

Computer Control Design and Modeling

Lecture Notes

Developed By:
akeel abdual aziz

Computer control design and modeling

Lecture 1

Introduction

A system or a process or a plant is a segment of environment that is under consideration.

Control is a term that describes the process of forcing a system to behave in a desired way in order to achieve certain objective(s)/goal(s).

Examples.

- Automobile steering control.
- Thousands of industries consider control in some or the other form such as quality control, production control, temperature control, pollution control, Precision control, etc.
- Robot control.
- Human body implements highly sophisticated control schemes for numerous purposes such as body temperature regulation, hormone level control, etc.

Control Engineering is not restricted to one field of engineering but equally applicable to different branches of engineering such as mechanical, chemical, civil, computer, electrical etc.

Control systems

Typically, a system (process) has one or more inputs and one or more outputs, which can be represented by a block, as shown in figure 1.

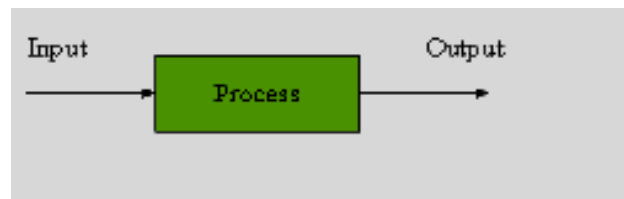


Figure 1: Typical representation of a process

LECT. 1

Control System is an interconnection of components forming a system configuration that will provide a desired system response.

Open-loop and closed-loop control systems

Depending on configuration, control systems can be categorized into mainly two classes:

- i) open-loop control systems;
- ii) closed-loop (or feedback) control systems.

Open-loop control systems

An open-loop control system utilizes a controller or actuating device to obtain a desired response directly without using feedback.

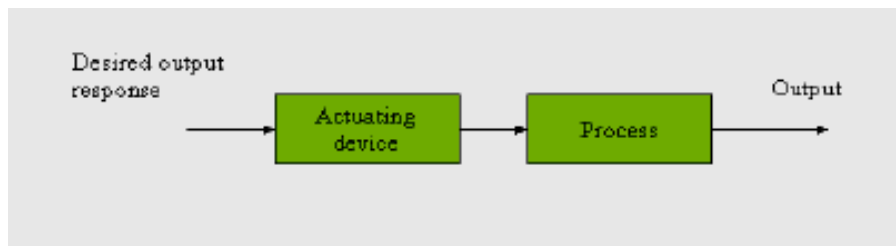


Figure 2: Open-loop control system

Example of an open loop control system: room temperature control. The structure of this system is presented in figure 3. In this system the inlet vent temperature is the input (control signal), and the output (controlled variable) is the room air temperature. The actuator comprises of the furnace and a pre-programmed on-off switch that triggers the furnace, which in turn activates the inlet vent temperature. The ambient temperature acts as a disturbance.

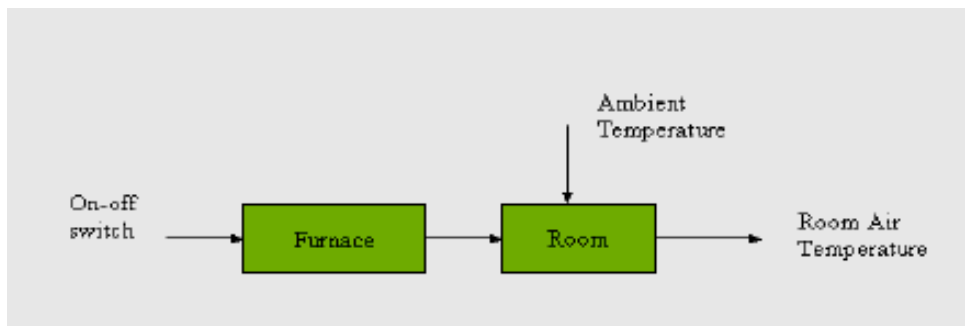


Figure 3: Example of an open loop control system: room temperature control

LECT. 1

The open-loop control system cannot adjust to changes of the ambient temperature.

Example of an open loop control system: controlling the position of a missile launcher from a remote location. This system is illustrated in figure 4. The input is the desired angular position of the missile launcher, and the control system consists of potentiometer, power amplifier, motor, gearing between the motor and missile launcher, and the missile launcher. For accurate positioning, the missile launcher should be precisely calibrated with reference to the angular position of the potentiometer, and the characteristics of the potentiometer, amplifier and motor should remain constant. Except for the potentiometer, the components that comprise this open loop control system are not precision devices. Their characteristics can easily change and result in false calibration and poor accuracies. In practice, simple open-loop control systems are never used for the accurate positioning of fire-control systems because of the inherent possibility of inaccuracies and the risks involved.

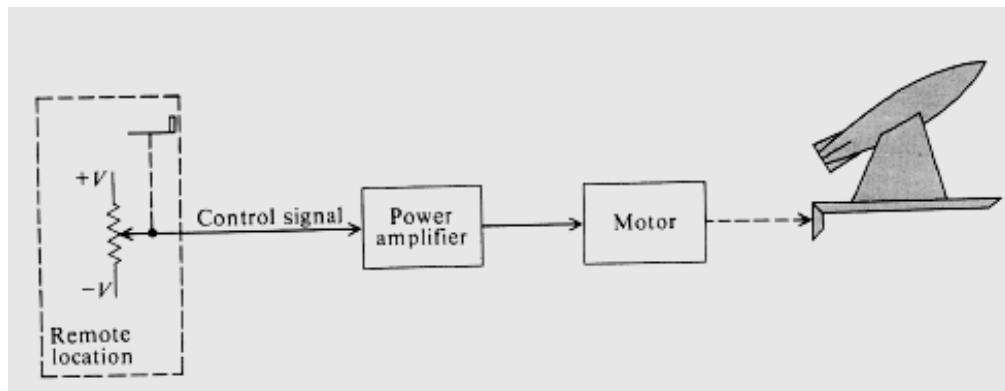


Figure 4: Example of an open loop control system: controlling the position of a missile launcher from a remote location

Closed-loop (feedback) control systems

The structure of a simple closed-loop feedback system is shown in figure 5. In contrast to an open-loop control system, a closed-loop utilizes the additional measure of the actual output to compare the actual output with the desired output (reference or command). This additional measure of the output is called the feedback.

LECT. 1

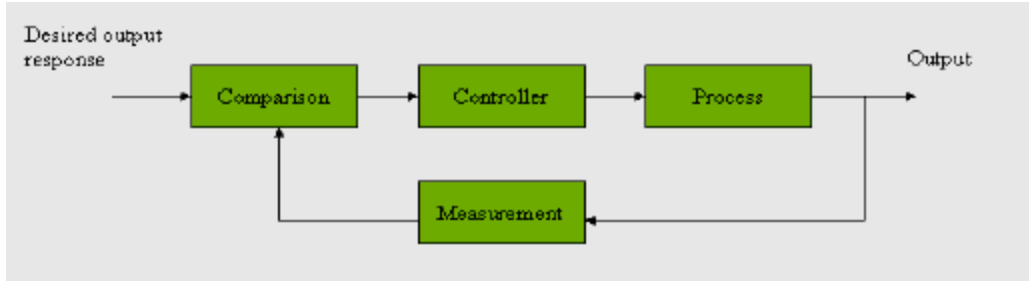


Figure 5: Closed-loop control system

Example of a closed-loop control system: room temperature control.

The previous example is now modified by incorporating the measure of the output, i.e. the room air temperature. The controller is the *thermostat* that takes into account a reference signal and the output feedback in order to set the switch position. Ideally, the thermostat should trigger the switch as soon as the error is negative and switch off when the room air has reached/exceeded the reference temperature. The features of this scheme are:

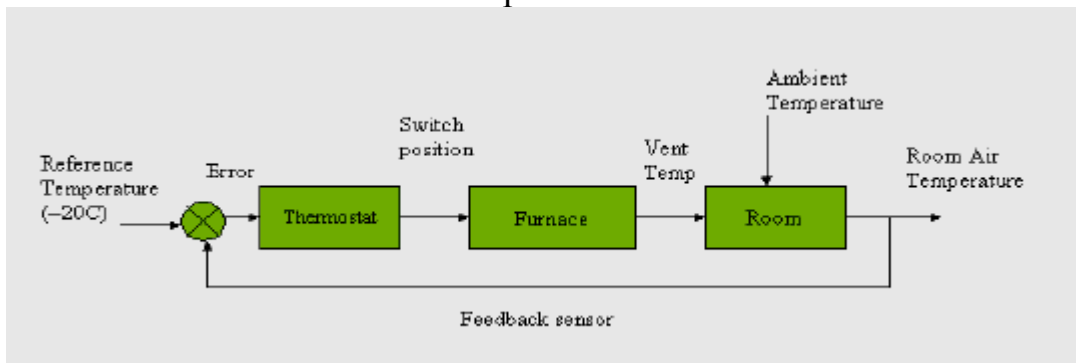


Figure 6: Example of a closed-loop control system: room temperature control

- If room temperature $<$ reference temperature, the furnace is switched on automatically, until room temperature \geq reference temperature.
- Can handle changes in the system (e.g. change in ambient temperature).
- It is negative feedback (the measured output is subtracted from the reference signal).

Example of a closed-loop control system: controlling the position of a missile launcher from a remote location

The previous example is modified by introducing a position feedback loop. This feedback loop consists of potentiometer R_2 and the difference amplifier. Should an error exist, it is amplified and applied to a motor drive which

LECT. 1

adjusts the output-shaft position until it agrees with the input-shaft position, and the error is zero.

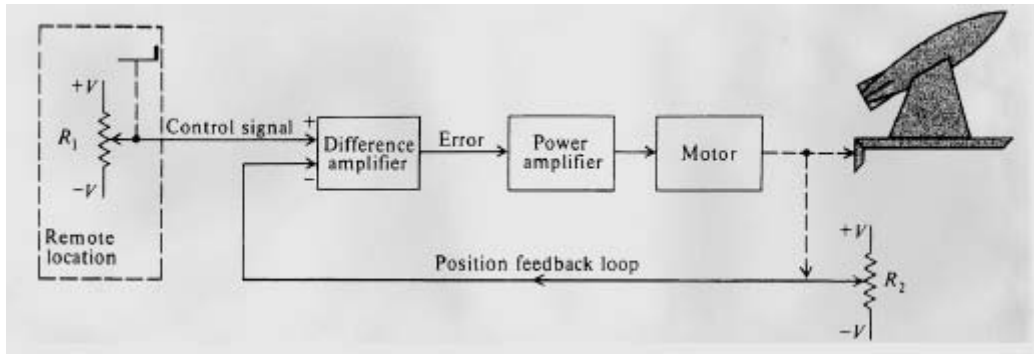
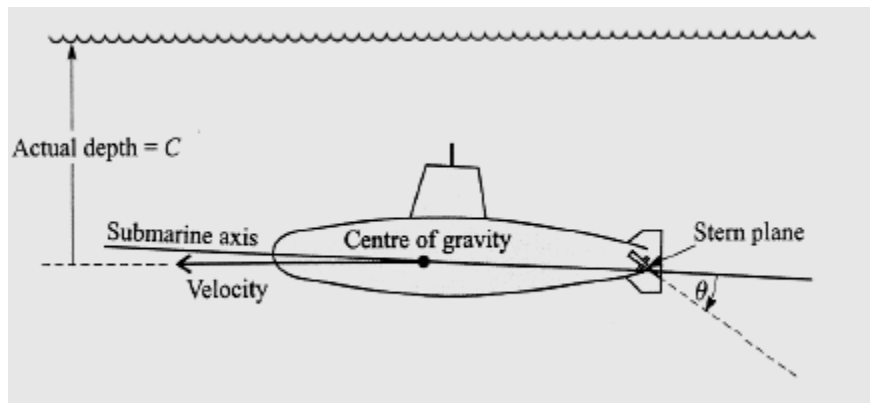


Figure 7: Example of a closed-loop control system: controlling the position of a missile launcher from a remote location

Feedback control systems used to control position, velocity, and acceleration are very common in industrial applications.

The important feature of the using of feedback is that the feedback control system can handle changes in the system. On the other hand, improper use of feedback can make the system unstable, so the stability issue arises.

Example of a closed-loop control system: Automatic depth control of a submarine



LECT. 1

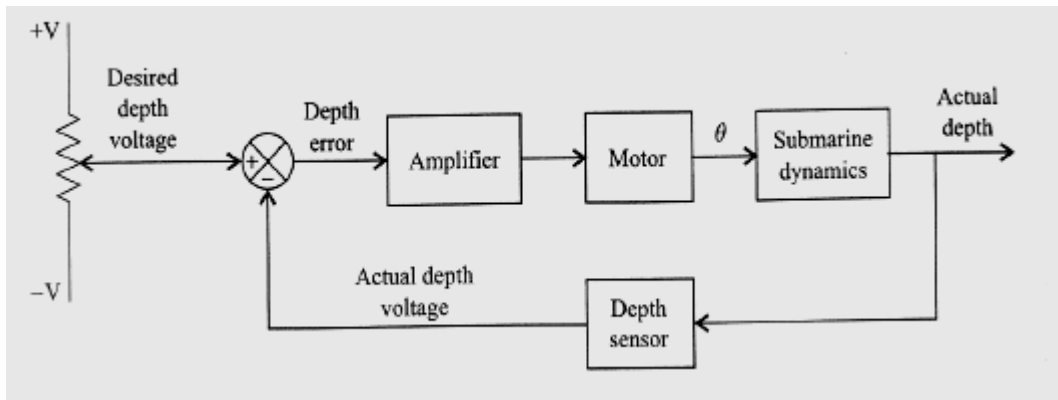


Figure 8: Automatic depth control of a submarine

Suppose the captain of the submarine wants the submarine to "hover" at a desired depth, and sets the desired depth as a voltage from calibrated potentiometer. The actual depth is measured by a pressure transducer which produces a voltage proportional to depth. The difference between the desired and the actual depth is amplified which then drives a motor that rotates the stern plane actuator angle θ in order that the stern plane rotation reduces the depth error of the submarine to zero.

Given a process, there are three steps to design a feedback control system which are:

1. **Modeling.** Obtain mathematical description of the system.
2. **Analysis.** Analyze the properties of the system.
3. **Design.** Given a plant, design a controller based on performance specifications.

End of Lecture One

Computer control design and modeling

Lecture 2

Mathematical models, Linear Systems and Linearization

Mathematical Models of Physical Systems

To understand and control complex systems, we must obtain mathematical models of these systems. The term **mathematical model**, in the control engineering perspective, implies a set of differential equations that describe the dynamic behavior of a process.

The set of differential equations that describe the behavior of physical systems are typically obtained by utilizing the physical laws of the process. These types of models are often called **first principles models**. Several examples of first-principles models are considered below.

Models of simple mechanical systems

The equations of a mechanical system may be obtained by a direct application of Newton second law.

Example: Ideal Mass-Spring System. This system is shown on figure 1.

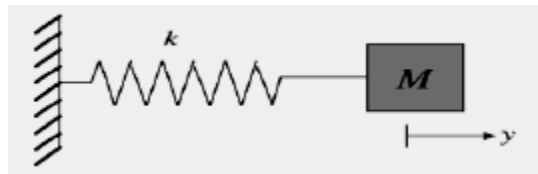


Figure 1: Ideal Mass-Spring System

In order to write the equation of motion, we consider the set of forces acting on the mass M . The force F from the spring acts against the displacement and is proportional to the displacement of the spring, i.e. $F = -ky$, where k is spring constant. Assume a frictionless surface. Applying Newton's second law, one can get the following equation of motion

or

$$M \frac{d^2 y}{dt^2} = -ky,$$

$$M\ddot{y} + ky = 0.$$

(1)

LECT. 2

The equation (1) is a second-order *linear* differential equation.

Example: Mass-Spring System with a Damper. The previous example is now modified by the addition of a damper with resistance R . Additionally; we assume that the body is subjected to an external force F_e .

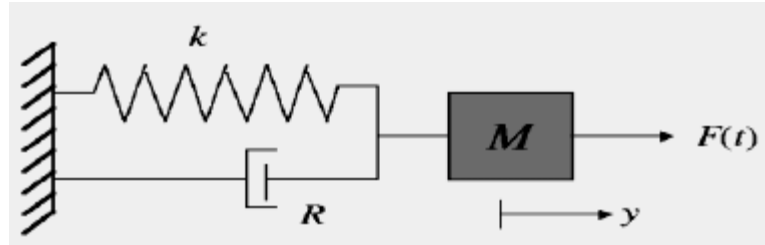


Figure 2: Mass-Spring System with a Damper

The damping force F_d is assumed to be proportional to the velocity \dot{y} and acts against the motion of the body, i.e.

$$F_d = -R\dot{y}.$$

Such a damping is called *viscous damping*. Using Newton's second law, we can write the equations of motion as follows:

$$M \frac{d^2 y}{dt^2} = F_e(t) - ky - R \frac{dy}{dt},$$

or

$$M\ddot{y} + R\dot{y} + ky = F_e(t). \quad (2)$$

Modeling of RLC Electrical Systems

Consider RLC electrical circuit shown in figure 3.

