function is used to represent the step function:

$$x_r(t) = M(1-e^{-zt}) \tag{2-9}$$

where M is the magnitude of the step and z is a constant dependent upon the smallest time constant (τ_{min}) of the control system. The constant is arbitrarily chosen such that

$$z \geq \frac{10}{\tau_{min}} \tag{2-10}$$

2.5 Method applied to a second order fixed element system

The block diagram in Figure 2-3 shows pseudo-derivative-feedback (PDF) control for a second order system with numerator dynamics. PDF control was developed by Phelan (Dec. 1970) as a solution to the problems associated with the principle of one master. In an effort to avoid the undesirable effects of differentiating the error signal, the output of the control system is fed back into the forward path of the loop following an integral, I action, control block. The overall effect of this configuration would be the same as if the output signal were differentiated and fed back preceding the integral block. I action is chosen over P action in the forward path because it is often unrealistic to expect instantaneous response to a step input as is the case for P action and because I action gives zero steady-state error (Palm p. 417). Note that for the second order plant in Figure 2-3, there are two PDF gains, operating on the output signal, and the first derivative of the output signal. Also note there are two proportional gains K_A and K_B which are included in the figure to account for

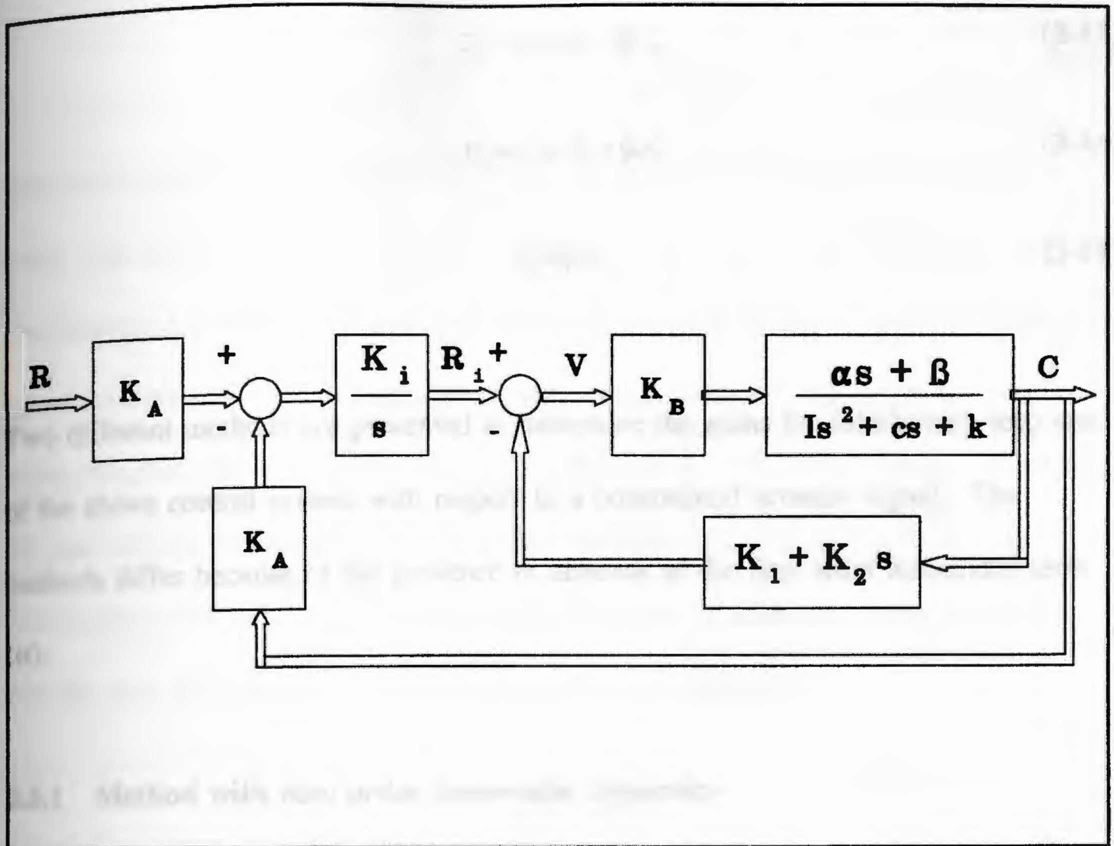

Figure 2-3 PDF Control of General Fixed Element System

miscellaneous proportionality factors common in control systems (such as amplifiers and potentiometer/tachometer gains). They do not affect the dynamics of the system, only the magnitude of the PDF gains, and therefore will be neglected in the following derivations. The overall system transfer function for the system in Figure 2-3 is

$$\frac{C(s)}{R(s)}=\frac{K_1(\alpha s+\beta)}{T_3 s^3+T_2 s^2+T_1 s+T_0} \tag{2-11}$$

$$T_3=I+\alpha K_2 \tag{2-12}$$

$$T_2 = c + \alpha K_1 + \beta K_2 \tag{2-13}$$

$$T_1 = k + \alpha K_i + \beta K_1 \tag{2-14}$$

$$T_0 = \beta K_i \tag{2-15}$$

Two different methods are presented to determine the gains for satisfactory response of the above control system with respect to a constrained actuator signal. The methods differ because of the presence or absence of the first order numerator term (α).

2.5.1 Method with zero order numerator dynamics

The characteristic equation is a third order differential equation, therefore standard formulas for damping ratio and time constant for a second order characteristic equation are of no use in determining the gains which will accomplish the goal. The fact that the system in Figure 2-2 is a multiple loop system will be useful however. The inner loop transfer function is:

$$\frac{C(s)}{R_i(s)} = \frac{\beta}{Is^2 + (c + \beta K_2)s + k + \beta K_1} \tag{2-16}$$

The characteristic equation is a second order differential equation from which the following equation for damping ratio is found (where IL refers to inner-loop):

$$\zeta_{IL}=\frac{c+\beta K_2}{2\sqrt{(I)(k+\beta K_1)}} \tag{2-17}$$

One would expect that the optimum values of K_2 and K_1 would be found when the inner loop damping ratio is 1 because critically damped systems often have desirable characteristics (fast, smooth response curves). However because $R_i(s)$ would never be so severe as a step function because of its position following the controller in the block diagram, the value of ζ_{IL} can be less than unity. Studies by Phelan and Ulsoy [6] have shown that the optimum value of the inner loop damping ratio for smooth fast response is 0.7 (Phelan, pp. 219-225). Through simulations, Phelan determined that the best relationship of K_1 to v_{max} and r_{max} came out to be

$$K_1=8\frac{V_{max}}{r_{max}} \tag{2-18}$$

From Equation (2-17):

$$K_2=\frac{1.4\sqrt{I(k+\beta K_1)}-c}{\beta} \tag{2-19}$$

Phelan states that there is no simple way to determine the gain K_i, analytically. He suggests a trial and error procedure of starting with a low value of K_i and gradually increasing it while providing step changes in the reference input equal to the maximum value expected. At each trial the value of K_i is increased until either the actuator saturates or the output response overshoots.

2.5.2 Method with first order numerator dynamics

If the method used by Phelan were to be applied to a fixed element system with first order numerator dynamics, the first step again would be to find the gains K_2 and K_1 to provide an inner-loop damping ratio of 0.707, where

$$\zeta_{il}=\frac{c+\alpha K_1+\beta K_2}{2\sqrt{(I+\alpha K_2)\,(k+\beta K_1)}} \qquad \textbf{(2-20)}$$

If K_1 were selected as it was in the zero order case, then the solution for K_2 would be in the form of a quadratic equation. The difficulties in determining the proper value of K_2 (which may be complex conjugates) make this method more difficult to analyze, that is, a solution might not exist which provide a real value for K_2.

The alternative method used to solve this problem uses the root-locus method to find the gains for satisfactory performance, without causing the actuator signal to saturate.

The characteristic equation written in root locus form with K_1 incorporated into the root locus variable K is:

$$1+K\frac{s+\frac{\beta}{\alpha}}{s\,(s^2+T_2 s+T_1)}=0 \qquad \textbf{(2-21)}$$

where

$$T_3=I+\alpha K_2 \qquad \textbf{(2-22)}$$

$$T_2 = (c+\alpha K_1+\beta K_2)/T_3 \tag{2-23}$$

$$T_1 = (k+\beta K_1)/T_3 \tag{2-24}$$

$$K = K_1/(\alpha T_3) \tag{2-25}$$

There is one zero for this configuration of the characteristic equation. It is

$$s = -\frac{\beta}{\alpha} \tag{2-26}$$

There are also three poles, one of which is at the origin. The other two poles can be placed anywhere by appropriate selection of the gains K_2 and K_1. By observing the behavior of the root locus for different configurations of the pole placement, it was possible to determine a method of solution which provided satisfactory response in a conveniently programmed algorithm. Three possible configurations were examined: complex conjugate poles, real repeated poles to the left of the zero, and real repeated poles to the right of the zero. Real, distinct poles were not considered because of the lack of basis for a root separation factor. The first configuration, complex conjugate poles was ruled out because of the likelihood of an oscillatory response. Either of the remaining configurations may yield satisfactory response characteristics without actuator saturation. The configuration with the poles to the right of the fixed zero was chosen because poles near the origin are less likely to cause saturation.