It consists of a source of current $r(t)$, a resistor characterized by its resistance R , a capacitor characterized by its capacitance C , and an inductor with inductance L . Differential equation of the RLC circuit can be obtained by utilizing *Kirchhoff's laws*, and the voltage-current relationships for R , L , and C . Indeed, *Kirchhoff's current law* implies that

$$r(t) = i_r + i_c + i_l,$$

LECT. 2

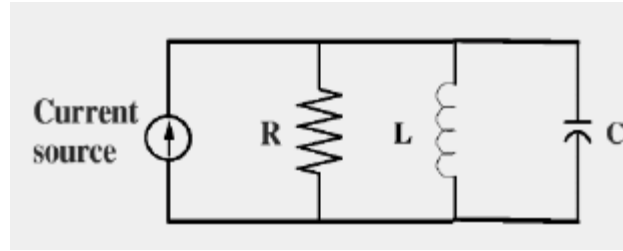


Figure 3: RLC Circuit

where i_r , i_c , and i_l are currents through the resistor, the capacitor, and the inductor respectively. On the other hand, *Kirchhoff's voltage law* implies in this particular case that the voltage v across any of the elements R , L , or C is the same. We have the following voltage-current relationships for R , L , and C :

$$\begin{aligned} i_r &= \frac{v}{R}, \\ i_c &= C \frac{dv}{dt}, \\ i_l &= \frac{1}{L} \int_0^t v(s) ds. \end{aligned}$$

Thus, we get the following equation of RLC circuit

$$\frac{v(t)}{R} + C \frac{dv(t)}{dt} + \frac{1}{L} \int_0^t v(s) ds = r(t). \quad (3)$$

It is worth noting that the equation (3) is analogous to the equation (2) of the Mass-Spring System with Damper. Indeed, if we rewrite the equation (2) in terms of the velocity $v(t) = \frac{dy(t)}{dt}$, we get

$$M \frac{dv}{dt} + Rv(t) + k \int_0^t v(s) ds = F_e(t). \quad (4)$$

The equations (3) and (4) are of equivalent form, and this fact represents so called *velocity-voltage analogy* between mechanical and electrical systems.

This equivalence between systems is beneficial to the analyst in understanding multidisciplinary systems that are similar to each other. The main advantage is that the solution to one system can be extended to all the analogous systems governed by the same set of differential equations. Therefore, a mechanical engineer can immediately extend the knowledge

LECT. 2

gained about the analysis and design of mechanical engineering systems to that of analogous systems in the other branches of engineering.

Linear Systems

Linear systems represent a very important class of systems. To introduce the notion of a linear system, consider a system represented by its block diagram, as shown in the figure 4.

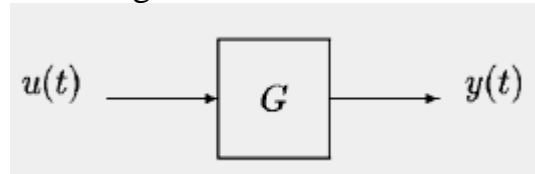


Figure 4: Representation of a system

Below, by $G(u)$ we will denote the output (reaction) of the system corresponding to given input (excitation) u .

Definition 1. System G is linear, if and only if

- i) It obeys **the superposition principle**,

$$G(u_1 + u_2) = G(u_1) + G(u_2)$$

for any possible inputs u_1, u_2 .

- ii) It obeys the principle of homogeneity, i.e.

$$G(\gamma u) = \gamma G(u)$$

for any possible input u and any constant $\gamma \in R$.

In essence, G is linear if and only if

$$G(\alpha u_1 + \beta u_2) = \alpha G(u_1) + \beta G(u_2)$$

for any possible inputs u_1, u_2 , and any constants $\alpha, \beta \in R$.

Otherwise, the system is called *non-linear*.

The principle of superposition, which applies to linear systems, is one of the most powerful tools in systems analysis. It allows us to say that the response of a system to sum of inputs is equal to the sum of the responses of the system to the inputs taken individually. This has very deep implications for

LECT. 2

analysis. If a complicated input to a linear system can be represented as a sum of simpler inputs, then the response of the system to the simpler inputs can be calculated separately and then added to get the response of the system to the complicated input.

Example. The function $y(x) = kx$, where k is a constant, is a linear function. Indeed, $k(\alpha x_1 + \beta x_2) = \alpha kx_1 + \beta kx_2$. The function $y = x^2$ is a nonlinear function. Indeed, in general $(x_1 + x_2)^2 \neq x_1^2 + x_2^2$.

Linear dynamical systems

Consider a dynamical system described by an ordinary differential equation of the form

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_{n-1} y = f(t). \quad (5)$$

The system (5) is an example of linear dynamical systems. However, the notion of linearity, in the sense of the above definition, cannot be directly applied to dynamical systems of the form (5). Indeed, a solution of the system (5) is not uniquely determined by input $f(t)$, but also depends on initial conditions $y(0), \frac{dy}{dt}(0), \dots, \frac{d^{n-1}y}{dt^{n-1}}(0)$. For linear *dynamical* systems, a more general version of superposition principle must be satisfied, which also includes some form of linearity with respect to initial states. For the purposes of this subject, however, the following slightly informal definition of a linear dynamical system will be sufficient.

Definition 2. A dynamical system is called linear if and only if it can be described by linear differential equations.

Linearization

All real life systems are nonlinear. However, almost all physical systems can be closely approximated by linear models within some range of the variables. The main reason to use linear models is that linear models make the analysis and design problems much simpler in terms of understanding and applicability. The process of finding a linear model which gives good approximation of given nonlinear model is known as **linearization**.

LECT. 2

To illustrate the process of linearization, consider the following very simple example. Consider the function

$$f(x) = x^2.$$

Suppose we need to find a linear function which approximate $f(x)$ near the point $x_0 = 1$. Clearly (see the figure), this linear approximation is given as follows

$$y = f(x_0) + \left. \frac{df}{dx} \right|_{x_0} (x - x_0) = 1 + 2(x - x_0).$$

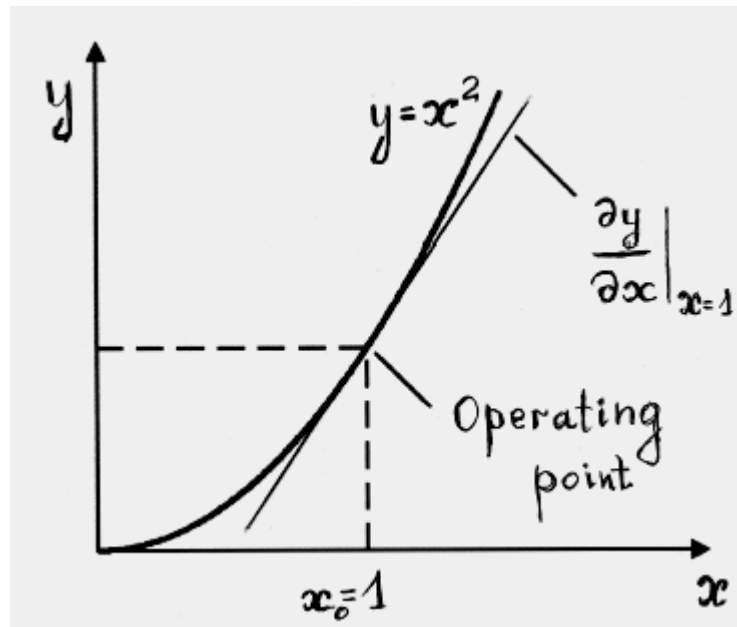


Figure 5: Linearization of the function $y = x^2$ near the point $x_0 = 1$

In general, linearization can be obtained using Taylor series expansion about the operating point. Suppose a (nonlinear) function $y = f(x)$ is given, then Taylor series expansion about the point x_0 is as follows

$$y = f(x_0) + \left. \frac{df}{dx} \right|_{x=x_0} (x - x_0) + \left. \frac{d^2 f}{dx^2} \right|_{x=x_0} \frac{(x - x_0)^2}{2!} + \dots$$

Neglecting the higher-order terms, we get the linear approximation of f near the point x_0 :

$$y = f(x_0) + \left. \frac{df}{dx} \right|_{x=x_0} (x - x_0).$$

LECT. 2

Example: linearization of the pendulum equations. The pendulum is described by the following equation

$$\ddot{q} = -mgl \sin q.$$

In the point $q = 0$ the torque $T = mgl \sin q = 0$. The linear approximation of the torque about the point $q = 0$ is given as follows

$$T = mgl \left. \frac{d \sin q}{dq} \right|_{q=0} (q - 0) = mgl \cos(0)q = mglq.$$

Therefore, the linearized equations of the pendulum about the point $q = 0$ is as follows

$$\ddot{q} = -mglq.$$

End of Lecture Two

Computer control design and modeling

Lecture 3

Solution of linear differential equations using Laplace transform

In general, the procedure of solving linear differential equation using Laplace transform consists of the following three steps:

- Take Laplace transform of each term in the differential equation. This step eliminates time and all of the time derivatives from the original equation and results in an algebraic equation in s .
- Solve the resulting algebraic equation for the transform of the desired time function.
- Obtain the inverse Laplace transform. This last step gives the solution of the differential equation.

Example: Ideal mass-spring system. The system is described by the following linear differential equation

$$M\ddot{y} + ky = 0, \quad (1)$$

where M is mass and k is spring constant.

Problem. Find the solution of (1) corresponding to initial conditions $y(0) = y_0$, $\dot{y}(0) = 0$.

Solution. Applying Laplace transform to both sides of the equation (1), one can write

$$Ms^2Y(s) - Msy(0) - M\dot{y}(0) + kY(s) = 0.$$

Taking into account initial conditions, we get

$$Ms^2Y(s) - Msy_0 + kY(s) = 0. \quad (2)$$

Equation (2) is an algebraic one. Its solution is

$$Y(s) = \frac{Msy_0}{Ms^2 + k} = y_0 \frac{s}{s^2 + \frac{k}{M}}.$$

Using Laplace transform table, one can easily find the inverse Laplace transform of $Y(s)$:

$$y(t) = y_0 \cos\left(\sqrt{\frac{k}{M}}t\right). \quad (3)$$

LECT. 3

Example: forced differential equation. Consider the following differential equation

$$\ddot{y}(t) + 5\dot{y}(t) + 4y(t) = 2e^{-2t}. \quad (4)$$

Problem: Find the solution corresponding to zero initial conditions $y(0) = \dot{y}(0) = 0$.

Solution. Taking the Laplace transform of both sides, we obtain

$$s^2Y(s) - sy(0) - \dot{y}(0) + 5sY(s) - 5y(0) + 4Y(s) = \frac{2}{s+2},$$

and, due to zero initial conditions,

$$s^2Y(s) + 5sY(s) + 4Y(s) = \frac{2}{s+2}$$

Solving for $Y(s)$ yields

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

Partial fraction expansion for $Y(s)$ has the form

$$Y(s) = \frac{C_1}{s+2} + \frac{C_2}{s+1} + \frac{C_3}{s+4}.$$

Residues are

$$C_1 = \left. \frac{2}{(s+1)(s+4)} \right|_{s=-2} = -1,$$

$$C_2 = \left. \frac{2}{(s+2)(s+4)} \right|_{s=-1} = \frac{2}{3},$$

$$C_3 = \left. \frac{2}{(s+2)(s+1)} \right|_{s=-4} = \frac{1}{3}.$$

Therefore,

$$y(t) = -e^{-2t} + \frac{2}{3}e^{-t} + \frac{1}{3}e^{-4t}.$$

The Transfer Function

In this lecture we will formulate the system representation by establishing a definition of a function that algebraically relates a system's output to its inputs. This function allows us to algebraically combine mathematical representations of subsystems to yield a total system representation.

Definition of transfer function

Consider a system described by linear time-invariant differential equation,

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_0 y(t) = b_m \frac{d^m r(t)}{dt^m} + b_{m-1} \frac{d^{m-1} r(t)}{dt^{m-1}} + \dots + b_0 r(t), \quad (5)$$

where $r(t)$ is input, and $y(t)$ is output of the system. Taking the Laplace transform of both sides of equation (5), we get

$$\begin{aligned} & a_n s^n Y(s) + a_{n-1} s^{n-1} Y(s) + \dots + a_0 Y(s) + \left\{ \begin{array}{l} \text{initial conditions} \\ \text{terms involving } y(t) \end{array} \right\} \\ & = b_m s^m R(s) + b_{m-1} s^{m-1} R(s) + \dots + b_0 R(s) + \left\{ \begin{array}{l} \text{initial conditions} \\ \text{terms involving } r(t) \end{array} \right\}, \end{aligned} \quad (6)$$

where $R(s)$ and $Y(s)$ are Laplace transform of $r(t)$ and $y(t)$ respectively. If we assume that all initial conditions are equal to zero, the equation (6) reduces to

$$a_n s^n Y(s) + a_{n-1} s^{n-1} Y(s) + \dots + a_0 Y(s) = b_m s^m R(s) + b_{m-1} s^{m-1} R(s) + \dots + b_0 R(s). \quad (7)$$

The last equation can be rewritten as follows

$$(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0) Y(s) = (b_m s^m + b_{m-1} s^{m-1} + \dots + b_0) R(s). \quad (8)$$

Denote by $G(s)$ the ratio of the output transform $Y(s)$ divided by input transform $R(s)$:

$$G(s) = \frac{Y(s)}{R(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}. \quad (9)$$

The ratio $G(s)$ is called **transfer function** of the system (5). Thus we have the following definition.

LECT. 3

Definition 1. Transfer function of a linear system is the ratio of Laplace transform of the output variable to the Laplace transform of the input variable, with all initial conditions assumed to be zero.

Transfer function is a property of the system elements only and is not dependent on the excitation (input). As we shall see later, the transfer function completely characterizes system's performance.

The transfer function can be represented as a block diagram with input, output, and the system transfer function inside the block, as shown in figure 1.

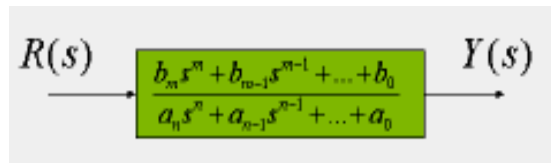


Figure 1: Block Diagram of a Transfer Function

From (9) it follows that

$$Y(s) = G(s)R(s), \quad (10)$$

i.e. under zero initial conditions, the Laplace transform of the output (response) is equal to the product of the transfer function and the Laplace transform of the input.

Example: transfer function. Suppose the system is described by the following differential equation

$$\frac{dy(t)}{dt} + 2y(t) = r(t), \quad (11)$$

where $r(t)$ is input, and $y(t)$ is output. Taking Laplace transform of both sides of (11), and assuming zero initial conditions, we have

$$sY(s) + 2Y(s) = R(s).$$

Therefore, the transfer function of system (11) is as follows

$$G(s) = \frac{Y(s)}{R(s)} = \frac{1}{s+2}.$$

Problem: find the response of (11) to the input $r(t) = u(t)$, where $u(t)$ is unit step function, assuming zero initial conditions.

LECT. 3

Solution. To solve this problem, one can use formula (10). The Laplace transform of unit step function $r(t) = u(t)$ is

$$R(s) = \frac{1}{s}.$$

Therefore, using formula (10), we see that the Laplace transform of the output $y(t)$ is

$$Y(s) = G(s)R(s) = \frac{1}{s(s+2)}.$$

Expanding by partial fractions, we have

$$Y(s) = \frac{k_1}{s} + \frac{k_2}{s+2}.$$

Calculating residues, we get

$$k_1 = \left. \frac{1}{s+2} \right|_{s=0} = \frac{1}{2},$$
$$k_2 = \left. \frac{1}{s} \right|_{s=-2} = -\frac{1}{2},$$

so the partial fraction expansion of $Y(s)$ is

$$Y(s) = \frac{1}{2s} - \frac{1}{2(s+2)}.$$

Applying the inverse Laplace transform, we finally get

$$y(t) = \frac{1}{2}u(t) - \frac{1}{2}e^{-2t}.$$

Example: Transfer function of mass-spring system with a damper. Consider a mass-spring system with a damper (see figure 2).

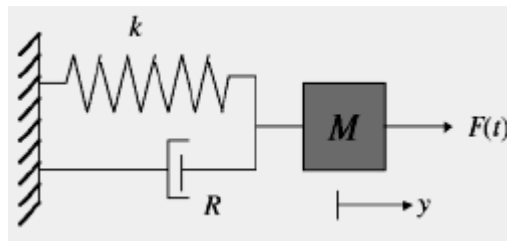


Figure 2: Mass-Spring System with a Damper

It is described by the following differential equation

$$M\ddot{y} + R\dot{y} + ky = F_e(t). \quad (12)$$

Taking Laplace transforms of both sides of (12), and assuming zero initial conditions, we get

LECT. 3

$$Ms^2Y(s) + RsY(s) + kY(s) = \hat{F}_e(s)$$

Where $Y(s) = L(y(t))$ and $\hat{F}_e(s) = L(F_e(t))$. The transfer function of the system is

$$G(s) = \frac{Y(s)}{\hat{F}_e(s)} = \frac{1}{Ms^2 + Rs + k}$$

Impulse response function

Consider a linear system, and suppose all initial conditions are zero. The response of the system to an impulse input signal $\delta(t)$ is called the **impulse response function** of the system. The impulse response function is usually denoted by $h(t)$.

The impulse response function $h(t)$ is closely related with the transfer function of the system. Indeed, let $G(s)$ be the transfer function of the system. Using formula (10), we get

$$\mathcal{L}[h(t)] = G(s)\mathcal{L}[\delta(t)],$$

(zero initial conditions are assumed). But we already proved that $L[\delta(t)] = 1$, therefore

$$\mathcal{L}[h(t)] = G(s).$$

The last equality can be considered as an alternative definition of the transfer function, as follows.

Definition 2. The transfer function of a system is equal to the Laplace transform of the impulse response function.

Problem. Find the transfer function

$$G_1(s) = \frac{X_1}{F(s)}$$

from input $f(t)$ to output $x_1(t)$ of the coupled mass-spring system.

LECT. 3

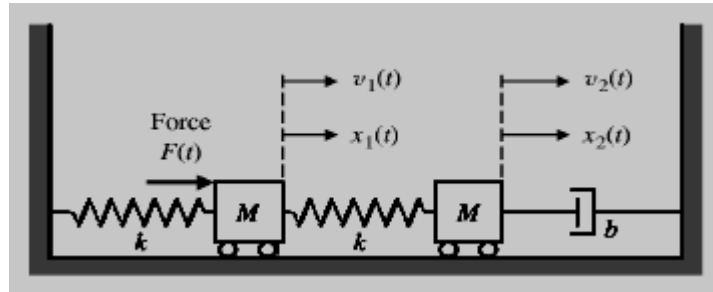


Figure 3: Coupled mass-spring system

Solution. The differential equations governing for the system are

$$M\ddot{x}_1 + kx_1 + k(x_1 - x_2) = f(t), \quad (13)$$

$$M\ddot{x}_2 + k(x_2 - x_1) + b\dot{x}_2 = 0. \quad (14)$$

Applying Laplace transform to both sides of (13), (14), we get the following matrix equation

$$\begin{bmatrix} Ms^2 + 2k & -k \\ -k & Ms^2 + bs + k \end{bmatrix} \begin{pmatrix} X_1(s) \\ X_2(s) \end{pmatrix} = \begin{pmatrix} F(s) \\ 0 \end{pmatrix}. \quad (15)$$

To obtain the transfer function description, one have to solve equation (15) with respect to variables $X_1(s), X_2(s)$. This can be done, for example, as follows. Consider the equation

$$Ax = y, \quad (16)$$

To calculate the solution of (16), one can use so called Cramer's rule. According to Cramer's rule,

$$x_i = \frac{\det A_i}{\det A},$$

where x_i is i-th element of vector x , and A_i is a matrix formed by replacing the i-th column of A by y .

Then

$$X_1(s) = \frac{1}{\Delta} \begin{bmatrix} F(s) & -k \\ 0 & Ms^2 + bs + k \end{bmatrix}$$

$$\therefore X_1(s) = F(s) \frac{Ms^2 + bs + k}{\Delta}$$

LECT. 3

Then

$$G_1(s) = \frac{X_1(s)}{F_s} = \frac{Ms^2 + bs + k}{\Delta} \quad (17)$$

Where Δ is the determinant of the matrix in the left-hand side of the equation (15)

End of Lecture Three

Properties of Laplace Transforms

$F(s)$	$f(t)$	Transform Pair
$\alpha F_1(s) + \beta F_2(s)$	$\alpha f_1(t) + \beta f_2(t)$	Superposition
$F(s)e^{-s\lambda}$	$f(t - \lambda)$	Time Delay ($\lambda \geq 0$)
$\frac{1}{ a } F\left(\frac{s}{a}\right)$	$f(at)$	Time scaling
$F(s + a)$	$e^{-at}f(t)$	Shift in frequency
$s^m F(s) - s^{m-1}f(0) - s^{m-2}f'(0) - \dots - f^{m-1}(0)$	$f^{(m)}(t)$	Differentiation
$\frac{1}{s} F(s)$	$\int f(\zeta) d\zeta$	Integration
$F_1(s)F_2(s)$	$f_1(t) \cdot f_2(t)$	Convolution
$\lim_{s \rightarrow \infty} sF(s)$	$f(0)$	Initial Value Theorem
$\lim_{s \rightarrow 0} sF(s)$	$\lim_{t \rightarrow \infty} f(t)$	Final Value Theorem
$\frac{1}{2\pi j} \int_{(c-j\infty)}^{(c+j\infty)} F_1(\zeta)F_2(s-\zeta)d\zeta$	$f_1(t)f_2(t)$	Time product
$-\frac{d}{ds}F(s)$	$tf(t)$	Multiplication by Time

Table of Laplace Transforms

$F(s)$	$f(t)$	$F(s)$	$f(t)$
1	$\delta(t)$	$\frac{b-a}{(s+a)(s+b)}$	$e^{-at} - e^{-bt}$
$1/s$	$1(t)$	$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}$
$1/s^2$	t	$\frac{a^2}{s(s+a)^2}$	$1 - e^{-at}(1+at)$
$2!/s^3$	t^2	$\frac{(b-a)s}{(s+a)(s+b)}$	$be^{-bt} - ae^{-at}$
$3!/s^4$	t^3	$\frac{a}{s^2+a^2}$	$\sin(at)$
$m!/s^{m+1}$	t^m	$\frac{s}{s^2+a^2}$	$\cos(at)$
$\frac{1}{s+a}$	e^{-at}	$\frac{s+a}{(s+a)^2+b^2}$	$e^{-at}\cos(bt)$
$\frac{1}{(s+a)^2}$	te^{-at}	$\frac{b}{(s+a)^2+b^2}$	$e^{-at}\sin(bt)$
$\frac{1}{(s+a)^3}$	$\frac{1}{2!}t^2e^{-at}$	$\frac{a^2+b^2}{s[(s+a)^2+b^2]}$	$1 - e^{-at}\left[\cos(bt) + \frac{a}{b}\sin(bt)\right]$
$\frac{1}{(s+a)^m}$	$\frac{1}{(m-1)!}t^{m-1}e^{-at}$		
$\frac{a}{s(s+a)}$	$1 - e^{-at}$		
$\frac{a}{s^2(s+a)}$	$\frac{1}{a}(at - 1 + e^{-at})$		

Euler's Identity

$$\sin(\omega t) = \frac{e^{j\omega t} - e^{-j\omega t}}{2j} \quad \cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2}$$

Computer control design and modeling

Lecture 4

Block Diagram Models

Graphically, a linear system can be represented as a block diagram with input, output, and the system transfer function inside the block, as shown in figure 1.

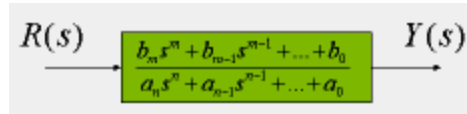


Figure 1: Block Diagram of a Transfer Function

In this lecture, we will learn more about block diagram representation of linear systems.

A **block diagram** of a system is a pictorial representation of the functions performed by components of the systems and the flow of signals between the components of the system. Obviously, there is no more information in the block diagram than in the set of simultaneous equations that represents the system; however, the block diagram depicts the same information much more concisely.

Example. Block diagram of a two-input, two-output system. Consider a linear system with two inputs and two outputs as shown below:

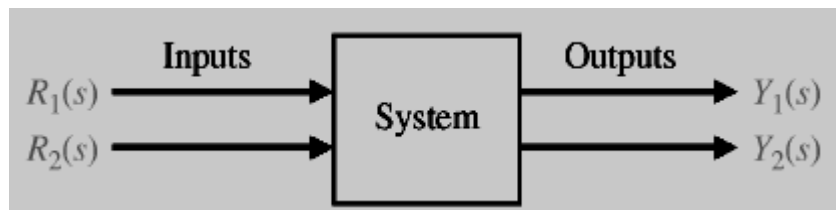


Figure 2: Two-input, two-output system

Using transfer function relations, this system can be written as follows

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} R_1(s) \\ R_2(s) \end{bmatrix}, \quad (1)$$

where G_{ij} is the transfer function relating the i th output variable to the j th input variable. Then, the corresponding block diagram is shown in figure 3.

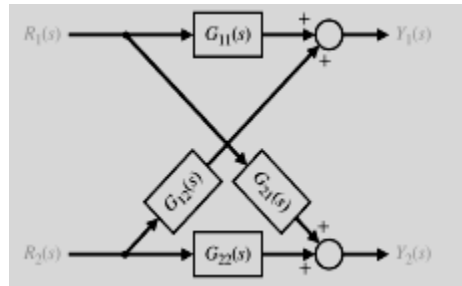
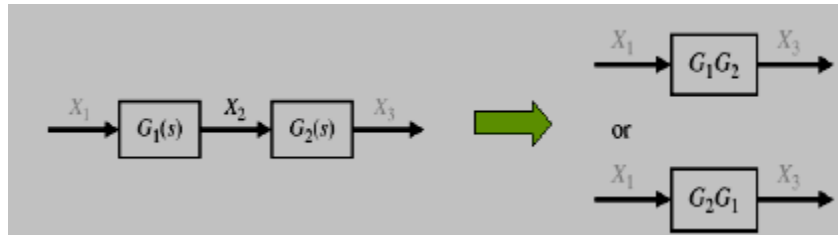


Figure 3: Block diagram of two-input, two-output system

Block diagram transformations

In practice, most of the control engineering systems involve variables that can be heavily interrelated. A complicated block diagram involving many blocks, summing points, and pickoff points can be reduced to a single equivalent block by a set of transformations. Below, we consider several examples of elementary block diagram transformations. All these transformations can be derived by simple algebraic manipulation of the equations represented the blocks.

Combining blocks in cascade



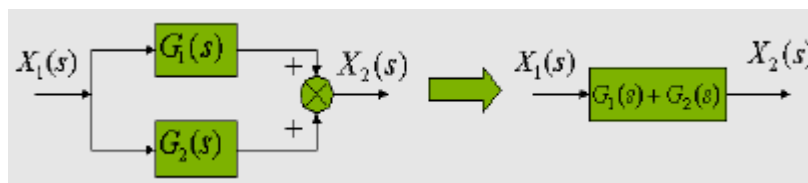
The underlying algebraic derivation is as follows: if $X_2(s) = G_1(s)X_1(s)$, and $X_3(s) = G_2(s)X_2(s)$, then

$$X_3(s) = G_2(s)G_1(s)X_1(s),$$

or (due to commutativity of multiplication)

$$X_3(s) = G_2(s)G_1(s)X_1(s).$$

Parallel subsystem



LECT. 4

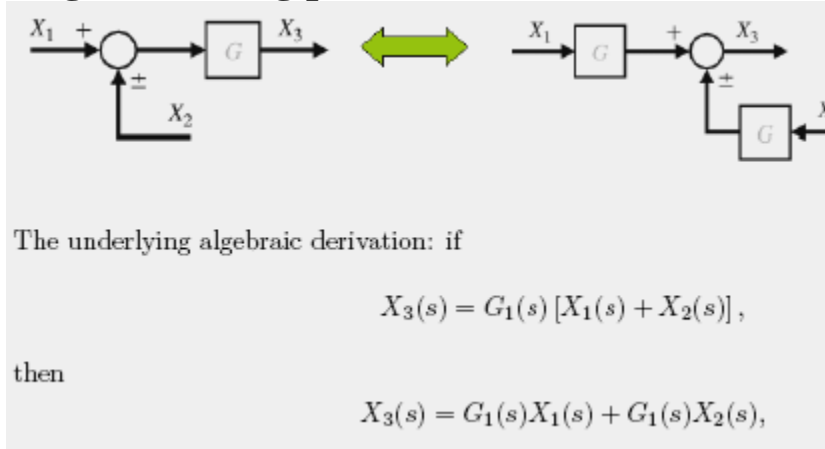
Algebraically, we have

$$X_2(s) = G_1(s)X_1(s) + G_2(s)X_1(s),$$

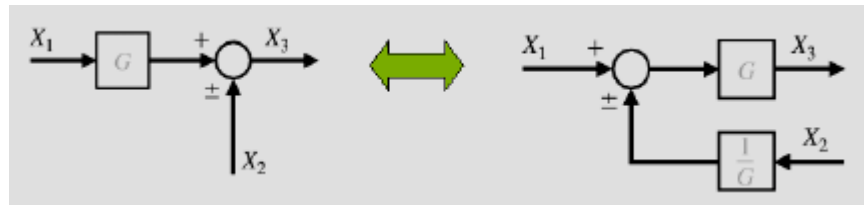
therefore

$$X_2(s) = [G_1(s) + G_2(s)] X_1(s).$$

Moving a summing point behind a block



Moving a summing point ahead of a block



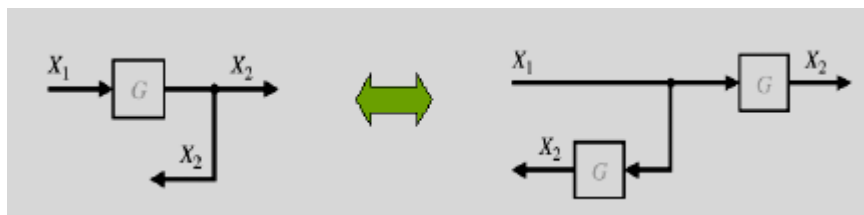
Algebraically, if

$$X_3 = G_1(s)X_1(s) + X_2(s),$$

then

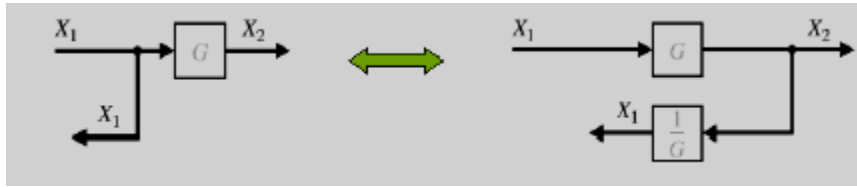
$$X_3(s) = G_1(s) \left[X_1(s) + \frac{1}{G_1(s)} X_2(s) \right].$$

Moving a pick-off point ahead of a block



The transformation is algebraically trivial.

Moving a pick-off point behind a block



Algebraically, if $X_2(s) = G_1(s)X_1(s)$, then

$$X_1(s) = \frac{1}{G_1(s)}X_2(s).$$

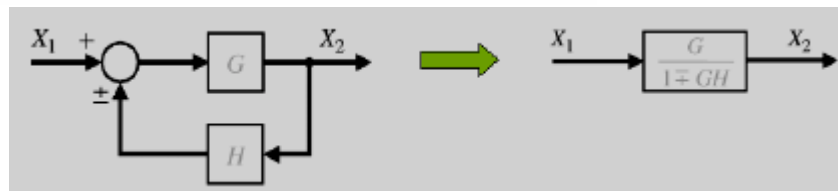
Eliminating a (negative or positive) feedback loop

To obtain the algebraic expression for transfer function of a negative feedback loop, one can write

$$X_2 = G(s) [X_1(s) - H(s)X_2(s)],$$

therefore

$$[1 + G(s)H(s)] X_2 = G(s)X_1(s),$$



and finally

$$\frac{X_2(s)}{X_1(s)} = \frac{G(s)}{1 + G(s)H(s)}. \quad (2)$$

Following the same line of reasoning, one can easily see that the transfer function of a positive feedback loop is given by the formula

$$\frac{X_2(s)}{X_1(s)} = \frac{G(s)}{1 - G(s)H(s)}. \quad (3)$$

The formulas (2), (3), are particularly important, because they represent many of the existing practical control schemes.

Block Diagram Reduction

The block diagram transformations described before allow us to reduce a block diagram of multiple subsystems to a single block representing the transfer function from input to output.

Example: block diagram reduction. Consider a multiple loop feedback control system shown in figure 4.