The plot which is constructed using the root-locus plotting guides (Schwarzenback [7] p. 160) shows the locus breaking away at some point between 0 and the position of the poles, and approaching infinity along an asymptote perpendicular to the real axis (Figure 2-4). The fastest, smoothest response (before adjustment for actuator saturation) will occur at the breakaway point s_{ba}. The solution for the breakaway point is found by solving the root-locus equation for K and differentiating to find the local minimum. The result is the cubic equation:

$$s^3+C_2s^2+C_1s+C_0=0 \tag{2-27}$$

where

$$C_2=(T_1\alpha+3T_2\beta)/(2T_2\alpha) \tag{2-28}$$

$$C_1=(t_1\beta)/(T_2\alpha) \tag{2-29}$$

$$C_0=(T_0\beta)/(2T_2\alpha) \tag{2-30}$$

The breakaway point should be the only real root between 0 and the pole position. The root-locus variable K at the breakaway point is

$$K=\frac{|s_{ba}||s_{ba}^2+T_2s_{ba}+T_1|}{|s_{ba}+\frac{\beta}{\alpha}|} \tag{2-31}$$

The gain values K_2 and K_1 selected to place the poles near the zero, and the value of K_1 determined by (above) do not guarantee that actuator saturation will not occur. Therefore, it is necessary to simulate the actuator response, and adjust the values accordingly. The method used to adjust the gains is to incrementally place the real, repeated poles closer to the origin in the root-locus plot, there by slowing the system down, until the actuator does not saturate. A graphical representation of the method is shown in Figure 2-5. This method is easily coded as a computer program algorithm.

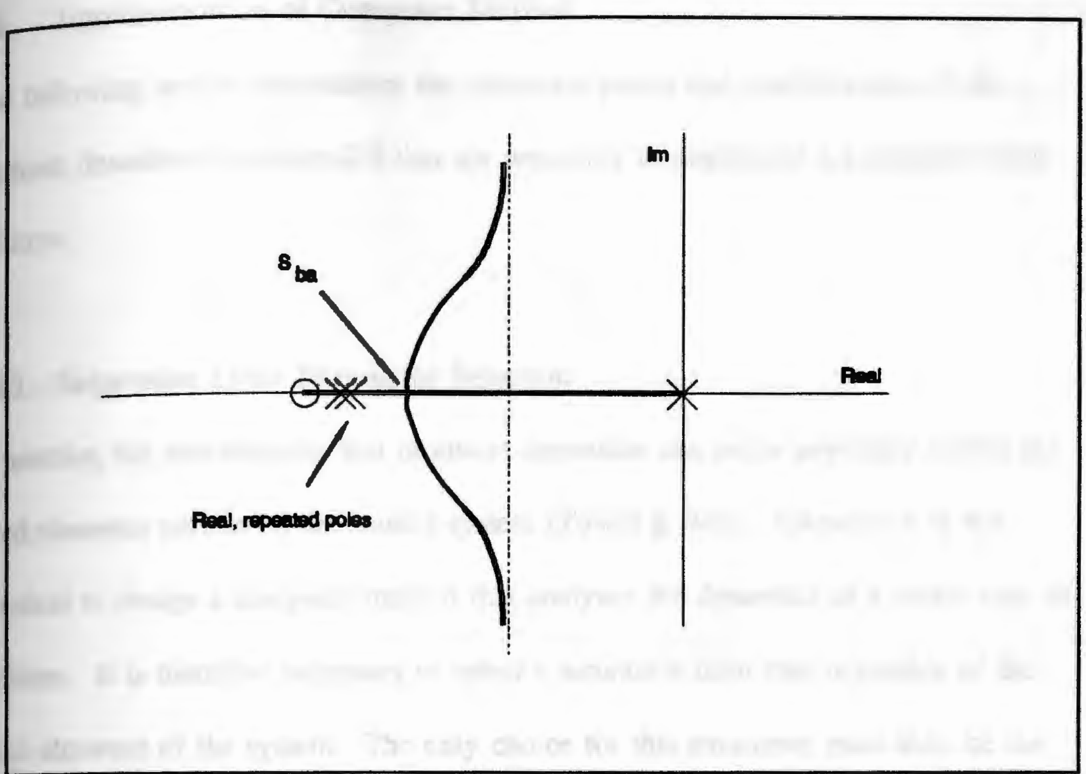

Figure 2-4 Root locus plot for first order numerator dynamic configuration

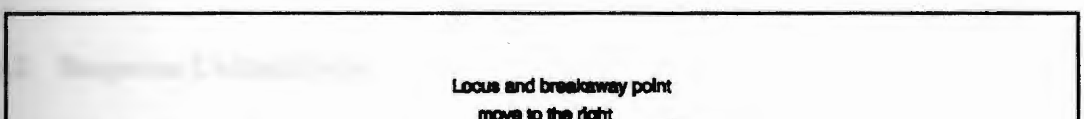

Figure 2-5 Pole adjustment method used to find non-saturating response for first order numerator configuration

2.6 Implementation of Computer Method

The following section summarizes the important points and modifications of the methods described in section 2.5 that are necessary to implement a computer based solution.

2.6.1 Saturation Limit Parameter Selection

In practice, the non-linearity that produces saturation can occur anywhere within the fixed elements portion of the control system (Towill p. 411). Likewise it is not practical to design a computer method that analyzes the dynamics of a single type of problem. It is therefore necessary to select a saturation limit that is outside of the fixed elements of the system. The only choice for this parameter must then be the control signal, V.

2.6.2 Response Calculations

The control signal, V, in Figure 2-2 can be represented as:

$$v(t)=K_I\int e(t)\,dt-K_1c(t)-K_2\frac{dc(t)}{dt} \quad \textbf{(2-32)}$$

where

$$e(t)=r(t)-c(t) \quad \textbf{(2-33)}$$

$$r(t)=M(1-e^{-zt}) \quad \textbf{(2-34)}$$

The overall system response, c(t), is found using the fourth order Runge Kutta method for solving third order differential equations. The step size required for the Runge Kutta method is chosen based on the least dominant root time constant (τ_s):

$$h=\frac{\tau_s}{100} \tag{2-35}$$

The integral of the error signal is found by sub-dividing each Runge Kutta step, h, by 10 and applying Simpson's Rule over the span of h. The integral of the error signal would become

$$\int e(t)\,dt=\frac{h}{30}\left(s_0+4s_1+s_2\right) \tag{2-36}$$

(Kreyzig [8] p. 789) where

$$s_0=e_0+e_{10} \tag{2-37}$$

$$s_1=e_1+e_3+e_5+e_7+e_9 \tag{2-38}$$

$$s_2=e_2+e_4+e_6+e_8 \tag{2-39}$$

and e_n is the error signal evaluated at each of the sub-intervals of the Runge Kutta step size.

2.6.3 Computational Differences from Method Discussed in Section 2.5

The method described by Phelan suggests a trial and error approach of adjusting the integral gain K_i upward from a low value until saturation occurs. Computationally this would require calculation of the entire control signal response (for about 4 time constants) before a determination of saturation (or not) could be made. To minimize computation time for this iterative procedure, the integral gain is initially selected to be some maximum value to ensure system stability. The gain is then adjusted downward until a control signal response is found that does not saturate. The initial value of the gain corresponds to the point where the root locus (Figure 2-6) crosses the imaginary axis.

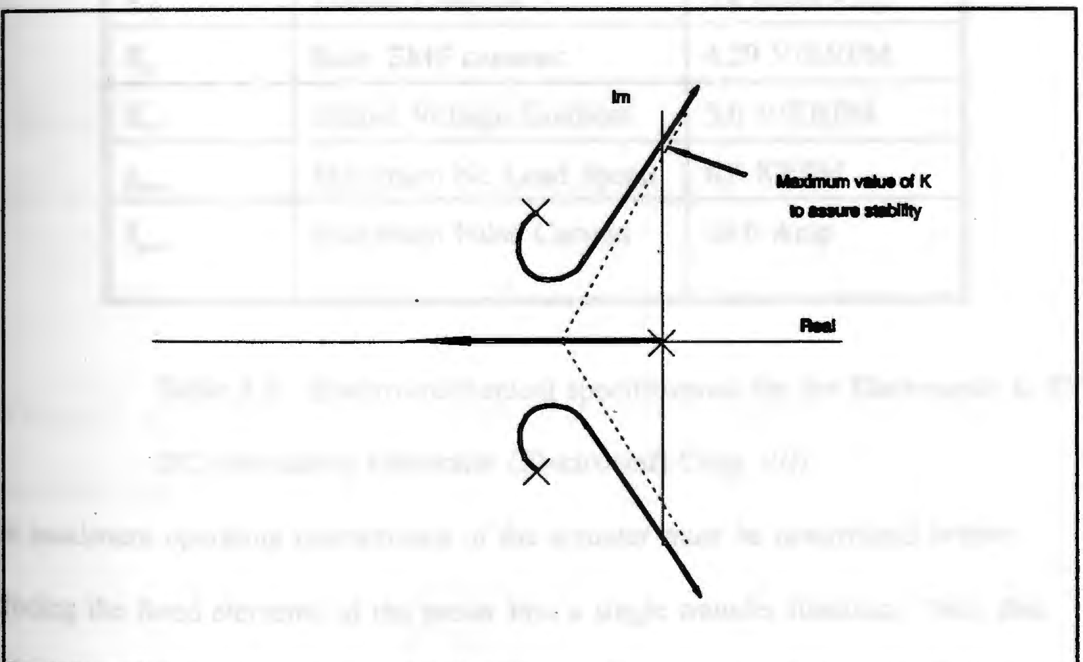

Figure 2-6 Root locus plot for zero order numerator dynamic configuration

3. APPLICATION OF COMPUTER METHOD

3.1 Derivation of Problem with Zero Order Numerator Dynamics

Figure 3-1 is a block diagram representing PDF control of a DC motor. Such instruments are used in a wide variety of precision velocity and positioning control systems. A manufacturer's specifications for the motor components for such a motor are listed in Table 1.

Term	Description	Value
c	Damping Factor	0.1 oz-in/KRPM
I	Armature Inertia	0.0055 oz-in-sec^2
R	Armature Resistance	1.55 Ohms
L	Armature Inductance	3.19 mH
K_T	Torque Constant	5.8 oz-in/Amp
K_e	Back EMF constant	4.29 V/KRPM
K_a	Output Voltage Gradient	3.0 V/KRPM
r_{max}	Maximum No Load Speed	6.0 KRPM
I_{max}	Maximum Pulse Current	24.0 Amp

Table 3.1 - Electro-mechanical specifications for the Electrocraft E-576 DC Servomotor Generator (Electrocraft Corp. [9])

The maximum operating chacteristics of the actuator must be determined before reducing the fixed elements of the motor into a single transfer function. Note that among the specifications is a maximum pulse current (which is important to avoid de-magnetization of the motor's components).

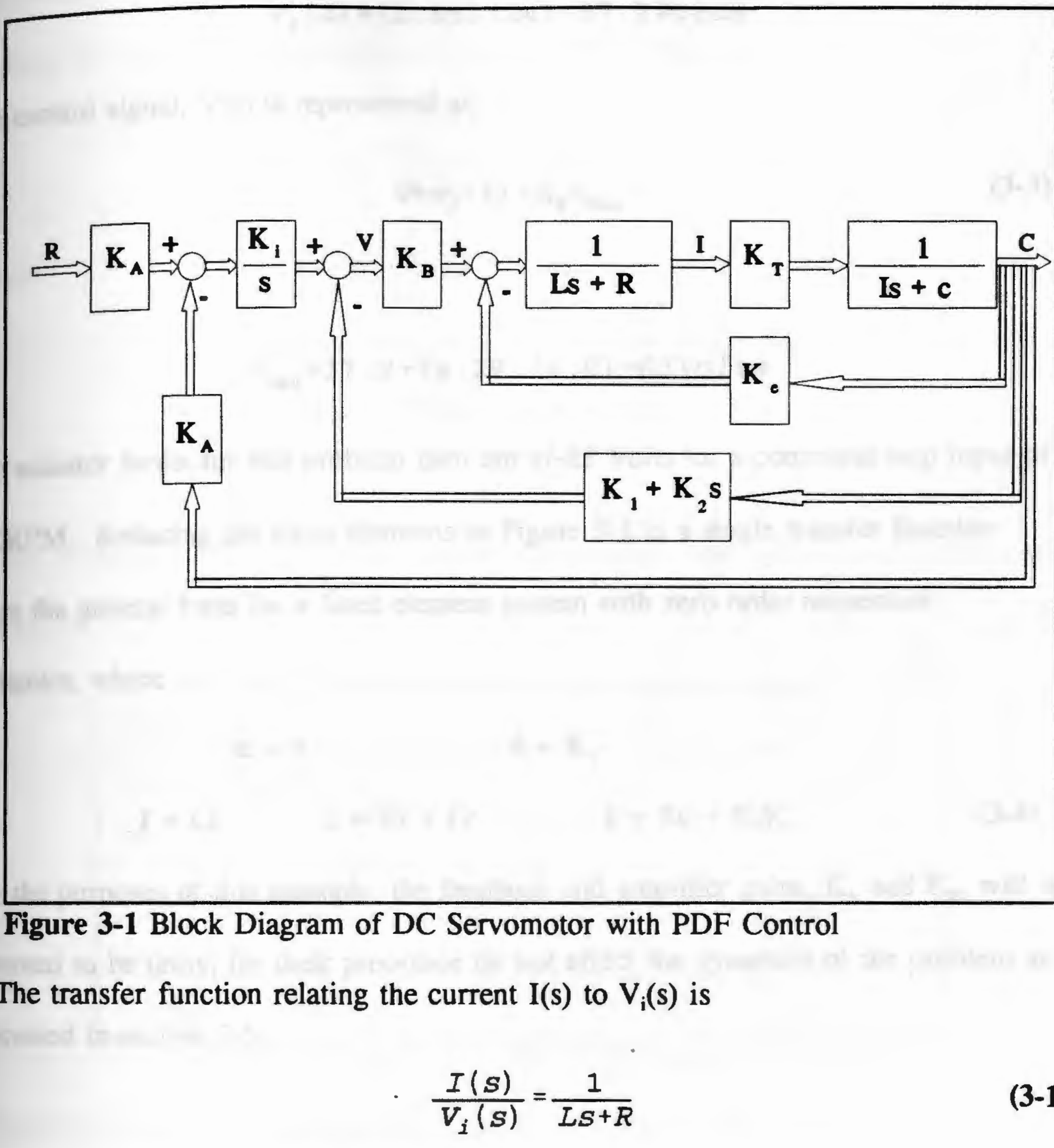

Figure 3-1 Block Diagram of DC Servomotor with PDF Control

The transfer function relating the current I(s) to $V_i(s)$ is

$$\frac{I(s)}{V_i(s)} = \frac{1}{Ls+R} \tag{3-1}$$

To simplify the problem, it is advantageous to neglect the effects of armature inductance temporarily so that

$$v_i(t) = Ri(t) \tag{3-2}$$

Substitution of the maximum pulse current for i(t) and the armature resistance gives

$$v_i(t)=(1.55)(24)=37.2\,Volts$$

The control signal, V(s) is represented as

$$V=v_i(t)+K_e C_{max} \tag{3-3}$$

$$V_{max}=37.2+(4.29)(6.0)=63\,Volts$$

The actuator limits for this problem then are +/-63 Volts for a command step input of 6 KRPM. Reducing the fixed elements in Figure 3-1 to a single transfer function gives the general form for a fixed element system with zero order numerator dynamics, where

$$\alpha = 0 \qquad \beta = K_T$$

$$I = LI \qquad c = RI + Lc \qquad k = Rc + K_T K_e \tag{3-4}$$

For the purposes of this example, the feedback and amplifier gains, K_A and K_B, will be assumed to be unity, for their precensce do not affect the dynamics of the problem as discussed in section 2.5.

3.2 Derivation of Problem with First Order Numerator Dynamics.

The liquid level system shown in Figure 3-2 consists of two coupled tanks, each of which has an outflow pipe with known diameters and lengths. The fluid resistance due to laminar pipe flow is given by the Hagen-Poiseuille formula

$$R=\frac{128\mu L}{\pi\rho D^4} \tag{3-5}$$

Based upon the block diagram shown in Fig. 3-3, the transfer function relating the volume flow rate, q, to the height of the liquid in the first tank is

$$\frac{H_1(s)}{Q(s)}=\frac{\alpha s+\beta}{Is^2+cs+k} \tag{3-6}$$

where

$$\alpha = R_1R_2A_2 \qquad \beta = g(R_1 + R_2) \tag{3-7}$$

$$I = A_1R_1A_2R_2 \quad c = g(A_1R_1 + A_1R_2 + A_2R_2) \quad k = g^2$$

Table 3-2 provides the appropriate parameters for a two tank system with fuel oil as the liquid.

Parameter	**Tank 1**	**Tank 2**
Tank Diameter (m)	1.0	0.75
Pipe Diameter (m)	0.04	0.05
Pipe Length (m)	0.2	0.1
Area (m^2)	0.78	0.44
Resistance (N-sec/kg-m^2)	3153.9	645.9

Table 3.2 - Parameters for Two Tank Liquid Level System

The required physical specifications of oil at 68 degrees F are:

Density, 968.9 kg/m3

Viscosity, 0.96 N-sec/m2

In order to select reasonable values for an actuator limit and a corresponding input command height for the first tank, it is necessary to examine the steady state value of the control signal in terms of the command request. The transfer function relating the control signal, V(s), to the command height $H_{1R}(s)$ is:

$$\frac{V(s)}{H_{1R}(s)}=\frac{K_1(Is^2+cs+k)}{T_3s^3+T_2s^2+T_1s+T_0} \tag{3-8}$$

Applying the final value theorem (Palm, p. 224-226) to the above transfer function and substituting the value for T_0 from eqn. the following relationship results:

$$V_{SS}=\frac{H_{1R}k}{\beta} \tag{3-9}$$

For a maximum step command in liquid level (of the first tank) that would ever be expected, say 2 meters, a corresponding steady state value of V (which in this case is flow rate) can be found required to maintain the height of 2 meters. The flow rate becomes:

$$V_{SS}=\frac{(2)\,(96.04)}{37238}=0.005158m^3/\text{sec}$$

The above flow rate is used as the upper actuator limit in the tank problem. A lower actuator limit of 0 is chosen to represent the flow rate when the input valve is completely closed. A reasonable value for the command liquid height in the first tank must be between 0 and 2 meters. A command request of 1 meter is used.