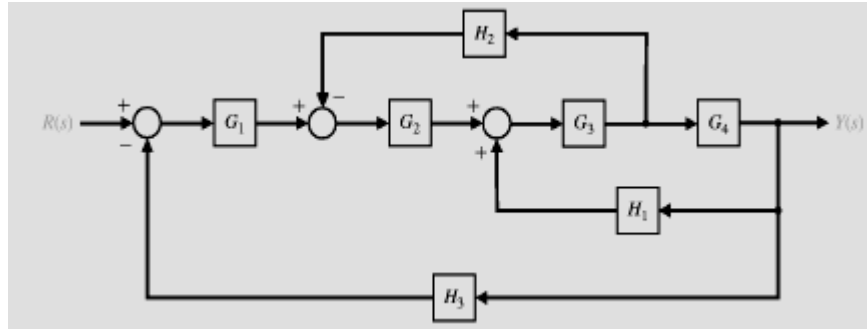


Figure 4: Multiple loop feedback control system

Problem. Find transfer function of the system $G(s) = \frac{Y(s)}{R(s)}$

Solution. The solution can be obtained, for example, as follows.

Step 1. Move H_2 behind block G_4 . The result is presented on figure 5.

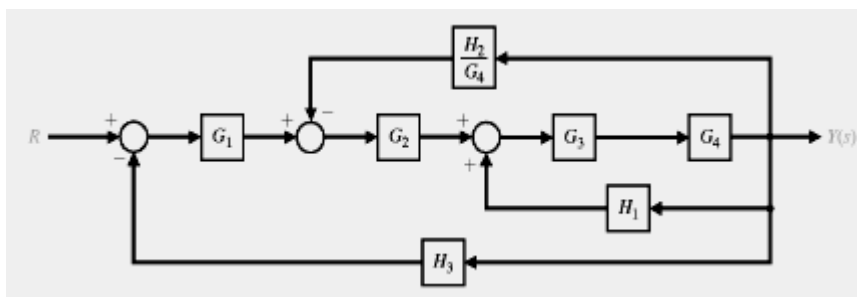


Figure 5: Step 1

Step 2. Eliminate the (positive) feedback loop $G_3G_4H_1$ to obtain the system presented in figure 6.

LECT. 4

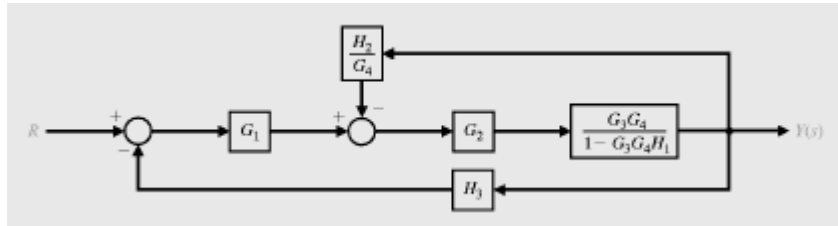


Figure 6: Step 2

Step 3. Eliminate the (negative) feedback loop containing $\frac{H_2}{G_4}$. The result is presented in figure 7.

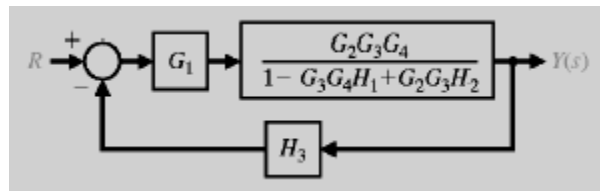


Figure 7: Step 3

Step 4. Obtain the transfer function by eliminating the (negative) feedback loop containing H_3 (see figure 8).

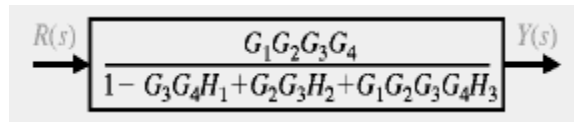


Figure 8: Step 4

Remarks

- The advantage of the block diagram approach is that it provides the engineer with a graphical representation of the system and the relationships between the input and output variables.
- In general, the block diagram reduction process is not unique, i.e. there can be multiple solutions to a block diagram reduction problem (with the same final result).

End of Lecture Four

Computer control design and modeling

Lecture 5

Signal-Flow Graph Models

In the previous lecture, we considered the block diagram representation of a linear system, and studied the block diagram reduction technique. Block diagrams are good for the representation of the interrelationships of controlled and input variables. However, it can get tedious and cumbersome as the block diagrams become more and more complex. An alternative method for determining the relationships between system variables has been developed by Mason and is called signal-flow graph method. Signal-flow graph models are designed to handle many more variables with greater ease than block diagrams. The main advantage of the signal-flow graph method is the availability of a flow graph gain formula, which provides the relation between system variables without requiring any reduction procedure or manipulation of the flow graphs.

A signal flow graph is a diagram consisting of nodes that are connected by several directed branches and is a graphical representation of a set of linear relations. The representation of systems in the signal-flow graph method is somewhat similar to the block diagram representation except that the block is replaced with a branch as shown in figure 1.



Figure 1:

The input/output points (junctions) are called **nodes**. The relation between each pair of variables is written next to the directional arrow. Given below are a list of terms that are commonly used in this representation.

A **path** is a branch or a continuous sequence of branches that can be traversed from one signal (node), to another signal (node).

LECT. 5

A **loop** is a closed path that originates and terminates at the same node, and along the path no node is met twice.

Two loops are **nontouching** if they do not have a common node.

Example: signal-flow graph of a two-input, two-output system. Consider a linear system with two inputs and two outputs as shown below:

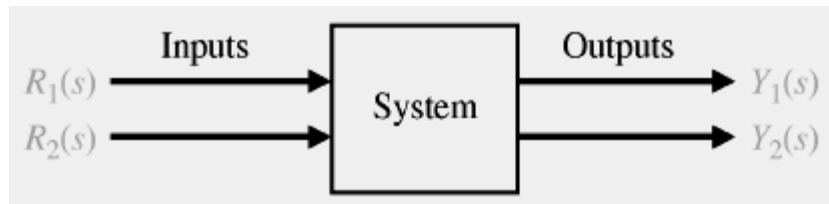


Figure 2: Two-input, two-output system

This system is described by transfer function relations as follows

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} R_1(s) \\ R_2(s) \end{bmatrix}, \quad (1)$$

where G_{ij} is the transfer function relating the i th output variable to the j th input variable. The corresponding signal-flow graph representation is shown on figure 3.

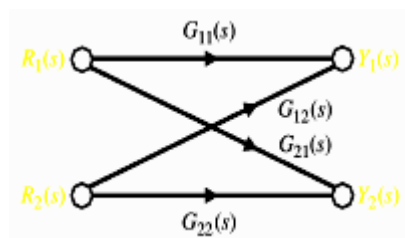


Figure 3: Signal-flow graph of a two-input, two-output system.

Example. Consider a system described by the following set of algebraic equations

$$a_{11}x_1 + a_{12}x_2 + r_1 = x_1, \quad (2)$$

$$a_{21}x_1 + a_{22}x_2 + r_2 = x_2, \quad (3)$$

where r_1 , r_2 are input variables, and x_1 , x_2 are output variables. The corresponding signal-flow graph is shown in figure 4.

LECT. 5

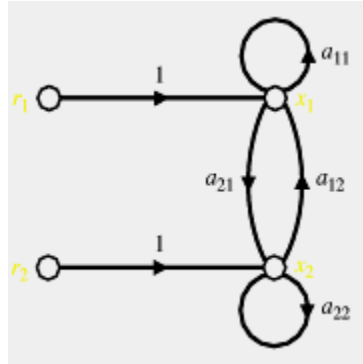


Figure 4: Signal-flow graph of the system (2), (3).

Equations (2), (3), can be rewritten as follows

$$(1 - a_{11})x_1 - a_{12}x_2 = r_1, \quad (4)$$

$$-a_{21}x_1 + (1 - a_{22})x_2 = r_2, \quad (5)$$

Using Cramer's rule, one can get the following solution of the equations (4), (5),

$$x_1 = \frac{(1 - a_{22})r_1 + a_{12}r_2}{(1 - a_{11})(1 - a_{22}) - a_{12}a_{21}} = \frac{(1 - a_{22})}{\Delta}r_1 + \frac{a_{12}}{\Delta}r_2, \quad (6)$$

and

$$x_2 = \frac{(1 - a_{11})r_2 + a_{21}r_1}{(1 - a_{11})(1 - a_{22}) - a_{12}a_{21}} = \frac{a_{21}}{\Delta}r_1 + \frac{(1 - a_{11})}{\Delta}r_2, \quad (7)$$

where Δ is the determinant of the set of equations (2), (3),

$$\Delta = (1 - a_{11})(1 - a_{22}) - a_{12}a_{21}.$$

Let us analyze the equations (6), (7). The determinant Δ can be rewritten as follows

$$\Delta = 1 - a_{11} - a_{22} - a_{12}a_{21} + a_{11}a_{22} = 1 - I_1 + I_2,$$

Where

$$I_1 = a_{11} + a_{22} + a_{12}a_{21},$$

$$I_2 = a_{11}a_{22}.$$

Note that the term I_1 is a sum of gains of all loops of the system. On the other hand, I_2 is a sum of gain products of all possible two loops that do not touch each other.

LECT. 5

Note also that numerators of transfer functions from an input r_i to an output x_j is equal to product of the gain of the corresponding path and the determinant of the part of the graph that does not touch the path. For example, numerator of transfer function from input r_1 to output x_1 is $1 - a_{22}$ which is equal to product of the gain 1 of the path from r_1 to x_1 and the determinant $1 - a_{22}$ of the part of the graph that does not touch the path $r_1 \rightarrow x_1$.

The generalization of the above considerations is called **Mason's rule**.

Mason's rule

Suppose we have a complex multiloop system, and we need to find the transfer function of this system from a given input $R(s)$ to a given output $Y(s)$. This transfer function can be found by using the following **Mason's formula**:

$$G(s) = \frac{Y(s)}{R(s)} = \frac{1}{\Delta(s)} \sum_{i=1}^n G_i(s) \Delta_i(s). \quad (8)$$

Here:

$\Delta(s)$ is the determinant of the system. $\Delta(s)$ can be calculated as follows

$$\begin{aligned} \Delta(s) = & 1 - \{\text{sum of all different loop gains}\} \\ & + \{\text{sum of the gains product of all combinations of two nontouching loops}\} \\ & - \{\text{sum of the gains product of all combinations of three nontouching loops}\} \\ & + \dots \end{aligned} \quad (9)$$

Further, n is a number of different forward paths from input $R(s)$ to output $Y(s)$.

G_i is gain of the i -th path from $R(s)$ to $Y(s)$.

$\Delta_i(s)$ is the determinant of the i -th forward path. Δ_i is a value of determinant Δ for that part of the signal flow graph that does not touch the i -th forward loop.

LECT. 5

Example: signal-flow graph. Consider a multiple loop feedback control system shown in figure 5.

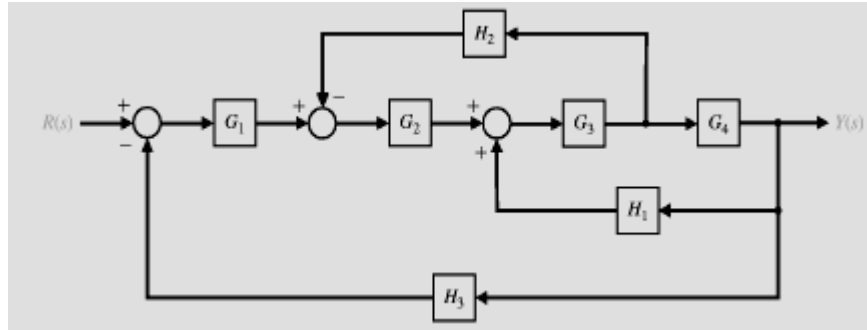


Figure 5: Multiple loop feedback control system

Problem. Find transfer function of the system $G(s) = \frac{Y(s)}{R(s)}$ using

Mason's rule

Solution. By drawing the signal-flow graph for the above system we have,

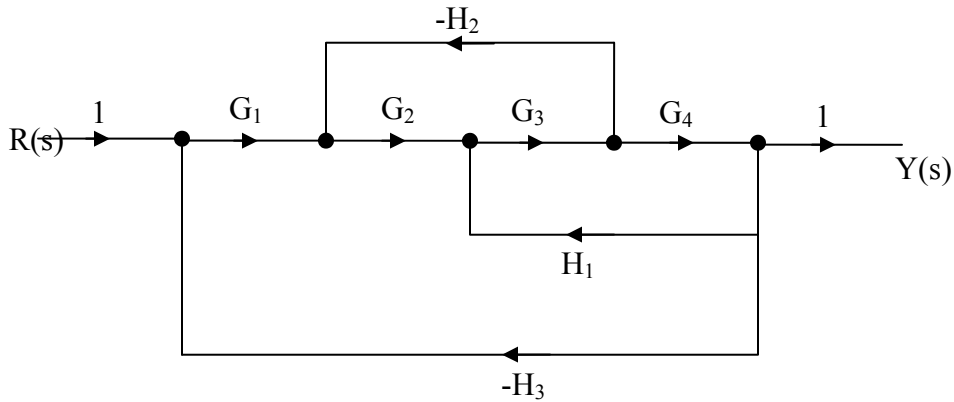


Figure 6: Signal-Flow graph for the above system

From the above figure there are three loops which are:

$$L_1 = - G_2 G_3 H_2$$

$$L_2 = G_3 G_4 H_1$$

$$L_3 = - G_1 G_2 G_3 G_4 H_3$$

And one forward path between the output and the input given by:

$$PG_1 = G_1 G_2 G_3 G_4$$

LECT. 5

Since there are no two (or more) nontouching loops, then:

$$\Delta(s) = 1 - (L_1 + L_2 + L_3)$$

And since the forward path touches all loops in the graph, then,

$$\Delta_1(s) = 1$$

Then

$$G(s) = \frac{Y(s)}{R(s)} = \frac{PG_1\Delta_1}{\Delta} = \frac{G_1G_2G_3G_4}{1 - G_3G_4H_1 + G_2G_3H_2 + G_1G_2G_3G_4H_3}$$

End of Lecture Five

Computer control design and modeling

Lecture 6

State Space Analysis

Introduction

A state space model is a description in terms of a set of first-order differential equations which are written compactly in a standard matrix form. This standard form has permitted the development of general computer programs, which can be used for the analysis and design of even very large systems.

State Space Models

The derivation of state space models is not different from that of transfer functions in that the differential equations describing the system dynamics are written first. In transfer function models these equations are transformed and variables are eliminated between them to find the relation between selected input and output variables. For state models, instead, the equations are arranged into set of first order differential equations in terms of selected state variables, and the output are expressed in these same state variables.

State variables should not normally derived from transfer functions, but directly from the original systems equations. But in this lecture examples will be given to relate state models to the transfer functions.