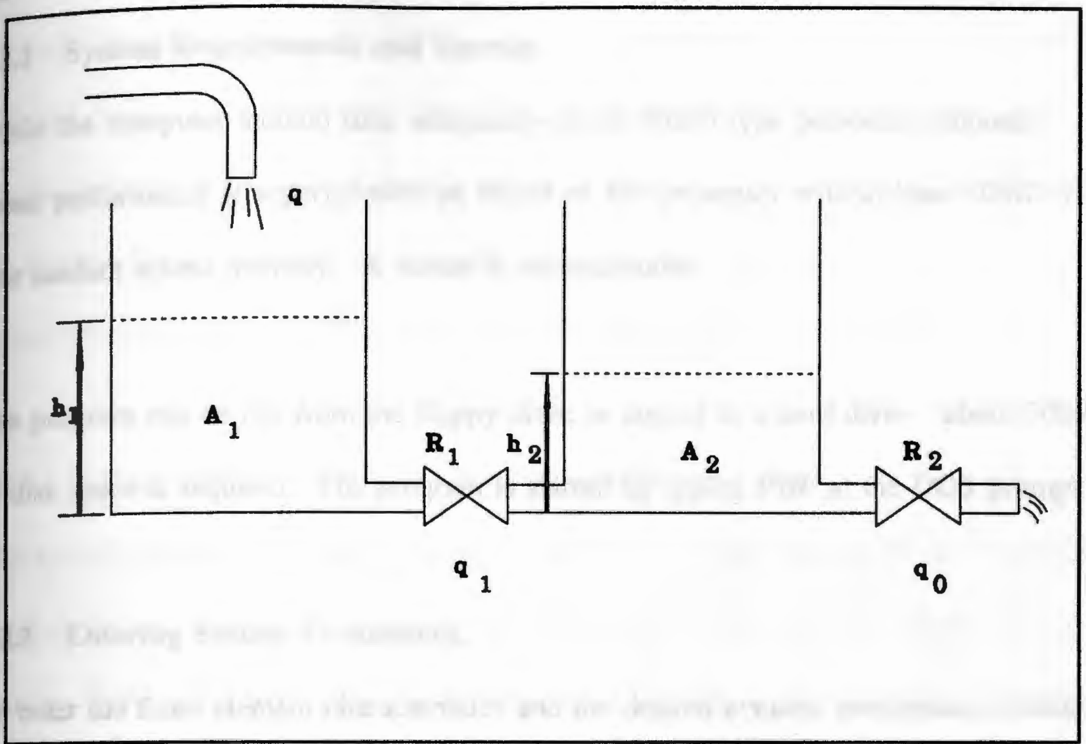

Figure 3-2 Two Tank Liquid Level Control System

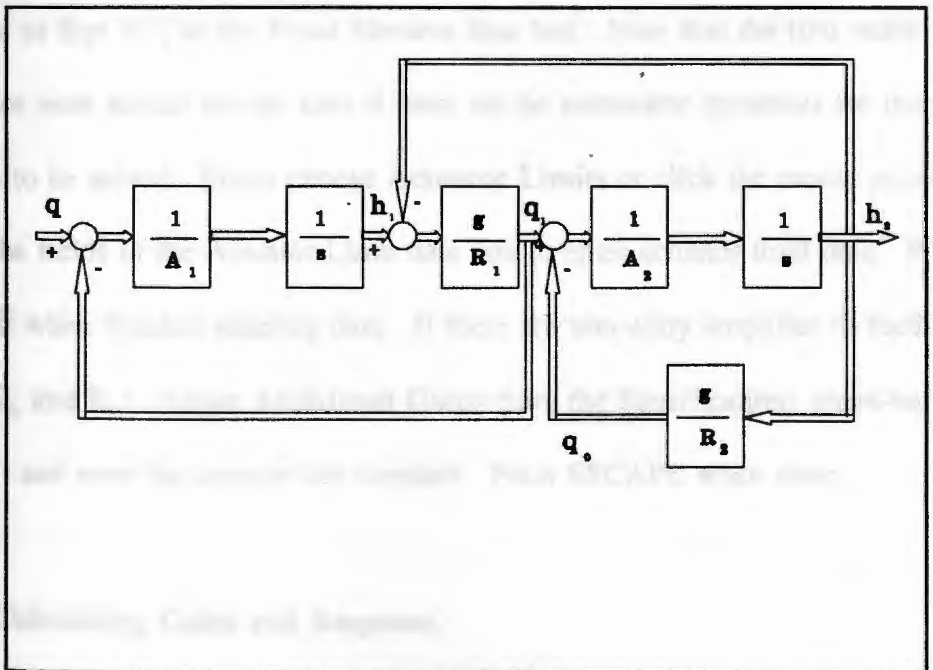

Figure 3-3 Block Diagram for Two Tank Liquid Level System

3.3 Operating the Computer Program

3.3.1 System Requirements and Startup

While the computer method runs adequately on an 80286 type personal computer, speed performance is superior with an 80386 or 486 processor with at least 400K of free randam access memory. A mouse is recommended.

The program can be run from the floppy drive or copied to a hard drive - about 300K of disk space is required. The program is started by typing **PDF** at the DOS prompt.

3.3.2 Entering System Parameters.

To enter the fixed element characteristics and the desired actuator performance limits, choose **Specifications** from the menu bar at the top of the display. Choose **Fixed Element** to enter the physical parameters of the system to be controlled (the form is the same as Eqn 3-7) in the Fixed Element data box. Note that the first order numerator term should remain zero if there are no numerator dynamics for the problem to be solved. Either choose **Actuator Limits** or click the mouse pointer on one of the fields in the Actuator Limit data box to enter actuator limit data. Press ESCAPE when finished entering data. If there are non-unity amplifier or feed-back gains (K_A and K_B), choose **Additional Gains** from the Specifications menu-bar selection and enter the appropriate constant. Press ESCAPE when done.

3.3.3 Calculating Gains and Response.

To calculate the gains and view the control signal response, select **Go** from the menu bar and then the **Go** sub-menu choice. After a few seconds a line graph is displayed with the control signal response. The PDF gains and control signal maximum and minimum values are displayed in data boxes on the right side of the display. Choose the **System Response** sub-menu from the **Go** menu-bar selection to view the overall system response with the gains that have been determined in the previous action.

3.3.4 Saving and Retrieving Specifications and Responses.

The system specifications can be saved to a disk file so that they can be conveniently recalled for another time. Select **File** from the menu-bar and the **Save Specs** submenu choice. Type a file name with no extension (a .PDF will be added). Press ESCAPE not ENTER when done. To recall a saved file select **File** from the menu-bar and the **Retrieve Specs** submenu choice. A box is displayed with all the PDF data files in the current directory. Select a file to retrieve by clicking the mouse pointer on the desired file.

To save a control signal or overall system response to a spreadsheet importable file, display the desired line plot (with **Go**) and choose **Save Response** from the **File** menu-bar selection. Type a file name (with or without an extention) for the destination file. Press ESCAPE not ENTER when done.

3.3.5 Quitting the Program.

Select **Quit** from the menu-bar to leave the program and return to DOS. A box is displayed verifying the action. Press the space bar to toggle the Yes/No field in the box and click **Quit** on the menu-bar again to complete the action.

4. RESULTS

4.1 Results for the zero order numerator dynamic problem example.

The results obtained from the computer method for the DC Motor example outlined in section 3.1 are (with control signal limits of +/- 63.0 Volts and a requested speed of 6 KRPM)

Input parameters (calculated from Table 3-1 and Equations 3-4):

$\alpha = 0.0$ $\beta = 5.8$

$I = 1.8 \times 10^{-5}$ $c = 8.844 \times 10^{-3}$ $k = 24.882$

Gains:

$K_2 = 2.188 \times 10^{-2}$

$K_1 = 8.400 \times 10^{1}$

$K_i = 1.198 \times 10^{5}$

Control Signal Min/Max:

$V_{min} = 0.0$ Volts

$V_{max} = 42.5$ Volts

Characteristic Roots:

$r_1 = -2.395 \times 10^{3}$ (Dominant Time Constant of 0.0004 sec)

$r_{2,3} = -2.573 \times 10^{3} \pm i3.082 \times 10^{3}$

Figures 4-1 and 4-2 provide the graphical plots of the control and system output signals.