Consider a system described by the nth-order differential equation

$$\frac{d^n w}{dt^n} + a_n \frac{d^{n-1} w}{dt^{n-1}} + \dots + a_2 \frac{dw}{dt} + a_1 w = r \quad (1)$$

Or the equivalent transfer function. A state model for this system is not unique but depends on the choice of a state variables $x_1(t), x_2(t), \dots, x_n(t)$. One possible choice is the following:

$$x_1 = w \quad x_2 = \dot{w} \quad \dots \quad x_n = \overset{(n-1)}{w} \quad (2)$$

Directly from these definitions and by substitution (1), n first-order differential equations are obtained:

LECT. 6

$$\begin{aligned} \dot{x}_1 &= x_2 & \dot{x}_2 &= x_3 \dots \dots \dots \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= -a_1 x_1 - a_2 x_2 - \dots - a_n x_n + r \end{aligned}$$

The output w can be expressed in terms of these state variables:

$$w = x_1$$

It only remains to write in a standard vector-matrix form. The general form of a state-space model is as follows:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \quad (\text{state equation}) \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \quad (\text{output equation}) \end{aligned} \quad (3)$$

Here \mathbf{x} is the state vector, the vector of the state variables; \mathbf{u} is the control (input) vector, and \mathbf{y} the output vector. \mathbf{A} is the system matrix. In the example above the control (input) vector is the scalar function r and the output vector the scalar function w . It may be seen that

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} w \\ \dot{w} \\ \vdots \\ \vdots \\ \overset{(n-1)}{\dot{w}} \\ w \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & & & 0 \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ 0 & & & & & 1 \\ -a_1 & -a_2 & \dots & \dots & \dots & -a_n \end{bmatrix} \quad (4)$$

$$\mathbf{C} = [1 \quad 0 \quad \dots \quad 0] \quad \mathbf{D} = 0$$

Example 1: A Transfer function without Zeros.

$$\frac{W}{R} = \frac{5}{s^3 + 6s^2 + 9s + 3}$$

$$\ddot{w} + 6\dot{w} + 9w = 5r$$

Choose state variables $x_1(t)$, and $x_2(t)$, then

$$x_1 = w \quad x_2 = \dot{w} \quad x_3 = \ddot{w}$$

LECT. 6

Then a state model representing this transfer function or the corresponding differential equation is obtained as in general case. The definitions and the differential equation yield

$$\dot{x}_1 = x_2 \quad \dot{x}_2 = x_3 \quad \dot{x}_3 = -3x_1 - 9x_2 - 6x_3 + 5r$$

In matrix form and with the output w expressed also in terms of all state variables.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -9 & -6 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} r \quad w = [1 \ 0 \ 0] \mathbf{x}$$

Example 2: A Transfer function with Zeros.

$$\frac{W}{R} = \frac{5s^2 + 2s + 2}{s^3 + 6s^2 + 9s + 3}$$

Or

$$\ddot{w} + 6\dot{w} + 9w + 3w = 5\ddot{r} + 2\dot{r} + 2r$$

First, consider only the denominator:

$$\frac{V}{R} = \frac{1}{s^3 + 6s^2 + 9s + 3} \quad \ddot{v} + 6\dot{v} + 9v + 3v = r$$

As in Example 1

$$\mathbf{x} = \begin{bmatrix} v \\ \dot{v} \\ \ddot{v} \end{bmatrix} \quad \dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -9 & -6 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

But

$$W = (5s^2 + 2s + 2) V \quad \text{or} \quad w = 5\ddot{v} + 2\dot{v} + 2v = [2 \ 2 \ 5] \mathbf{x}$$

Hence the output equation, with y=w, is

$$y = \mathbf{C} \mathbf{x} \quad \mathbf{C} = [2 \ 2 \ 5]$$

LECT. 6

Example 3:

$$\frac{Y}{U} = \frac{K(s+a)}{s+b} \quad \text{or} \quad \dot{y} + by = Kau + K\dot{u}$$

By the approach of the last example, $V/U = 1/(s+b)$ yield a state equation $\dot{v} = -bv + u$. with state variable v . But then $Y = K(s+a)V$, so $Y = K\dot{v} + Kav$, and \dot{v} is now not a state variable. With equal powers in numerator and denominator, the method of Example 2 can be used if a division is performed first:

$$Y = K\left(1 + \frac{a-b}{s+b}\right)U = KU + X \quad \text{Where} \quad \frac{X}{U} = \frac{K(a-b)}{s+b}$$

This yield $\dot{x} = -bx + K(a-b)u, y = Ku + x$.

Example 4: RC Simple Lag Circuit.

In this circuit input voltage $e_i(t)$ and output voltage $e_o(t)$ are related by

$$RC\dot{e}_o + e_o = e_i \quad \frac{E_o}{E_i} = \frac{1}{RCs + 1}$$

The output is measurable and meaningful and a suitable state variable, identified as x . This is also the output y . Thus, with input $e_i = u$, both the differential equation and the transfer function lead to the state model

$$\dot{x} = \frac{-x}{RC} + \frac{u}{RC} \quad y = x \quad (u = e_i)$$

Example 5: Spring-Mass-Damper.

Mass position x and applied force f are related by

$$m\ddot{x} + c\dot{x} + kx = f \quad \frac{X(s)}{F(s)} = \frac{1}{ms^2 + cs + k}$$

Here also the transfer function and differential equation descriptions will yield the same state model ($x_1 = x \quad x_2 = \dot{x} \quad u = f \quad y = x_1$):

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-c}{m} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \quad y = [1 \quad 0] \mathbf{x}$$

End of Lecture six

Computer control design and modeling

Lecture 7

Time-domain response of a first-order and second-order control systems

In this lecture, we will study in details the time-domain response of a first-order and a second-order control systems. In particular, we will

- study how to use poles and zeros to determine time-response of a system,
- introduce performance specifications of transient response of a first-order system,
- learn how to determine transfer function of a first order system from time-domain response data.
- describe different types of natural responses of a second-order (stable) system,
- define performance specifications for a second-order system,
- learn how to use poles to determine the nature of response without exact calculation of the response.

Poles and zeros of a transfer function

Let us recall the definitions of poles and zeros of a transfer function. Consider a transfer function

$$F(s) = \frac{b_1 s^m + b_2 s^{m-1} + \dots + b_{m+1}}{s^n + a_1 s^{n-1} + \dots + a_n} = \frac{B(s)}{A(s)}.$$

Poles of a signal (system) are the roots of the denominator polynomial $A(s)$.

It is clear that poles of the system may also be defined as the values of s that cause the transfer function to become infinite. If the factor in denominator can be canceled by the same factor in the numerator, the transfer function may be not infinite at the root of this factor. In control systems, however, the root of the canceled factor in the denominator is usually also referred as a pole even though the transfer function is not infinite at this value.

Zeros of a signal (system) are the roots of the numerator polynomial $B(s)$.

LECT. 7

Similarly to the case of poles, the root of the canceled factor in the numerator is usually also referred as a zero even though the transfer function may be not zero at this value.

Example. Poles, zeros, and time response of a first order system

Consider a system described by the transfer function

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

Let us find the unit step response of the system. The Laplace transform of a unit step signal is $\frac{1}{s}$, therefore the Laplace transform of the unit step response of the system is

$$C(s) = G(s)\frac{1}{s} = \frac{s+2}{s(s+5)} = \frac{K_1}{s} + \frac{K_2}{s+5}$$

The residues are

$$K_1 = \left. \frac{s+2}{s+5} \right|_{s=0} = \frac{2}{5}$$

$$K_2 = \left. \frac{s+2}{s} \right|_{s=-5} = \frac{3}{5}$$

Thus, the time-domain response of the system is

$$c(t) = \frac{2}{5} + \frac{3}{5}e^{-5t}$$

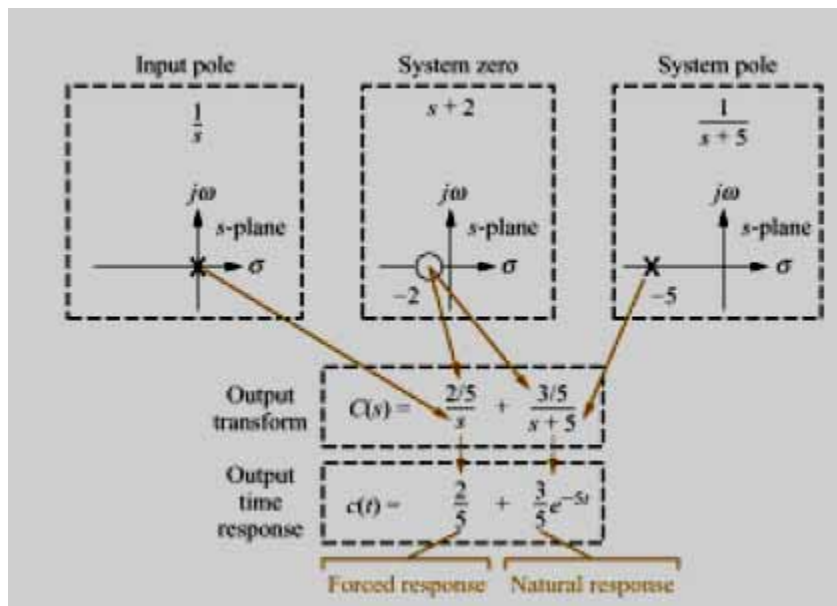


Figure 1: System response

LECT. 7

- The part of the response that corresponds to the poles of the input function is called **forced response**. One can see that a pole of the input function determine the *form* of the forced response.
- The part of the response that corresponds to the poles of the transfer function is called **natural response**. Again, a pole of the transfer function determines the form of the natural response.
- Since a pole of the transfer function is located at the real axis, the natural response of the system has an exponential form Ke^{-at} . The farther to the left a pole on the negative real axis, the faster the exponential response will decay to zero.

Transient response specifications of a first-order system

Time constant

Consider a first-order system

$$G(s) = \frac{K}{s+a}.$$

The Laplace transform of its step response is as follows

$$C(s) = \frac{K}{s(s+a)} = \frac{K/a}{s} + \frac{-K/a}{s+a}.$$

Taking the inverse Laplace transform, one can find the time-domain step response as follows

$$c(t) = \frac{K}{a} (1(t) - e^{-at}). \quad (1)$$

The response (1) is plotted in figure 2 (for simplicity, $K = a$ is assumed).

Definition 1. The value

$$T = \frac{1}{a}$$

is called the **time constant** of the response.

When $t = 1/a$, we have

$$c\left(\frac{1}{a}\right) = 1 - e^{-a \cdot \frac{1}{a}} \Big|_{t=\frac{1}{a}} \approx 1 - 0.37 = 0.63.$$

LECT. 7

We see that the time constant is the time that takes for the step response to rise to 63% of its final value.

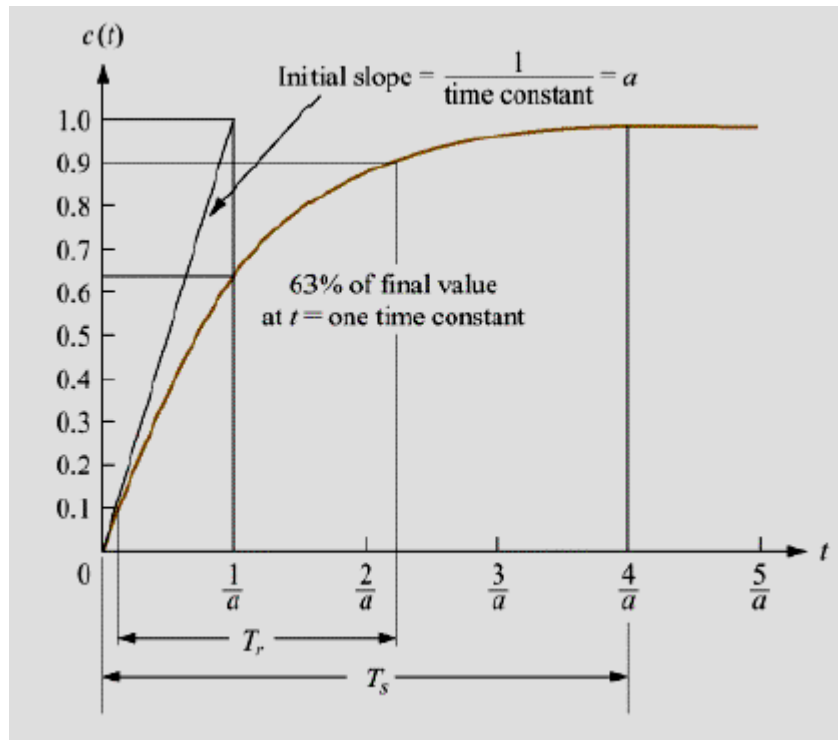


Figure 2: First order system response to a unit step

The time constant can obviously be evaluated from the pole as shown in figure 3. One can say that the pole is located at the *reciprocal* of the time constant, and the farther the pole from imaginary axis, the faster the transient response.

Rise time and settling time

The time constant can be considered as a transient response specification for a first-order system. The following two transient response specifications can also be used.

Rise time T_r is defined as the time for the response to go from 0.1 to 0.9 of its final value. To determine rise the time from the time constant, denote t_0 , t_1 as follows $c(t_1) = 0.9$, $c(t_0) = 0.1$. We have $1 - e^{-at_1} = 0.9$, therefore $at_1 = \ln 10 \approx 2.31$. On the other hand, $1 - e^{-at_0} = 0.1$,

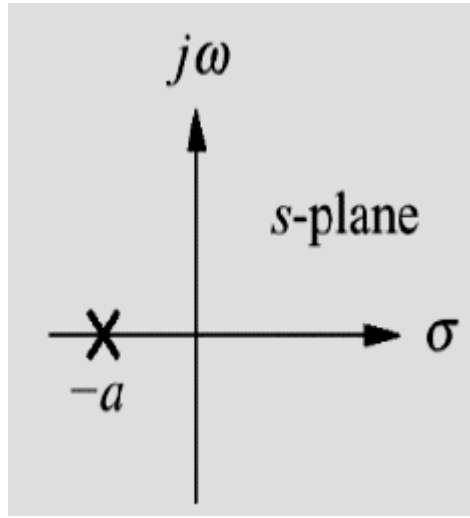


Figure 3: Pole plot

therefore $at_0 = \ln\left(\frac{10}{9}\right) \approx 0.11$. Therefore the rise time can be calculated as follows

$$T_r = t_1 - t_0 \approx \frac{2.31}{a} - \frac{0.11}{a} = \frac{2.2}{a} = 2.2T,$$

where $T = 1/a$ is the time constant.

Settling time T_s is the time for the response to reach, and stay within, 2% of its final value. To calculate settling time, put $1 - e^{-aT_s} = 0.98$, therefore $aT_s = \ln 50 \approx 3.91$, and we have

$$T_s \approx \frac{3.91}{a}.$$

First-order transfer functions via testing

Often it is not possible or practical to obtain a system's transfer function analytically. However, with a step input, we can measure the time constant and steady state value, from which the transfer function can be calculated.

Consider again a first-order system

$$G(s) = \frac{K}{s + a},$$

LECT. 7

and its step response is

$$C(s) = \frac{K}{s(s+a)} = \frac{K/a}{s} + \frac{-K/a}{s+a}.$$

One can identify K and a from laboratory testing as follows. Assume, the input step response is given in figure 4.

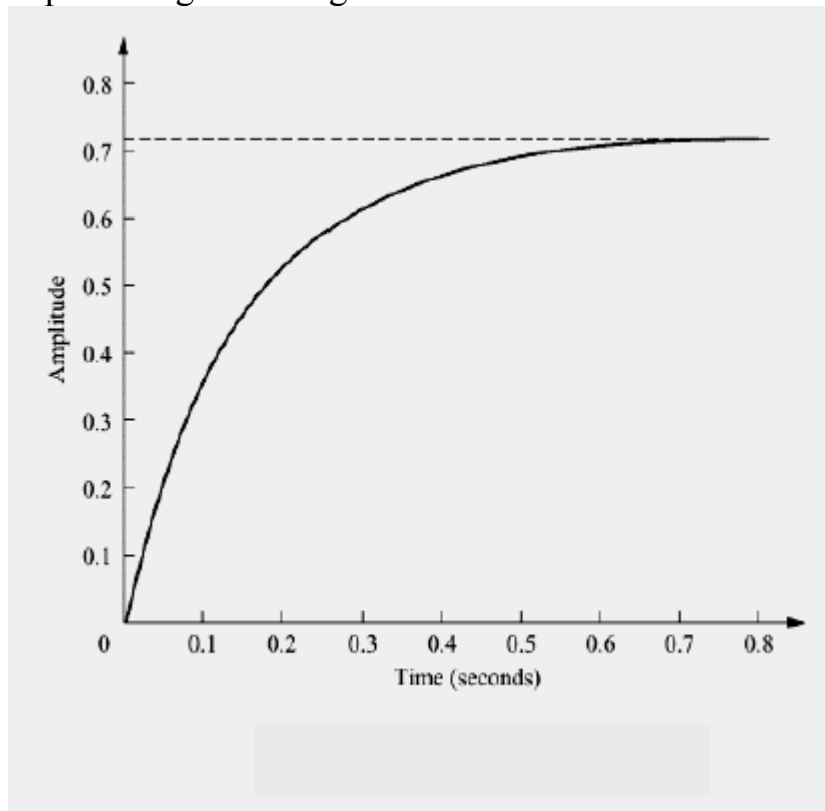


Figure 4: Step response of a system

From the response, we measure the time constant, that is, the time for the amplitude to reach 63% of its final value. Since the final value is about 0.72, the time constant is evaluated where the curve reaches $0.63 \times 0.72 \approx 0.45$, i.e. about 0.13. Hence

$$a = \frac{1}{0.13} = 7.7.$$

To find K it is clear that the forced response reaches a steady state value of

$$\frac{K}{a} \approx 0.72$$

LECT. 7

Since $a \approx 0.13$, we see that $K \approx 5.54$. Therefore

$$G(s) \approx \frac{5.54}{s + 7.7}.$$

Time-domain response of a second-order control system

Comparing to the simplicity of a first-order system, a second order system exhibits a wide range of responses. For example, a second order system can display characteristics much like a first-order system or display damped or pure oscillations for its transient response. Second order systems are very important in control systems engineering, since many control systems design methods are based on second-order system analysis.

A second-order system without zeros

Consider a second-order system without zeros described by a transfer function of the following general form

$$G(s) = \frac{k}{s^2 + as + b}. \quad (2)$$

Denote $\omega_n = \sqrt{b}$, $\xi = \frac{a}{2\omega_n}$. Equation (2) can therefore be rewritten as follows

$$G(s) = \frac{k}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (3)$$

It is clear that the term in the numerator is simply a scale or input multiplying factor that can take on any value without affecting the form of the derived results. For simplicity, we take $k = \omega_n^2$, and finally get a transfer function of the following form

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (4)$$

The parameter ω_n is called *natural frequency* of a second order system, and ζ is called *damping ratio*.

Consider a unit step response of a second order system (4). We have

$$C(s) = G(s) \frac{1}{s} = \frac{K_1}{s} + \frac{K_2 s + K_3}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (5)$$

LECT. 7

The first term in the right-hand side of equation (5) corresponds to forced response, while the second term determines the natural response of a second-order system. It is easy to see that $K_1 = 1$. Solving for the poles of the transfer function in equation (4) yields

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \quad (6)$$

The form of the natural unit step response of a second-order (stable) system is determined by the value of damping ratio ζ .

Case $\zeta = 0$

If $\zeta = 0$, we have two imaginary complex conjugate poles $s_{1,2} = \pm j\omega$. These poles generate a sinusoidal natural response whose frequency is equal to ω_n . This type of response is called *undamped*. It is shown in figure 5.

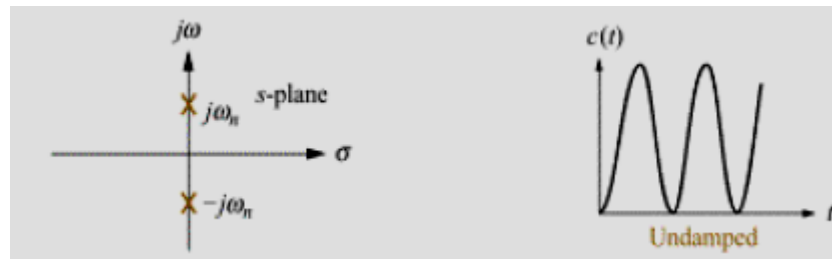


Figure 5: Undamped response

Case $0 < \zeta < 1$

In this case equation (6) results in a pair of complex conjugate poles with negative real part

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}.$$

These poles generate a natural response of the form of damped sinusoid with an exponential envelope whose time constant is equal to the reciprocal of the pole's real part. This type of response is called *underdamped response*. It is shown in figure 6.

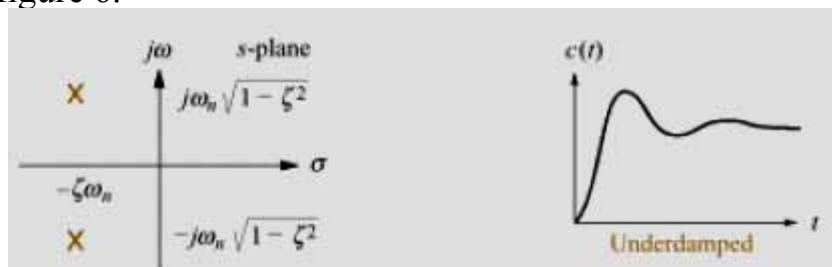


Figure 6: Underdamped response

LECT. 7

Case $\zeta > 1$

If $\zeta > 1$, then the formula (6) gives two negative real poles $s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$, $s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$. The corresponding natural response is equal to sum of two exponentials with time constants equal to reciprocal of the pole locations

$$c(t) = K_1 e^{s_1 t} + K_2 e^{s_2 t}.$$

This case is illustrated in figure 7. The response is called *overdamped*.

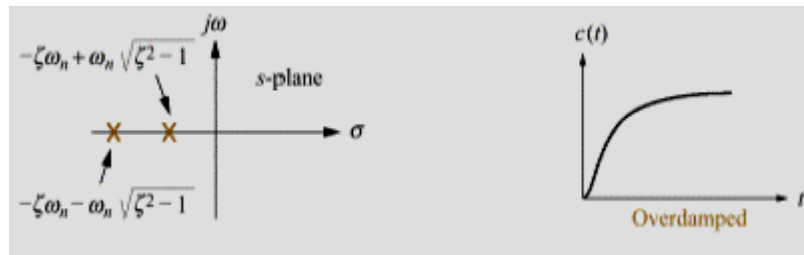


Figure 7: Overdamped response

Case $\zeta = 1$

In this case formula (6) gives two equal real poles

$$s_{1,2} = -\omega_n.$$

The corresponding natural time-domain response has a form

$$c(t) = K_1 e^{-\omega_n t} + K_2 t e^{-\omega_n t}.$$

This type of response is called *critically damped* response. This is the fastest possible response without overshoot.

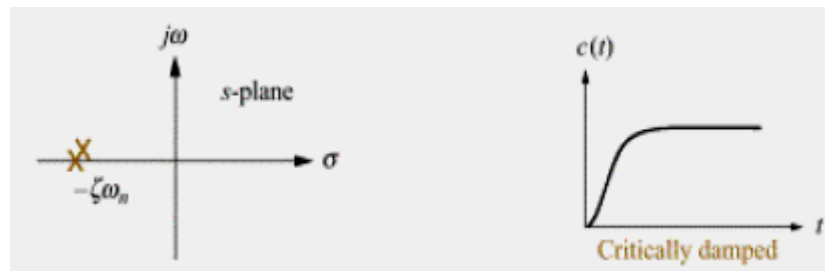


Figure 8: Critically damped response

Performance specifications of a second-order system

In this section we will introduce performance specifications for an underdamped second-order system. As in the case of first order systems, standard performance measures are usually defined in terms of the step response of a system.

The step response of the second order system (4) with $0 < \zeta < 1$ is given by the following formula

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \cos(\omega_n \sqrt{1-\zeta^2} t - \phi),$$

where $\phi = \tan^{-1}(\zeta/\sqrt{1-\zeta^2})$. The following performance specifications can be defined for the underdamped response of a second-order system.

Peak time T_p The time required to reach the first peak. Peak time can be calculated by the formula

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}. \quad (7)$$

Percent overshoot, %OS is the amount that the waveform overshoots the steady-state, or final, value at the peak time, expressed as a percentage of the steady-state value. Percent overshoot can be evaluated from ζ , ω_n using the following formula

$$\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100.$$

It is clear that the percent overshoot is a function only of the damping ratio ζ .

Settling time T_s is the time required for damped oscillations to reach and stay within $\pm 2\%$ of the steady-state (final) value. T_s can be evaluated by the formula

$$T_s = \frac{4}{\zeta\omega_n}. \quad (8)$$

Rise time T_r is the time required for the waveform to go from 0.1 to 0.9 of the final value. It is difficult to obtain exact analytic expression for T_r .

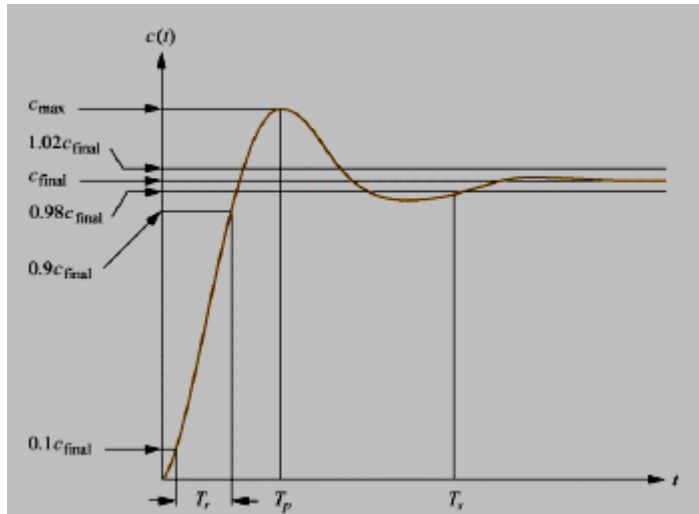


Figure 9: Second order underdamped response specifications

Performance characteristics vs. pole location

Let us consider the relation between performance characteristics and the location of the poles. Consider an example of the pole plot of a second order underdamped system, shown in figure 10. It is clear that, in this figure, $\cos \theta = \zeta$.

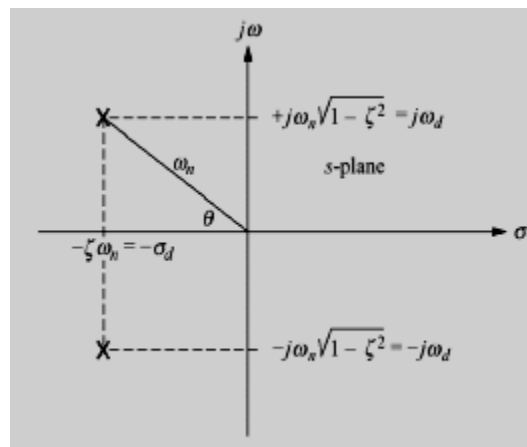


Figure 10: Pole plot for an underdamped second-order system

Comparing equations (7), (8) with the pole location, we see that

$$T_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} = \frac{\pi}{\omega_d},$$

where $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ is the imaginary part of the pole. On the other hand,

$$T_s = \frac{4}{\zeta \omega_n} = \frac{4}{\sigma_d}$$

LECT. 7

where σ_d is the magnitude of the real part of the pole.

We see that

- The peak time T_p is inversely proportional to the imaginary part of the pole.
- The settling time T_s is inversely proportional to the real part of the pole.
- Since $\zeta = \cos\theta$, radial lines are lines of constant ζ . Since percent overshoot is only a function of ζ , radial lines are lines of constant percent overshoot %OS.

In figure 11 the step responses are shown as the poles are moved in vertical direction, keeping the real part the same. We see that the frequency changes, but the envelope remains the same. Since all curves fit under the same exponential decay curve, the settling time is virtually the same for all waveforms.

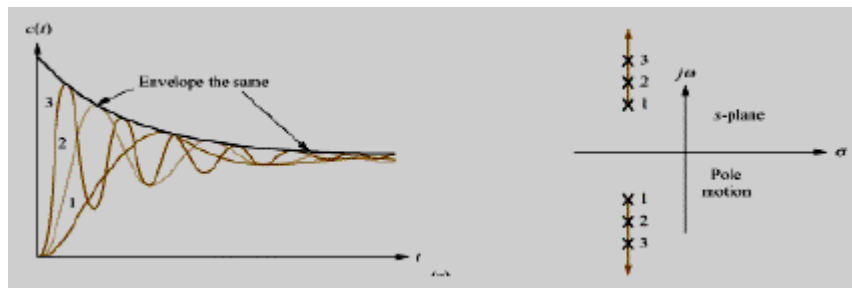


Figure 11: Poles are moved in vertical direction

In figure 12 the step responses are shown as the poles are moved in horizontal direction, keeping the imaginary part the same. As the poles move to the left, the response damps out more rapidly, while the frequency remains the same. It is clear that peak time is the same for all waveforms.

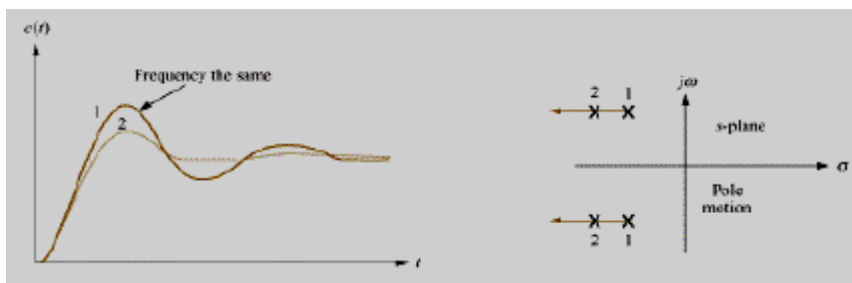


Figure 12: Poles are moved in horizontal direction

LECT. 7

In figure 13 the poles are moved along a constant radial line. We see that the percent overshoot remains the same. The farther the poles are from origin, the more rapid the response.

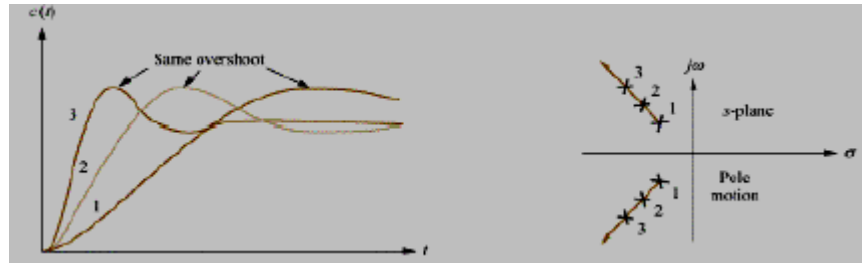


Figure 13: Poles are moved along a constant radial line

Example of finding ζ , ω_n , T_p , %OS, and T_s from pole location

Consider a pole plot shown in figure 11.

Problem. Find ζ , ω_n , T_p , %OS, and T_s .

Solution.

- The damping ratio is given by

$$\zeta = \cos \theta = \cos[\arctan(7/3)] \approx 0.394.$$

- The natural frequency

$$\omega_n = \sqrt{7^2 + 3^2} \approx 7.616.$$

- The peak time

$$T_p = \frac{\pi}{\omega_d} = \frac{\pi}{7} \approx 0.449.$$

- The percent overshoot

$$\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100 \approx 26.018\%.$$

LECT. 7

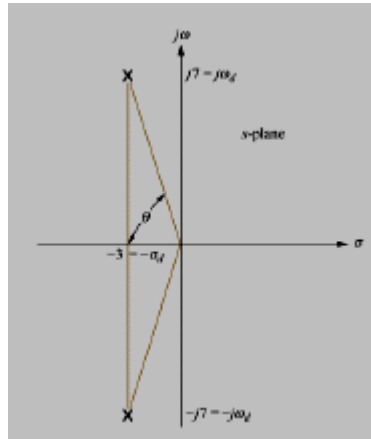


Figure 14: Pole plot

- The settling time

$$T_s = \frac{4}{\sigma_d} = \frac{4}{3} \approx 1.333.$$

End of Lecture seven

Computer control design and modeling

Lecture 8

Stability of linear systems. Routh-Hurwitz Criterion

As previously given, the time response of a system is a sum of the forced and natural responses

$$c(t) = c_{\text{forced}}(t) + c_{\text{natural}}(t).$$

The form of natural response depends only on the system, not the input. On the other hand, the form of forced response is dependent on the input. If the natural response grows without bounds, then eventually the natural response will be much greater than the forced response, and the system is no longer controlled. Therefore, for a control system to be useful, the natural response must eventually approach zero, or, at worst, oscillate.

Definition. A linear system is said to be:

- **stable** if the natural response approaches zero as time approaches infinity;
- **unstable** if the natural response grows without bound as time approaches infinity;
- **marginally stable** if the natural response neither decays nor grows without bound, but remains constant or oscillates as time approaches infinity.

Therefore, control system must be designed to be stable.

Stability vs. poles location

- Poles in the left half-plane yield either pure exponentially decreasing or damped sinusoidal natural responses. Therefore, if all the poles of the system are in the left half-plane (have negative real parts), then the system is stable.
- Poles in the right half-plane yield either pure exponentially increasing or exponentially increasing sinusoidal natural responses. Therefore, if a system has at least one pole in the right half-plane (has positive real parts), then the system is unstable.
- Poles of multiplicity greater than one on the imaginary axis lead to the sum of responses of the form

$$K t^n \cos(\omega t + \phi),$$

LECT. 8

which grows without bound as $t \rightarrow \infty$. Therefore, if a system has poles of multiplicity greater than one on the imaginary axis, then the system is unstable.

- Poles of multiplicity one on the imaginary axis yield pure sinusoidal natural response. Thus, if all poles of the system are only in the left half plane or on the imaginary axis, and all the poles on the imaginary axis are of multiplicity one, then the system is marginally stable.

A necessary condition for stability

Suppose a transfer function has only left half-plane poles, i.e. the system is stable. Then the factors of denominator of the transfer function consists of products of terms such as $(s+a_i)$, where a_i either real and positive, or complex with positive real parts. The products of such terms is a polynomial with all positive coefficients. Therefore:

- if the system is stable, then all the coefficients of the denominator must be positive.

It means that if any of the coefficients of the denominator polynomial is negative or missing, then the system is not stable.

Unfortunately, if all the coefficients of the denominator are positive and not missing, we do not have definite information about the system's pole location.

Routh-Hurwitz Criterion

- Routh-Hurwitz Criterion provides a method that yields stability information without the need to solve for poles of a system.
- Using Routh-Hurwitz Criterion one can find how many poles are in the left half-plane, right half-plane, and on the imaginary axis. However, using this method, one cannot find the exact coordinates of the poles.
- The method requires two steps:
 - generate a data table called Routh table;
 - interpret the Routh table to tell how many system poles are in each section (left half-plane, right half-plane, and imaginary axis) of the complex plane.