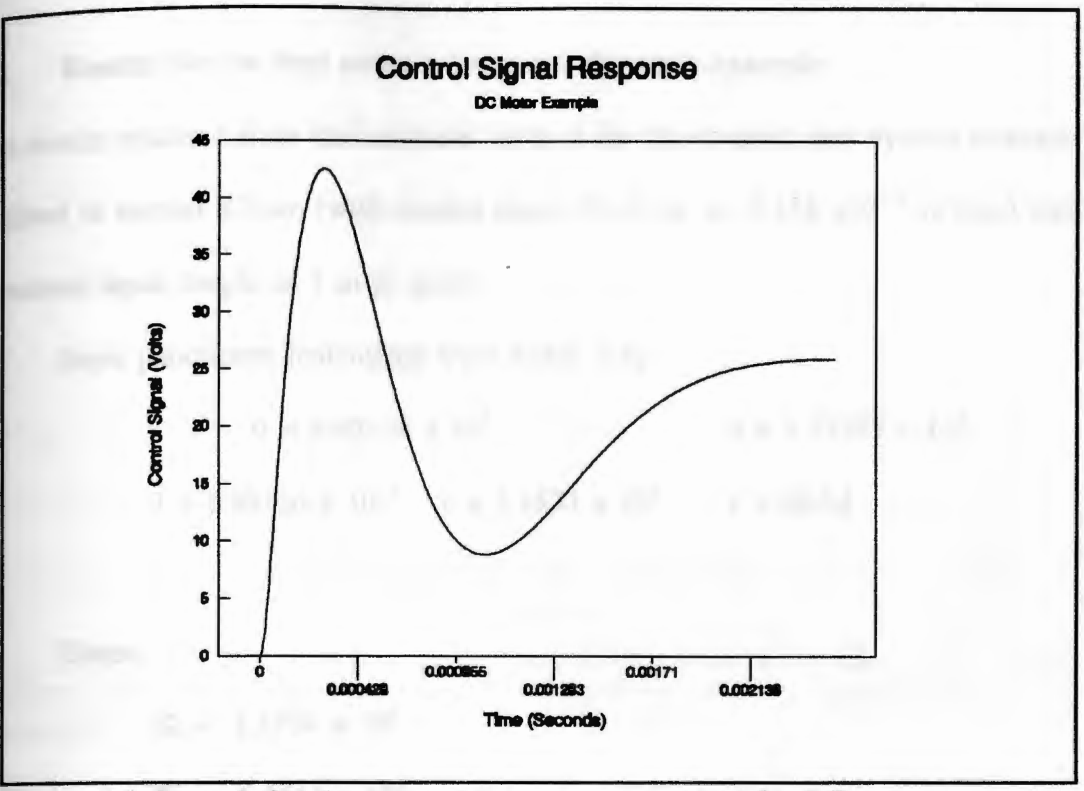

Figure 4-1 Control signal generated by computer method for DC motor example

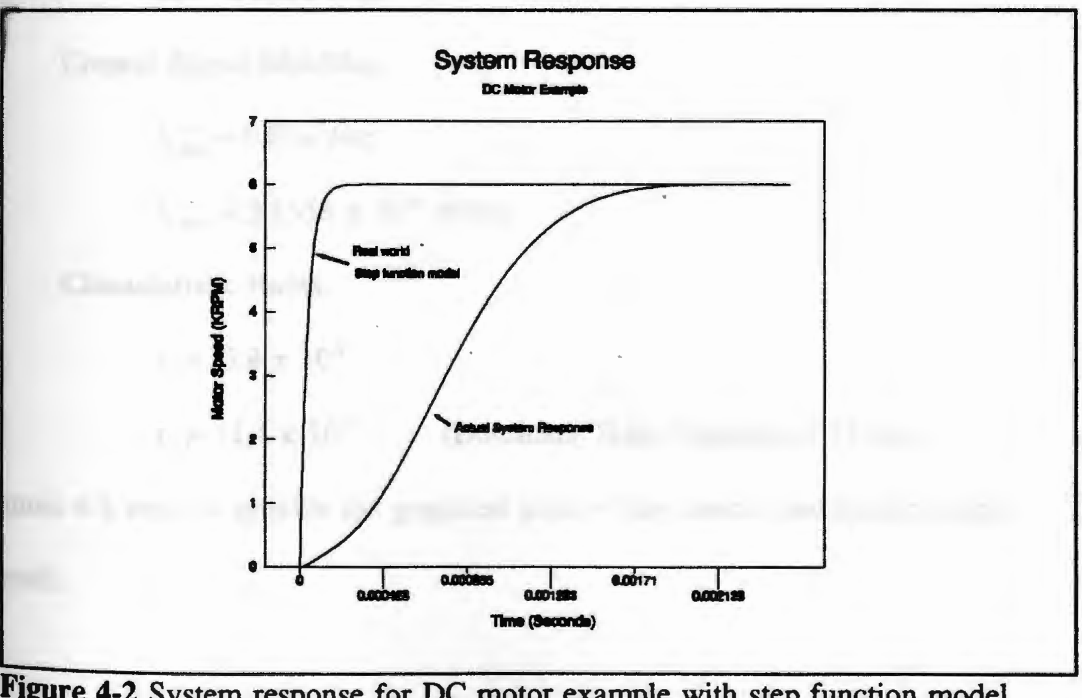

Figure 4-2 System response for DC motor example with step function model

4.2 Results for the first order numerator dynamic example

The results obtained from the computer method for the coupled tank system example outlined in section 3.2 are (with control signal limits of +/- 5.158 x10^{-3} m^3/sec) and a requested liquid height of 1 m in tank1:

Input parameters (calculated from Table 3-2):

$\alpha = 8.96328 \times 10^5$ $\qquad$ $\beta = 3.72377 \times 10^4$

$I = 6.99136 \times 10^5$ $\qquad$ $c = 3.1831 \times 10^4$ $\qquad$ $k = 96.04$

Gains:

$K_2 = -1.1724 \times 10^0$

$K_1 = -1.3541 \times 10^{-2}$

$K_i = -8.0617 \times 10^{-5}$

Control Signal Min/Max:

$V_{min} = 0.0$ m^3/sec

$V_{max} = 5.1558 \times 10^{-3}$ m^3/sec

Characteristic Roots:

$r_1 = -3.8 \times 10^{-2}$

$r_{2,3} = -1.4 \times 10^{-2}$ $\qquad$ (Dominant Time Constant of 71 sec)

Figures 4-3 and 4-4 provide the graphical plots of the control and system output signals.

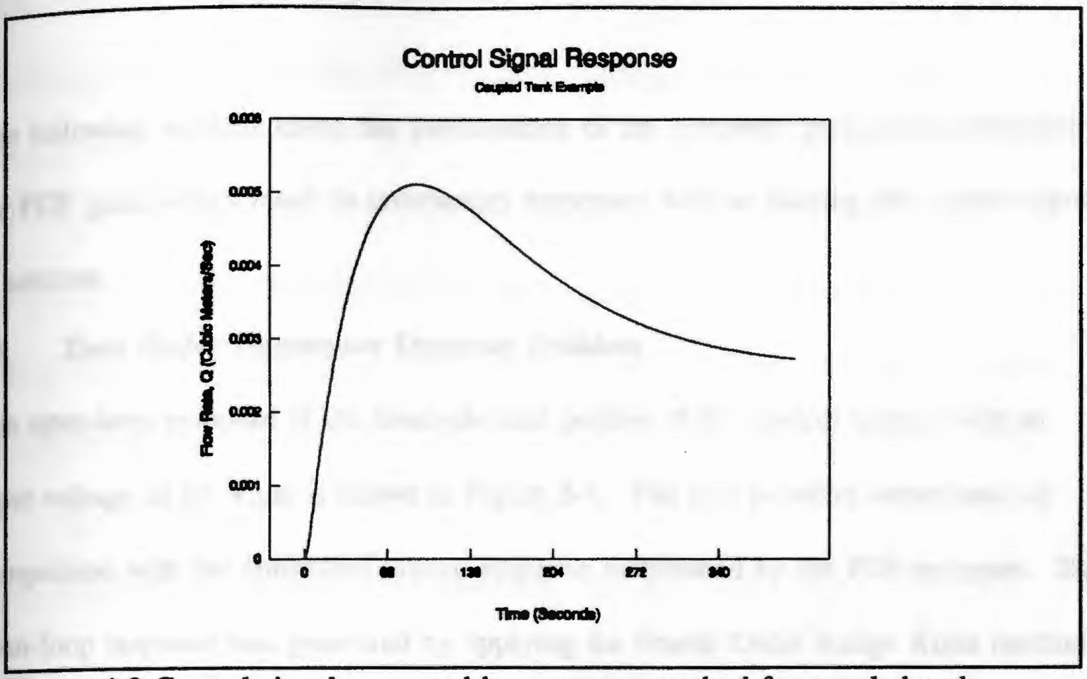

Figure 4-3 Control signal generated by computer method for coupled tank example

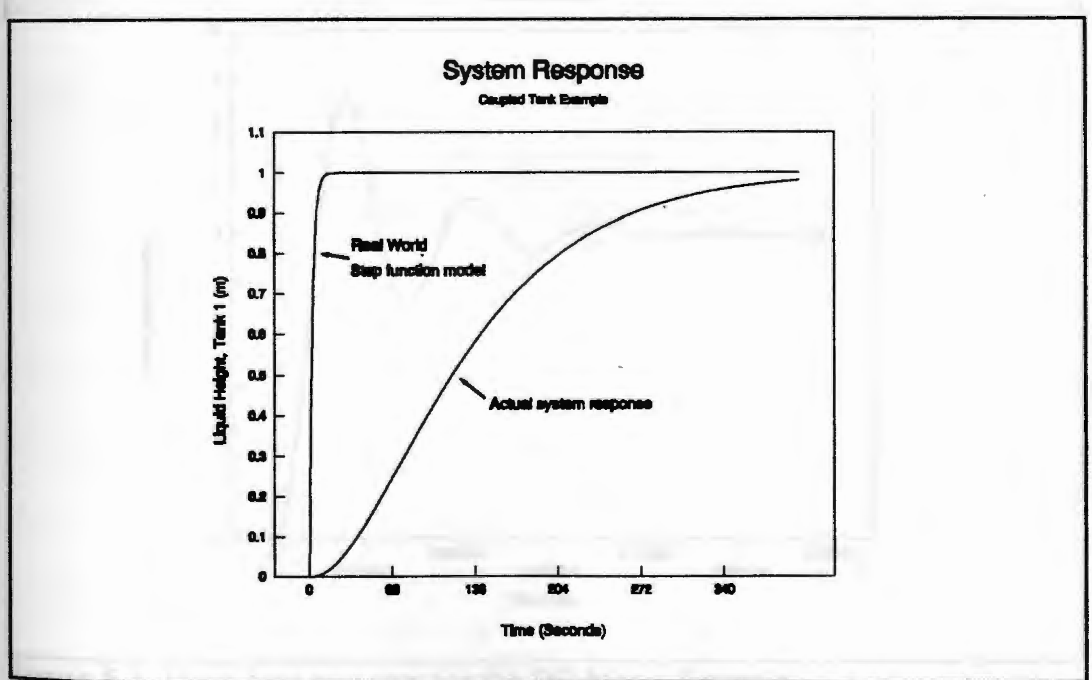

Figure 4-4 System response for coupled tank example with step function model

5. Discussion

The following sections assess the performance of the computer program in determining the PDF gains which result in satisfactory responses without causing the control signal to saturate.

5.1 Zero Order Numerator Dynamic Problem

The open-loop response of the fixed-element portion of the control system with an input voltage of 63 Volts is shown in Figure 5-1. The plot provides some basis of comparison with the controlled system response determined by the PDF program. The open-loop response was generated by applying the Fourth Order Runge Kutta method

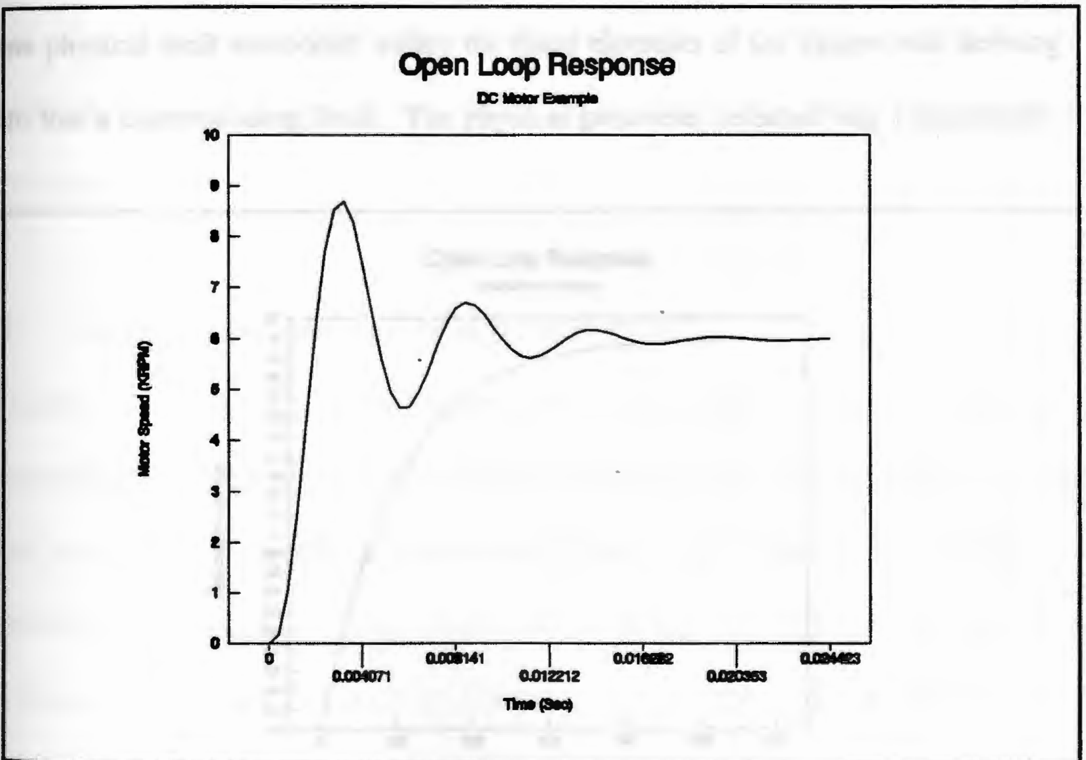

Figure 5-1 Open-loop response for the DC Motor Example

to the second order transfer function of the fixed elements of the control system. The dominant root characteristics of the fixed elements are (time constant and damping ratio):

$\tau_D =$ 0.0041 sec $\qquad\qquad \zeta = 0.21$

The results from the PDF program indicate a significant smoothing of the response with no accompanying saturation of the control signal (voltage) supplied to the motor. The dominant root characteristics for the PDF solution presented in section 4.1 are

$\tau_D = 0.0004$ sec $\qquad\qquad \zeta = 1.00$

which represent a faster, smoother responding system.

The determination of the control signal limit for this problem involved the selection of some physical limit embedded within the fixed elements of the system and deriving from that a corresponding limit. The physical parameter selected was a maximum

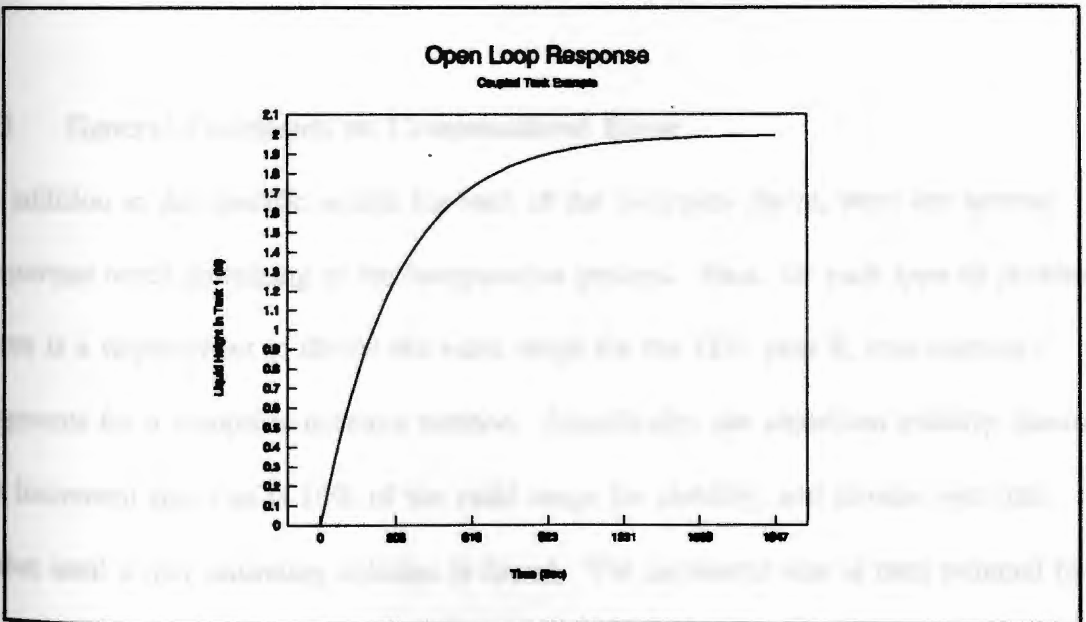

Figure 5-2 Open-loop response for the coupled tank example

pulse current. There may be a more appropriate parameter to use to derive the limit (such as the maximum torque a motor can generate) but such data was not available from the manufacturer's specifications.

5.2 First Order Numerator Dynamics Problem

The open-loop response for the tank example is shown in Figure 5-2. The dominant root characteristics for the fixed elements are:

$\tau_D =$ 303 sec $\qquad\qquad \zeta = 1.94$

It can be observed from Figure 5-2 that the liquid in the tank would reach a height of 1 meter (the requested liquid height entered into the PDF program) in about 200 seconds if the input valve were fully opened (a flow rate of 0.005 m^3/sec) and left open. Of course the controlled response determined by the program would be preferable, especially if there were a design constraint that the liquid height not exceed its set-point value.

5.3 General Comments on Computational Error

In addition to the specific results for each of the examples above, there are several important notes pertaining to the computation process. First, for each type of problem, there is a requirement to divide the valid range for the PDF gain K_i into discrete segments for a computer iteration method. Specifically, the algorithm initially chooses an increment size that is 10% of the valid range for stability, and iterates with this value until a non-saturating solution is found. The increment size is then reduced by factor of 10 and a more precise solution is found. An unavoidable result of this segmentation process is that it is nearly impossible to find the absolute optimum gains that will provide a control signal response that does not saturate and have the smallest time constant and smoothest response possible.

Also, there are potential round-off or truncation errors inherent in the Runge Kutta numerical method. The magnitude of any such errors would be far less than is required to cause computational mistakes when comparing the calculated control signal to the limit specified by the user or in generating visual differences in the graphic plots.

6. CONCLUSIONS AND RECOMMENDATIONS

The purpose of this study was to investigate the design of feedback control systems within the limitations of the finite energy delivery capability of the system's physical elements. The development of the computer algorithms to accomplish the investigation indicated that not only is it possible to design control systems in this manner, but that there are many possible combinations of design methods which provide acceptable performance within the saturation prevention constraint.

While the configuration of the problem examined by the computer program is only capable of examining two general classes of physical systems, it does approximate the dynamics of a wide variety of potential systems to be controlled. Some possible areas for further study of this type of problem include:

- Investigate the effects of different input functions, such as a ramp function, to see if the same or similar methods of solution can be developed.
- Apply different methods to prevent actuator saturation such as the anti-reset windup method.
- Expand complexity and flexibility of the physical parameters to be controlled by increasing the order of the dynamics of the fixed elements.
- Investigate different types of non-linearities inherent in real systems

- Investigate different computer programming technology to develop more flexible tools for automated control system design. This might involve the the development of a core library of object code which provides simulation and graphics utilities around which different controller design methods can be developed.