Suppose, for example, we need to determine stability of the system

$$G(s) = \frac{K(s)}{a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0}$$

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2			
s^1			
s^0			

Figure 1: Initial Layout for Routh table

Generating a basic Routh table

- Create the initial Routh table shown in figure 1.
 - Label the rows with powers of s from the highest power of the denominator to s^0 .
 - In the first row, write horizontally the coefficients from the highest power to the lowest one, skipping every other coefficient.
 - In the second row, write horizontally all the coefficients that skipped in the first row, from the highest power to the lowest one.
- Fill in the remaining entries as follows:
 - Each entry is a negative determinant of entries in the previous two rows divided by the entry in the first column directly above the calculated row.
 - The left column of the determinant is always the first column of the previous two rows.
 - The right column of the determinant is the column of the previous two rows that is above and to the right of the entry.

LECT. 8

- Any row may be multiplied by a positive constant (not by a negative one!!!). This operation will not change the values of the rows below (this follows from the properties of determinant).
- The table is completed when all the rows are completed down to s^0 . A completed Routh table is shown in figure 2.

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$-\frac{\begin{vmatrix} a_4 & a_2 \\ a_3 & a_1 \end{vmatrix}}{a_3} = b_1$	$-\frac{\begin{vmatrix} a_4 & a_0 \\ a_3 & 0 \end{vmatrix}}{a_3} = b_2$	$-\frac{\begin{vmatrix} a_4 & 0 \\ a_3 & 0 \end{vmatrix}}{a_3} = 0$
s^1	$-\frac{\begin{vmatrix} a_3 & a_1 \\ b_1 & b_2 \end{vmatrix}}{b_1} = c_1$	$-\frac{\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$	$-\frac{\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$
s^0	$-\frac{\begin{vmatrix} b_1 & b_2 \\ c_1 & 0 \end{vmatrix}}{c_1} = d_1$	$-\frac{\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$	$-\frac{\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$

Figure 2: Completed Routh table

Example 1: Creating a Routh table. Consider a feedback system in figure 3.

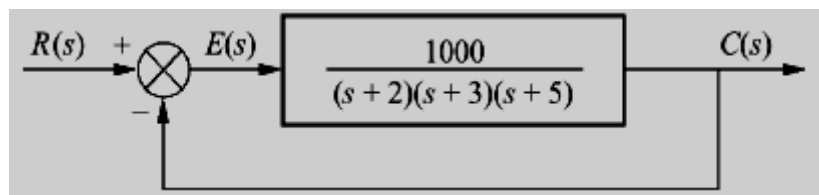


Figure 3: Feedback system

The transfer function of the closed loop system is as follows

$$\frac{C(s)}{R(s)} = \frac{1000}{s^3 + 10s^2 + 31s + 1030} \quad (1)$$

The problem is to generate the corresponding Routh table. The answer is shown in figure 4.

Interpreting the basic Routh table

If there is no zeros in the first column of the Routh table, then the Routh table can be interpreted as follows:

- The number of roots of the polynomial that are in the right half-plane is equal to the number of sign changes in the first column.

For example, the table shown in figure 4 has two sign changes in the first column. Thus, the system (1) has two poles in the right half-plane, therefore it is unstable.

s^3	1	31	0
s^2	10 1	1030 103	0
s^1	$-\frac{\begin{vmatrix} 1 & 31 \\ 1 & 103 \end{vmatrix}}{1} = -72$	$-\frac{\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}}{1} = 0$	$-\frac{\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}}{1} = 0$
s^0	$-\frac{\begin{vmatrix} 1 & 103 \\ -72 & 0 \end{vmatrix}}{-72} = 103$	$-\frac{\begin{vmatrix} 1 & 0 \\ -72 & 0 \end{vmatrix}}{-72} = 0$	$-\frac{\begin{vmatrix} 1 & 0 \\ -72 & 0 \end{vmatrix}}{-72} = 0$

Figure 4: Completed Routh table for example 1

Routh-Hurwitz Criterion: Special Cases

Zero in the first column of a row

If the first element of the row is zero, division by zero would be required to form the next row. To avoid this, a small number ϵ is assigned to replace the zero in the first column. The value of ϵ is then allowed to approach zero from either positive or negative side, after which the signs of the entries in the first column can be determined.

LECT. 8

Example 2. Consider a system

$$G(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3}$$

It's Routh table is shown on figure 5.

There are two sign changes, therefore two poles are in the right half-plane.

s^5	1	3	5
s^4	2	6	3
s^3	8ϵ	$\frac{7}{2}$	0
s^2	$\frac{6\epsilon - 7}{\epsilon}$	3	0
s^1	$\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	0	0
s^0	3	0	0

Figure 5: Completed Routh table for example 2

Entire row is zero

Sometimes, we can find that an entire row consists of zeros. This case must be handled differently from the case of a zero only in the first column of the row.

Example 3. Consider a system

$$G(s) = \frac{10}{s^5 + 7s^4 + 6s^3 + 42s^2 + 8s + 56}$$

It's Routh table is shown on figure 6. In particular, one can see that the third row consists of zeros. In this case we should:

- Return to the row immediately above the row of zeros.
- Form an auxiliary polynomial, using the entries of that row as coefficients. The polynomial will start from the power of s in the label column, and continue by skipping every other one and diminishing in power. Thus, the polynomial is as follows

$$P(s) = s^4 + 6s^2 + 8.$$

LECT. 8

- Differentiate the above polynomial with respect to s to obtain

$$\frac{dP(s)}{ds} = 4s^3 + 12s + 0.$$

- Use coefficients of the last polynomial to replace the row of zeros.

s^5	1	6	8
s^4	1 1	6 6	8 8
s^3	1 1 1	12 3	0 0 0
s^2	3	8	0
s^1	$\frac{1}{3}$	0	0
s^0	8	0	0

Figure 6: Routh table for example 3

The remainder of the table is formed in the standard way. We see that all entries in the first column are positive. Hence, there are no right half-plane poles, and the system is stable.

Example 4. The characteristics equation of a given system is

$$s^4 + 6s^3 + 11s^2 + 6s + K = 0$$

What the range of K values in order to insure that the system is stable?

s^4	1	11	K
s^3	6	6	0
s^2	10	K	0
s^1	$\frac{60 - 6K}{10}$	0	
s^0	K		

For the system to be stable, the following restrictions must be placed upon the parameter K: $60 - 6K > 0$ or $K < 10$, and $K > 0$. Thus K must be greater than zero and less than 10.

End of Lecture eight

Computer control design and modeling

Lecture 9

Steady-State Errors

As given before, in feedback control systems (as shown in figure 1), the input signal usually represents a desired output response, and the role of control system is to force the actual output to follow the input (desired output). The accuracy of this process is one of the main concerns of control system engineers. For example, if a control system is designed to stop an elevator at a desired floor, then the elevator must eventually be level enough with the floor for the passengers to exit. In particular, one of the main characteristics of a control system is the difference between the desired output and the actual output as time tends to infinity.

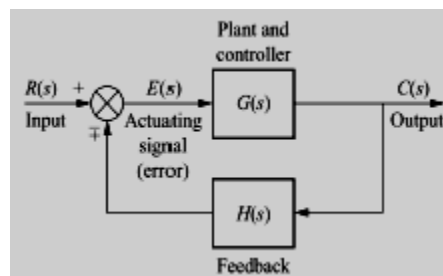


Figure 1: Feedback control system

Definition. Steady-state error is the difference between the input and the output for a prescribed test input as $t \rightarrow \infty$.

Test inputs

The following signals are usually used as test inputs as shown in figure 2:

- Step input $u(t) = 1(t)$. Step input represents constant position and it is useful in determining the ability of the control system to position itself with respect to stationary target.
- Ramp input $u(t) = t$. Ramp input represents constant-velocity input and it is useful to determine the ability of the system to track a constant-velocity target.

LECT. 9

- Parabola $u(t) = \frac{1}{2}t^2$. Parabolic input is a constant-acceleration input, so one can use this input to determine the ability of the system to track an accelerating target.

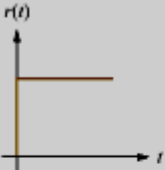
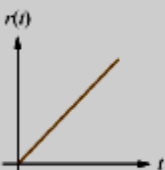
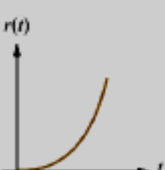
Waveform	Name	Physical interpretation	Time function	Laplace transform
	Step	Constant position	1	$\frac{1}{s}$
	Ramp	Constant velocity	t	$\frac{1}{s^2}$
	Parabola	Constant acceleration	$\frac{1}{2}t^2$	$\frac{1}{s^3}$

Figure 2: Test inputs for evaluating steady-state errors

Steady-state Error in Unity Feedback Systems

Consider the following unity feedback system as shown in figure 3.

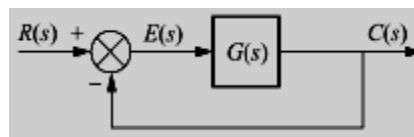


Figure 3: Unity feedback system

In this figure, the signal $E(s)$ is the error between the input $R(s)$, and the output $C(s)$. The goal of this section is to express the steady-state error in terms of transfer function $G(s)$ in the forward path (open loop transfer function).

From figure 3, we have

$$E(s) = R(s) - C(s).$$

LECT. 9

But

$$C(s) = E(s)G(s),$$

Therefore it is easy to get

$$E(s) = \frac{R(s)}{1 + G(s)}.$$

Assume the closed-loop system is stable. Then, applying final value theorem, one can determine the value $e(\infty) = \lim_{t \rightarrow \infty} e(t)$ as follows

$$e(\infty) = \lim_{s \rightarrow 0} \frac{sR(s)}{1 + G(s)}. \quad (1)$$

Equation (1) allows us to calculate the steady-state error $e(\infty)$ for given input $R(s)$ and transfer function $G(s)$.

Steady-state error for the step input

For step input, $R(s) = \frac{1}{s}$. Using equation (1), we get

$$e_{\text{step}}(\infty) = \lim_{s \rightarrow 0} \frac{s(1/s)}{1 + G(s)} = \frac{1}{1 + \lim_{s \rightarrow 0} G(s)}. \quad (2)$$

The value

$$K_p = \lim_{s \rightarrow 0} G(s)$$

is called static position error constant. In order to have zero steady-state error, $G(s)$ must satisfy

$$K_p = \lim_{s \rightarrow 0} G(s) = \infty.$$

To satisfy the previous equation, $G(s)$ must take on the following form

$$G(s) = \frac{(s + z_1)(s + z_2) \dots}{s^n (s + p_1)(s + p_2) \dots}, \quad (3)$$

where $n \geq 1$, i.e. at least one pole of $G(s)$ must be at the origin (when $n = 1$ then the system is called type 1 system), or equivalently, at least one pure integration must be present in the forward path.

LECT. 9

If there is no integration ($n = 0$ or type 0 system), then we have

$$K_p = \lim_{s \rightarrow 0} G(s) = \frac{z_1 z_2 \dots}{p_1 p_2 \dots},$$

i.e. the static position error constant K_p is finite, and therefore (as given in formula (2)), the corresponding steady-state error is finite.

Steady-state error for the ramp input

For ramp input, we have $R(s) = \frac{1}{s^2}$. Using formula (1), we get

$$e_{\text{ramp}}(\infty) = \lim_{s \rightarrow 0} \frac{s(1/s^2)}{1 + G(s)} = \lim_{s \rightarrow 0} \frac{1}{s + sG(s)} = \frac{1}{\lim_{s \rightarrow 0} sG(s)}. \quad (4)$$

The value

$$K_v = \lim_{s \rightarrow 0} sG(s)$$

is called static velocity error constant. To obtain zero steady-state error for a ramp input, one must have

$$K_v = \lim_{s \rightarrow 0} sG(s) = \infty.$$

To satisfy the last equation, $G(s)$ must be of the form (3) with $n \geq 2$, i.e. there must be at least two integrations in the forward path (when $n = 2$ then the system is called type 2 system).

If only one integrator exists in the forward path (type 1 system), then

$$K_v = \lim_{s \rightarrow 0} sG(s) = \frac{z_1 z_2 \dots}{p_1 p_2 \dots}$$

is finite, i.e. we have constant steady-state error.

If there is no integration in the forward path (type 0 system), then

$$K_v = \lim_{s \rightarrow 0} sG(s) = 0,$$

and the steady-state error is infinite and lead to diverging ramps.

LECT. 9

Steady-state error for the parabolic input

For the parabolic input $u(t) = \frac{1}{2} t^2$, it's Laplace transform is $R(s) = \frac{1}{s^3}$. Using formula (1), we get

$$e_{\text{parab}}(\infty) = \lim_{s \rightarrow 0} \frac{s(1/s^3)}{1 + G(s)} = \lim_{s \rightarrow 0} \frac{1}{s^2 + s^2 G(s)} = \frac{1}{\lim_{s \rightarrow 0} s^2 G(s)}. \quad (5)$$

The value

$$K_a = \lim_{s \rightarrow 0} s^2 G(s)$$

is called static acceleration error constant. In order to have zero steady-state error, we must have

$$K_a = \lim_{s \rightarrow 0} s^2 G(s) = \infty.$$

To satisfy the last equation, $G(s)$ must take on the form (3), where $n \geq 3$ (when $n = 3$ then the system is called type 3 system). In other words, to have zero steady-state error for a parabolic input, there must be at least three integrators in the forward path.

If there are only two integrators in the forward path (type 2 system), then

$$K_a = \lim_{s \rightarrow 0} s^2 G(s) = \frac{z_1 z_2 \dots}{p_1 p_2 \dots}$$

is finite, and therefore, the steady-state error for a parabolic input is finite. If the number of pure integrators in the forward path is less than two, then

$$K_a = \lim_{s \rightarrow 0} s^2 G(s) = 0,$$

which implies the steady-state error for a parabolic input is infinite.

The following table gives a summary of the different cases given above:

	Step Input $u(t)=1$	Ramp Input $u(t)=t$	Acceleration $u(t)=\frac{1}{2} t^2$
Type 0 system	$\frac{1}{1 + K_p}$	∞	∞
Type 1 system	0	$\frac{1}{K_v}$	∞
Type 2 system	0	0	$\frac{1}{K_a}$

Static error constants as steady-state error performance specifications

Static error constants can be used to specify the steady-state error characteristics of control systems. Just as damping ratio, settling time, peak time, and percent overshoot are used as specifications for a system's transient response, so the static position error constant K_p , static velocity error constant K_v , and static acceleration error constant, K_a , can be used as specifications for a control system's steady state error.

Example: Steady-state error via error constants

Problem 1. For each system on figure 4, evaluate the static error constants and find the expected error for the standard step, ramp, and parabolic inputs.

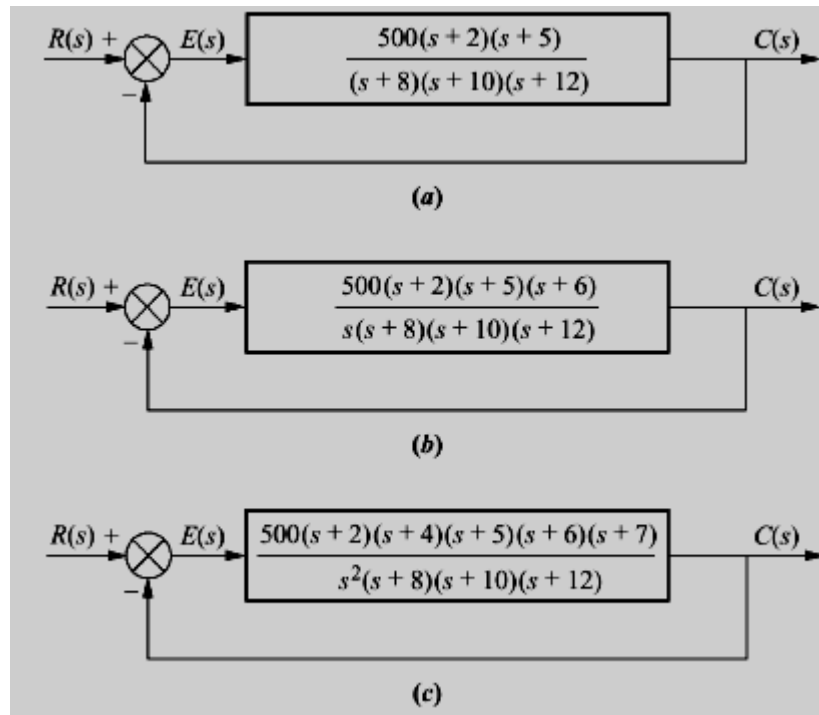


Figure 4: Feedback control systems for Problem 1

Solution. First, we need to verify that all the systems are stable. Second, for the system in figure 4, (a), we see that

$$K_p = \lim_{s \rightarrow 0} G(s) = \frac{500 \times 2 \times 5}{8 \times 10 \times 12} \approx 5.2,$$

$$K_v = \lim_{s \rightarrow 0} sG(s) = 0,$$

LECT. 9

and

$$K_a = \lim_{s \rightarrow 0} s^2 G(s) = 0.$$

Thus, for a step input, we have

$$e_{\text{step}}(\infty) = \frac{1}{1 + K_p} = 0.161.$$

For a ramp input,

$$e_{\text{ramp}}(\infty) = \frac{1}{K_v} = \infty,$$

and for a parabolic input

$$e_{\text{parab}}(\infty) = \frac{1}{K_a} = \infty.$$

For the system in figure 4, (b), we have

$$\begin{aligned} K_p &= \lim_{s \rightarrow 0} G(s) = \infty, \\ K_v &= \lim_{s \rightarrow 0} sG(s) = \frac{500 \times 2 \times 5 \times 6}{8 \times 10 \times 12} \approx 31.25, \\ K_a &= \lim_{s \rightarrow 0} s^2 G(s) = 0, \end{aligned}$$

Therefore

$$\begin{aligned} e_{\text{step}}(\infty) &= \frac{1}{1 + K_p} = 0, \\ e_{\text{ramp}}(\infty) &= \frac{1}{K_v} = \frac{1}{31.25} \approx 0.032, \\ e_{\text{parab}}(\infty) &= \frac{1}{K_a} = \infty. \end{aligned}$$

Finally, for the system in figure 4, (c),

$$\begin{aligned} K_p &= \lim_{s \rightarrow 0} G(s) = \infty, \\ K_v &= \lim_{s \rightarrow 0} sG(s) = \infty, \\ K_a &= \lim_{s \rightarrow 0} s^2 G(s) = \frac{500 \times 2 \times 4 \times 5 \times 6 \times 7}{8 \times 10 \times 12} = 875, \end{aligned}$$

Therefore

$$\begin{aligned} e_{\text{step}}(\infty) &= \frac{1}{1 + K_p} = 0, \\ e_{\text{ramp}}(\infty) &= \frac{1}{K_v} = 0, \\ e_{\text{parab}}(\infty) &= \frac{1}{K_a} = \frac{1}{875} \approx 0.00114. \end{aligned}$$

LECT. 9

Example: Gain design to meet steady-state error specifications

Problem 2. Given a control system in figure 5, find the value of K so that there is 10% error in the steady-state.