7.0 REFERENCES

1. Phelan, R. *Automatic Control Systems*. Ithaca, New York: Cornell University Press, 1977.

2. Palm, W. *Control Systems Engineering*. New York: John Wiley & Sons, 1986.

3. Towill, D.R. *Transfer Function Techniques for Control Engineers*. London: Iliffe Books LTD, 1970.

4. Astrom, K. J. and T. Hagglund. *Automatic Tuning of PID Controllers*. Research Triangle Park, NC: Instrument Society of America, 1988.

5. Boyd, S., Craig Barratt, and Stephen Norman. "Linear Controller Design: Limits of Performance Via Convex Optimization." Proceedings of the IEEE, March 1990, pp. 529-574.

6. Ulsoy, A.G. "Optimal Pseudo-Derivative Feedback Control." M.S. thesis. Cornell University 1975.

7. Schwarzenback, J. and K.F. Gill. *System Modelling and Control*. New York: John Wiley & Sons, 1978.

8. Kreyszig, E. *Advanced Engineering Mathematics*. New York: John Wiley & Sons, 1983.

9. Electrocraft Corp. *DC Motors, Speed Controls, Servo Systems*. 5th edition. Hopkins, MN, 1980.

APPENDIX A

The following computer listing printouts contain the algorithms that perform the PDF gain calculations for both the first order and zero order numerator dynamic configurations and the subsequent control signal calculation.

```
/*
*       phelan()
*
*DESCRIPTION:
*
*This function uses Phelan's method for the zero order numerator
*       dynamics problem.  Finds the PDF gains and response which do not
*       cause the control signal to exceed the actuator limits.
*
*       VARIABLES:
*
*       (global)
*               alpha           first-order numerator term
*               beta            zero-order numerator term
*               inertia         second-order denominator term
*               damping         first-order denominator term
*               spring          zero-ordre denominator term
*               upper_limit     upper saturation limit
*               rmax            command step function magnitude
*               kd2, kd1, ki    PDF gains
*               ka, kb          feedback and amplifier gains
*               term3, term2    characteristic equation terms
*               term1, term0
*
*               increment       magnitude of Ki adjustment per iteration
*               fvalue          steady state value check (for step magnitude)
*
*       (local)
*               kmax            maximum value of ki for stability
*               done            flag indicating precise solution found
*
*       PSEUDOCODE:
*
*       IF (the input step function magnitude causes the steady state
*       value of the control signal to saturate)
*               display an error message
*               return
*       ENDIF
```

```
*       calulate kd2 and kd1 using phelan's methodology
*       find maximum value of ki for stability and increment size
*       DO
*               find characteristic roots while slowly decreasing ki
*       WHILE ( complex conjugate roots are more dominant than the
*               the real root, in order to avoid an oscillatory solution)
*       DO
*               DO
*                       find a non-saturating response while decreasing ki
*                       IF (a solution is found)
*                               don't continue
*                       ENDIF
*               WHILE   ( ki is positive )
*               IF ( solution was not found )
*                       back up one iteration step and decrease
*                       the increment size
*               ENDIF
*       WHILE ( a precise solution has not been found )
*       plot the control signal response
*       return
*****************************************************************/
int     phelan( )
{
        double k, kmax;
        int     done, icount;

        fvalue = upper_limit*beta*kb/spring;
        if ( rmax > fvalue )
        {
                sprintf( message, "\
  The step size for these control signal parameters   \
\n  is too large ( the steady state control signal      \
\n  will exceed your specified limits). Please choose \
\n  another step size.");
                pop_Prompt( message, -1, -1, 6, 56, 0x47, bd_1);
                return(0);
        }

        kd1 = 8.0*upper_limit/(ka*rmax);
        kd2 = (2.0*0.707*sqrt( inertia*( spring + beta*kd1*kb ) ) - damping)/(beta*kb);

        term3 = inertia;
        term2 = (damping + beta*kd2*kb);
        term1 = (spring + beta*kd1*kb);
```

```
    kmax = term2*term1/(ka*kb*term3*beta);
    increment = -0.01*kmax;

    /*      Ensure gains don't result in oscillatory sol'n          */

    do
    {
            char_roots = cubic( term2/term3, term1/term3, kmax*beta/term3);
            kmax += increment;
    } while( char_roots.real[0] < char_roots.real[1]);

    ki = kmax - increment;

    done = 0;
    do
    {
            do
            {
                    if (done = cntrl_sgnl())
                            break;
                    ki += increment;
            }while( ki > fabs(increment) );

            if (!done)
            {
                    k -= increment;
                    increment /= 10.0;
                    k += increment;
            }
    } while (!done);
    sed_Close(wait);
    plot_response( 2, done);
    return(1);
}
```

```
/*
*     go()
*
*     DESCRIPTION:
*     This function performs the root-locus iteration procedure, for
*     the first-order numerator dynamic configuration problem.  Finds
*     the PDF gains and response which do not cause the control signal
*     to exceed the actuator limits.
*
*     VARIABLES:
*
*     (global)
*            alpha            first-order numerator term
*            beta             zero-order numerator term
*            inertia          second-order denominator term
*            damping                  first-order denominator term
*            spring           zero-ordre denominator term
*            upper_limit      upper saturation limit
*            rmax             command step function magnitude
*            kd2, kd1, ki     PDF gains
*            ka, kb           feedback and amplifier gains
*
*            poles            root-locus position of the repeated poles
*            increment        magnitude of pole adjustment per iteration
*            breakpt                  point where locus breaks away from real axis
*            fvalue           steady state value check (for step magnitude)
*
*     (local)
*            zero             root-locus position of the zero
*            k                root locus variation parameter
*            found            flag indicating rough solution found
*            done             flag indicating precise solution found
*
*     PSEUDOCODE:
*
*     IF (the input step function magnitude causes the steady state
*     value of the control signal to saturate)
*            display an error message
*            return
*     ENDIF
*     calulate zero, initial pole position and increment
*     DO
*            move poles to the right a little
*            calculate integral gain at locus breakaway point
```

```
*               calculate control signal response until saturation occurs
*               IF ( a non-saturating response was found)
*                    back up one iteration step and find
*                    a more precise solution
*               ENDIF
*               IF ( poles become positive )
*                    back up one iteration step and decrease
*                    the increment size
*               ENDIF
*       WHILE ( a precise solution has not been found )
*       plot the control signal response
*       return
*****************************************************************************/
int     go()
{
        double zero, k;
        int     done, found;

        found = 0;
        fvalue = upper_limit*(beta*kb)/spring;
        if ( fvalue < rmax )
        {
                sprintf( message, "\
  The step size for these control signal parameters   \
\n   is too large ( the steady state control signal      \
\n   will exceed your specified limits).  Please choose \
\n   another step size.");
                pop_Prompt( message, -1, -1, 6, 56, 0x47, bd_1);
                return(0);
        }

        zero = - beta/alpha;
        increment = 0.1*fabs(zero);
        poles = zero;

        do
        {
                poles += increment;
                breakpt = calculate_breakaway( poles );
                k = fabs(breakpt)*fabs( pow(breakpt,2.0) + (term1/term2)*breakpt +
(term0/term2))/
                        fabs( breakpt + (beta/alpha));
                term3 = inertia + alpha*kd2*kb;
                ki = (k*term3)/(alpha*kb);
```

```
        done = cntrl_sgnl();
        if (done && !found)
        {
                found = 1;
                done = 0;
                poles -= increment;
                increment /= 10.0;
        }

        if ( fabs(poles) < fabs(increment) )
        {
                poles -= increment;
                increment /= 10.0;
        }

    } while ( !done );

    sed_Close(wait);
    plot_response( 2, done);
    return(1);
}
```

```
/*
*       calculate_breakaway(point)
*
*       DESCRIPTION:
*       This function calculates the the PDF gains, KD2 and KD1, required
*       to place real, repeated poles at the position specified by the
*       input argument 'point'.  Subsequent to the gain calculation, the
*       actual breakaway point, where the root locus splits away from the
*       negative real axis is calculated and returned to the calling function.
*
*       VARIABLES:
*
*       (global)
*               breakaway       structure of type cubic_root to store
*                               results of breakaway calculation.
*               kd2, kd1        PDF gains
*               term3, term2    characteristic equation terms
*               term1, term0
*
*       (local)
*               tc              time constant of input parameter, point
*
*               a1,a2,b1,b2,    These variables relate to the following
*               acon, bcon      set of simultaneous equations (which
*                               result from choosing kd2 and kd1 such
*                               that the poles are equal.
*
*                               | a1  a2 | | kd2 | = | acon |
*                               | b1  b2 | | kd1 | = | bcon |
*
*               det             value of above determinant
*               c2, c1, c0      intermediate variables
*               maxval                  maximum of the 3 real roots (breakaway)
*
*
*       PSEUDOCODE:
*
*       calculate kd2 and kd1 to place poles at point
*       find breakaway point for these poles
*       RETURN the breakaway point
******************************************************************/
double  calculate_breakaway( double point )
```

```
{
        double det, tc;
        double  a1, a2, acon, b1, b2, bcon, tsqrd;
        double  c2, c1, c0;
        double  maxval;
        int     j;
        tc = -1.0/point;

        tsqrd = tc*tc;
        a1 = 2.0*alpha - tc*beta;
        a2 = -tc*alpha;
        acon = tc*damping - 2.0*inertia;
        b1 = alpha;
        b2 = -tsqrd*beta;
        bcon = tsqrd*spring - inertia;

        det = a1*b2 - b1*a2;
        kd2 = (double) (acon*b2 - bcon*a2)/(det*kb);
        kd1 = (double) (a1*bcon - b1*acon)/(det*kb);

        term2 = inertia + alpha*kd2*kb;
        term1 = damping + alpha*kd1*kb + beta*kd2*kb;
        term0 = spring + beta*kd1*kb;

        c2 = (term1*alpha + 3.0*term2*beta)/(2.0*term2*alpha);
        c1 = (term1*beta)/(term2*alpha);
        c0 = (term0*beta)/(2.0*term2*alpha);
        breakaway = cubic( c2, c1, c0);
        maxval = (double) max( (double) breakaway.real[1], (double) max( (double)
breakaway.real[0], (double) breakaway.real[2]));
        return( maxval );

}
```

```
/*
*       4th order Runge Kutta Function to solve 3rd order differential
*       equations.  The equations are of the form:
*
*               x' = v
*               v' = a
*               a' = -ma - cv - kx + f(x,v,a,t)
*
*       The function f(x,v,a,t) is a pointer to a specific function.  In
*       this way, this routine can be used to evalute many functions.
****************************************************************************
*/
double          run3( double t,  double h)
{
        double k1, k2, k3, k4, l1, l2, l3, l4, m1, m2, m3, m4;

        k1 = h*f(x, v, a, t);
        l1 = h*v;
        m1 = h*a;

        k2 = h*f(x + l1/2.0, v + m1/2.0, a + k1/2.0, t + h/2.0);
        l2 = h*(v + k1/2.0);
        m2 = h*(a + m1/2.0);

        k3 = h*f(x + l2/2.0, v + m2/2.0, a + k2/2.0, t+ h/2.0);
        l3 = h*(v + k2/2.0);
        m3 = h*(a + m2/2.0);

        l4 = h*(v + l3);
        m4 = h*(a + m3);
        k4 = h*f(x + l3, v + m3, a + k3, t + h);

        x += (double) ((l1 + 2.0*l2 + 2.0*l3 + l4)/6.0);
        v += (double) ((m1 + 2.0*m2 + 2.0*m3 + m4)/6.0);
        a += (double) ((k1 + 2.0*k2 + 2.0*k3 + k4)/6.0);

        return(x);

}
```

```
/*
*       simpsons_rule( n, h, x_n)
*
*       DESCRIPTION:
*
*       Uses simpsons rule to calculate an integral
*
*       VARIABLES:
*
*               n               number of subdivisions on x axis
*               h               runge kutta step size
*               x_n             dependent value to start with
*               e[j]            error signal evaluated at subdivision
*               s0,s1,s2        intermediate simpson's rule variables
*               j               counter
*
*       PSEUDOCODE:
*
*               FOR( j= 0 to j= number of subdivisions)
*                       calculate error signal at each subdivision
*               ENDFOR
*               calculate integral
*               return integral
*
*****************************************************************/
double          simpsons_rule(int n, double h, double x_n)
{
        int     j;
        double          s0, s1, s2;
        double          e[103];

        for( j = 0; j <= n; j++)
        {
                e[j] = r_of_t - c_of_t;
                if (j == n)
                        break;
                r_of_t = rmax*(1.0 - exp( -z*(x_n+h) ));
                c_of_t = run3(x_n, h);
                x_n += h;
        }
        s0 = 0.0;
        s1 = 0.0;
        s2 = 0.0;
```

```
	s0 = e[0] + e[n];
	for (j=1; j <= n-1; j+=2)
		s1 += e[j];
	for (j=2; j <= n-2; j+=2)
		s2 += e[j];

	return( h*(s0 + 4.0*s1 + 2.0*s2)/3.0 );
}
```

```
/*
*     cntrl_sgnl()
*
*     DESCRIPTION:
*
*     This function determines the control signal based upon the
*     equation:
*                      /                            dc(t)
*          v(t) = Ki  |  e(t) - kd1 c(t) - kd2-------
*                    /                              dt
*     where e(t) is the error signal, c(t) and dc(t)/dt, are the
*     output signal and it's derivative.  The PDF gains are ki, kd1,kd2.
*     Simpson's rule is used to calculate the integral and the fourth
*     order Runge Kutta method is used to calculate the control signal
*     and it's derivative.  At any point, if the control signal exceeds
*     the input maximum or minimum limit, then the procedure stops and
*     returns to the calling routine.
*
*     VARIABLES:
*
*     (global)
*          alpha           first-order numerator term
*          beta            zero-order numerator term
*          inertia         second-order denominator term
*          damping                 first-order denominator term
*          spring          zero-ordre denominator term
*          kd2, kd1, ki    PDF gains
*          ka, kb          feedback and amplifier gains
*          term3, term2    characteristic equation terms
*          term1, term0
*          char_roots      cubic root structure which holds
*                          the characteristic roots
*          r_of_t          input step function value
*          c_of_t          output system response
*          dc_of_t                 derivative of output system response
*          v_of_t          control signal
*          f               pointer to R.K. function to evaluate
*          x, v, a         Runge Kutta dependent variables
*          xchart[]        array of x values for plotting
*          ychartn[]       arrays of y values for plotting
*          dom_tc                  dominant root time constant
*
*
*     (local)
```

```
*          integral           value of integral of error, returned from
*                             simpsons rule function
*          stepsize           area over which to evaluate integral
*          t1                 independent variable, time
*          domroot,smlroot         based upon characteristic roots
*          i, j               counters
*
*     PSEUDOCODE:
*
*     determine dominant root based upon characteristic eqn
*     select Runge Kutta stepsize and length of response based on root
*     FOR ( time = 0 seconds to time = 6 dominant time constants)
*          calculate v(t) using simpsons rule and runge kutta
*          IF ( v(t) saturates )
*               return to calling program
*          ENDIF
*          IF ( 10 iterations have occured)
*               record chart variables to be plotted
*          ENDIF
*     ENDFOR
*     return the number of points to plot
*
*************************************************************************/
int   cntrl_sgnl()
{
      double        stepsize, integral = 0.0;
      double        t1=0.0;
      double        domroot, smlroot;
      int     i = 0;
      int     j;
      f = f1;

      integral = 0.0;

      term3 = inertia + alpha*kd2*kb;
      term2 = (damping + alpha*kd1*kb + beta*kd2*kb)/term3;
      term1 = (spring + alpha*ki*ka*kb + beta*kd1*kb)/term3;
      term0 = (ki*ka*kb*beta)/term3;

      char_roots = cubic( term2, term1, term0);

      x = 0.0;
      v = 0.0;
      a = 0.0;
```

```
	domroot = max( char_roots.real[1], max( char_roots.real[0],
char_roots.real[2]));
	smlroot = min( char_roots.real[1], min( char_roots.real[0], char_roots.real[2]));

	z = -10.0*smlroot;

	dom_tc = -1.0/domroot;

	numdy = (ki*ka*kb*alpha)/term3;

	stepsize = dom_tc/1000.0;

	r_of_t = 0.0;
	c_of_t = 0.0;

	xchart[0] = ychart1[0] = ychart2[0] = 0.0f;

	j = 1;
	for( t1 = 0.0; t1 <= 6.0*dom_tc; t1 += stepsize)
	{
		integral += simpsons_rule( 10, stepsize/10, t1);
		dc_of_t = v;
		v_of_t = ki*integral - kd1*c_of_t - kd2*dc_of_t;

		if (!stepback)
		{
			if (!check_actuator(v_of_t))
				return(0);
		}

		if (++i == 10)
		{
			xchart[j] = (float) (t1 + stepsize);
			ychart1[j] = (float) v_of_t;
			ychart2[j] = (float) c_of_t;
			ychart3[j] = (float) r_of_t;
			i = 0;
			j++;
		}
	}
	return(j);
}
```

BIBLIOGRAPHY

Astrom, K. J. and T. Hagglund. *Automatic Tuning of PID Controllers.* Research Triangle Park, NC: Instrument Society of America, 1988.

Boyd, S., Craig Barratt, and Stephen Norman. "Linear Controller Design: Limits of Performance Via Convex Optimization." Proceedings of the IEEE, March 1990, pp. 529-574.

Electrocraft Corp. *DC Motors, Speed Controls, Servo Systems.* 5th edition. Hopkins, MN, 1980.

Kreyszig, E. *Advanced Engineering Mathematics.* New York: John Wiley & Sons, 1983.

Palm, W. *Control Systems Engineering.* New York: John Wiley & Sons, 1986.

Phelan, R. *Automatic Control Systems.* Ithaca, New York: Cornell University Press, 1977.

Towill, D.R. *Transfer Function Techniques for Control Engineers.* London: Iliffe Books LTD, 1970.

Schwarzenback, J. and K.F. Gill. *System Modelling and Control.* New York: John Wiley & Sons, 1978.

Ulsoy, A.G. "Optimal Pseudo-Derivative Feedback Control." M.S. thesis. Cornell University 1975.