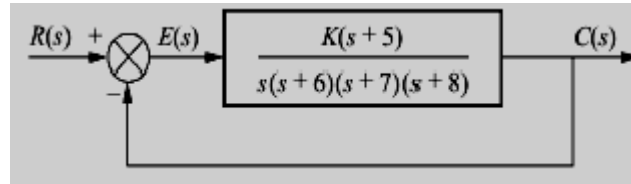


Figure 5: Feedback control systems for Problem 2

Solution. Since the system has one integrator, the error stated in the problem must apply to a ramp input. Thus,

$$e_{\text{ramp}}(\infty) = \frac{1}{K_v} = 0.1.$$

Therefore,

$$K_v = \lim_{s \rightarrow 0} sG(s) = \frac{K \times 5}{6 \times 7 \times 8} = \frac{1}{e_{\text{ramp}}} = 10,$$

which implies

$$K = \frac{10 \times 6 \times 7 \times 8}{5} = 672.$$

It remains to check, using Routh-Hurwitz criterion, that the closed-loop system is stable with this gain.

End of Lecture nine

Computer control design and modeling

Lecture 10

Steady-state errors for nonunity feedback systems

Control systems often do not have unity feedback because of compensation used to improve performance or because of the physical model for the system. In order to derive a method for handling steady state errors for nonunity feedback systems, take a nonunity feedback control system and for a unity feedback system by adding and subtracting unity feedback paths, as shown in figure 1.

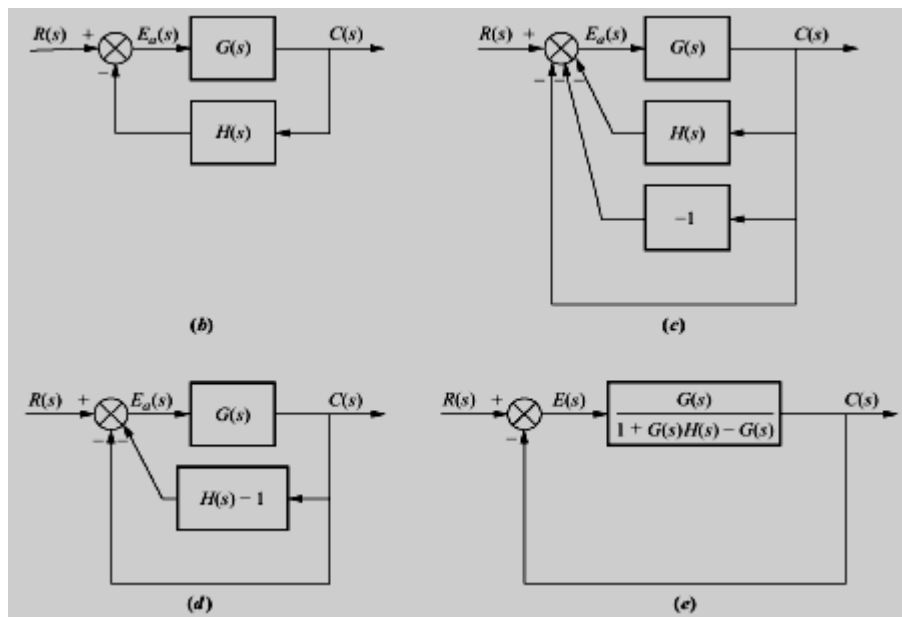


Figure 1: Forming an equivalent unity feedback system

It is clear that when nonunity feedback is present, the plant actuating signal $E_a(s)$ is not the actual error or difference between the input and the output.

Example 1. For the system shown in figure 2, find the steady-state error for a unit step input.

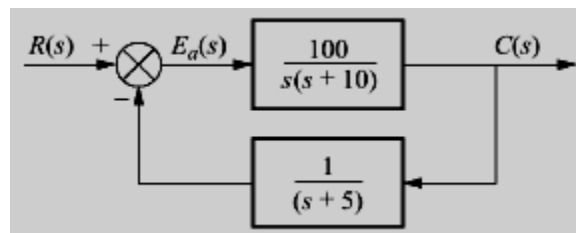


Figure 2: Feedback system for Example 1

LECT. 10

Solution. The first step is to make sure that the system is stable. Second, the system should be converted into an equivalent unity feedback system. Using the formula

$$G_e(s) = \frac{G(s)}{1 + G(s)H(s) - G(s)},$$

where $G_e(s)$ is the transfer function in the forward path of the equivalent unity feedback system, one can find

$$G_e(s) = \frac{100(s + 5)}{s^3 + 15s^2 - 50s - 400}.$$

The position error constant

$$K_p = \lim_{s \rightarrow 0} G_e(s) = -\frac{5}{4}.$$

Finally, the steady-state error is

$$e_{\text{step}}(\infty) = \frac{1}{1 + K_p} = -4.$$

Steady-state error for disturbances

A disturbance signal is an unwanted input signal that affects the system's behavior. Many control systems are subject to disturbances that cause the system to provide an inaccurate output. For example,

- Electronic amplifiers have inherent noise generated within the integrated circuits or transistors.

It is the job of control systems engineer to properly design the control system to partially eliminate the affects of disturbances. One of the advantages of using feedback is that the effect of unwanted disturbances can be effectively reduced.

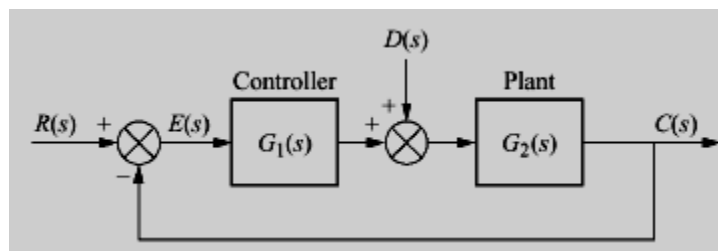


Figure 3: Feedback control system with disturbances

LECT. 10

Consider a system with disturbances shown in figure 3. In this figure, a disturbance $D(s)$ is injected between the controller and the plant. For the system with disturbances, the error $E(s)$ is given by the following formula

$$E(s) = \frac{1}{1 + G_1(s)G_2(s)}R(s) - \frac{G_2(s)}{1 + G_1(s)G_2(s)}D(s) = E_R(s) + E_D(s).$$

Applying final value theorem, we get

$$e(\infty) = e_R(\infty) + e_D(\infty), \quad (1)$$

where

$$e_R(\infty) = \lim_{s \rightarrow 0} \frac{s}{1 + G_1(s)G_2(s)}R(s), \quad (2)$$

and

$$e_D(\infty) = -\lim_{s \rightarrow 0} \frac{sG_2(s)}{1 + G_1(s)G_2(s)}D(s). \quad (3)$$

Here, $e_R(\infty)$ is the steady-state error due to $R(s)$, and $e_D(\infty)$ is the steady state error due to $D(s)$. How to reduce the error due to disturbances? If it is assume for example that $D(s)$ is a step disturbances, $D(s) = \frac{1}{s}$. Substituting this value into the last equation, we get

$$e_D(\infty) = -\frac{1}{\lim_{s \rightarrow 0} \frac{1}{G_2(s)} + \lim_{s \rightarrow 0} G_1(s)}. \quad (4)$$

The value $\lim_{s \rightarrow 0} G_1(s)$ is sometimes called dc gain of the system $G_1(s)$. The last formula shows that the steady-state error due to step disturbances can be reduced by increasing the dc gain of the controller $G_1(s)$.

Example 2. Steady-state error due to step disturbances. Consider a system in figure 4.

Problem. Find the steady-state error due to step disturbance $D(s)$.

Solution. The system is stable. Using formula (4), we get

$$e_D(\infty) = -\frac{1}{\lim_{s \rightarrow 0} \frac{1}{G_2(s)} + \lim_{s \rightarrow 0} G_1(s)} = \frac{1}{0 + 1000} = 0.001.$$

We see that, dc gain of $G_2(s)$ is infinite in this example, so the steady state error due to the step disturbance is inversely proportional to the dc gain of

LECT. 10

the controller $G_1(s)$. Thus, the effect of the step disturbance can be reduced by increasing dc gain of the controller.

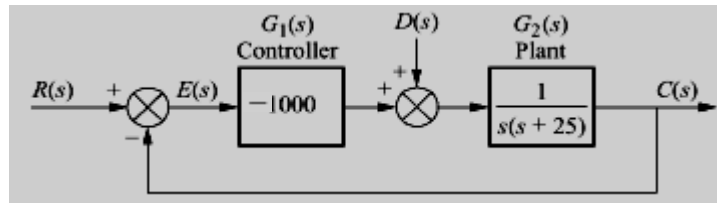


Figure 4: Feedback control system for example 2

Dynamic error constants

Dynamic error constants can be used to relate error function with time. These constants give the error at any time and can be used to calculate steady state error.

As given before,

$$E(s) = \frac{1}{1 + G(s)} R(s) \quad (5)$$

and by dividing numerator by denominator we have,

$$E(s) = \left(\frac{1}{k_1} + \frac{1}{k_2} s + \frac{1}{k_3} s^2 + \dots \right) R(s) \quad (6)$$

Where k_1 is the dynamic position error constant,

k_2 is the dynamic velocity error constant,

k_3 is the dynamic acceleration error constant.

Then

$$\therefore E(s) = \frac{1}{k_1} R(s) + \frac{1}{k_2} sR(s) + \frac{1}{k_3} s^2 R(s) + \dots \quad (7)$$

and by taking the inverse Laplace transform, we have

$$e(t) = \frac{1}{k_1} r(t) + \frac{1}{k_2} \dot{r}(t) + \frac{1}{k_3} \ddot{r}(t) + \dots \quad (8)$$

The last equation gives the error as a function of time. To calculate steady state error, we must use the following equation:

LECT. 10

$$\text{steady state error} = \lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} \left(\frac{1}{k_1} r(t) + \frac{1}{k_2} \dot{r}(t) + \frac{1}{k_3} \ddot{r}(t) + \dots \right) \quad (9)$$

Example 3.

Calculate the dynamic error constants for the system with the following open loop transfer function (for unity feedback):

$$G(s) = \frac{10}{s(s+1)}$$

And then find the steady state error for the following input:

$$r(t) = a_0 + a_1 t + a_2 t^2$$

Solution.

Since

$$\frac{E(s)}{R(s)} = \frac{1}{1+G(s)} = \frac{s+s^2}{10+s+s^2} = 0.1s + 0.09s^2 - 0.019s^3 + \dots$$

Then, the dynamic error constants are:

$$k_1 = \infty$$

$$k_2 = \frac{1}{0.1} = 10$$

$$k_3 = \frac{1}{0.09} = 11.1$$

$$k_4 = \frac{1}{-0.019} = -52.63$$

Then

$$\therefore E(s) = 0.1sR(s) + 0.09s^2R(s) - 0.019s^3R(s) + \dots$$

and

$$\therefore e(t) = 0.1\dot{r}(t) + 0.09\ddot{r}(t) - 0.019\dddot{r}(t) + \dots$$

where

$$r(t) = a_0 + a_1 t + a_2 t^2$$

$$\dot{r}(t) = a_1 + 2a_2 t$$

$$\ddot{r}(t) = 2a_2$$

$$\dddot{r}(t) = 0$$

LECT. 10

Then

$$\begin{aligned}e(t) &= 0.1(a_1 + 2a_2t) + 0.09(2a_2) \\ &= 0.1a_1 + 0.18a_2 + 0.2a_2t\end{aligned}$$

The steady state error can be calculated as given below:

$$\begin{aligned}\text{steady state error} &= \lim_{t \rightarrow \infty} e(t) \\ &= \lim_{t \rightarrow \infty} (0.1a_1 + 0.18a_2 + 0.2a_2t)\end{aligned}$$

Then from the last equation, it is clear that the steady state error is infinite as $t \rightarrow \infty$.

End of Lecture ten

Computer control design and modeling

Lecture 11

Root Locus Techniques

Introduction

Root locus is a graphical method for sketching the locus of the closed-loop system's poles as a system parameter is varied. Root locus is a powerful method of analysis and design for stability and transient response which is applicable for higher-order systems.

Consider a feedback control system shown in figure 1.

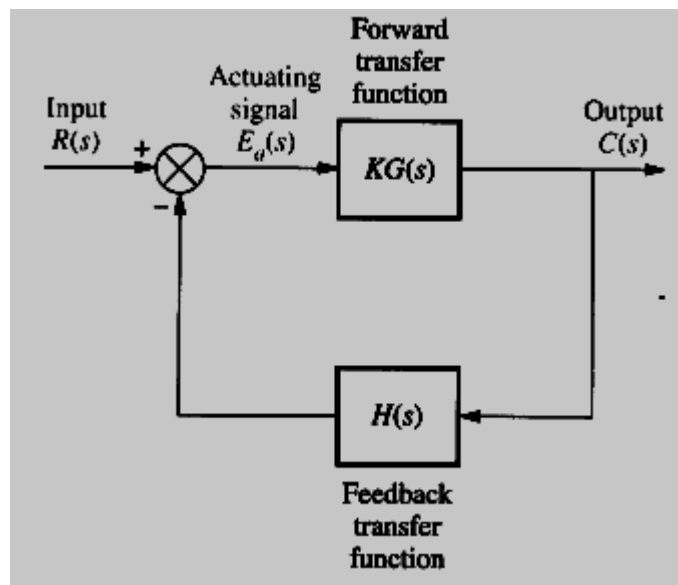


Figure 1: Feedback control system

The transfer function of the closed-loop system is given by the following formula

$$T(s) = \frac{KG(s)}{1 + KG(s)H(s)} \quad (1)$$

The equation

$$1 + KG(s)H(s) = 0$$

is called characteristic equation of the closed-loop system (1), and the roots of the characteristic equation are the poles of the closed loop system. For the systems of order higher than two it is usually hard to determine the exact location of the poles of the closed-loop system based on knowledge of poles

LECT. 11

location of $G(s)$ and $H(s)$. The root locus technique will be used to give us a picture of the poles of $T(s)$ as K is varied.

Definition of Root Locus

Definition 1. The root locus is the path of the roots of the characteristic equation traced out in the complex plane as a system parameter is changed.

Example. Root locus for a video camera control system

Consider a control system of an automatic video camera shown in figure 2.

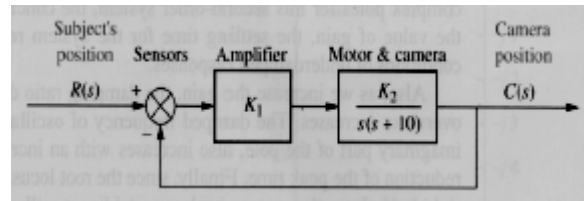


Figure 2: Automatic video camera control system

The closed-loop transfer function of ‘this system is as follows

$$\frac{C(s)}{R(s)} = \frac{K}{s^2 + 10s + K}$$

where $K = K_1K_2$. In figure 3, the pole location for different values of gain K is given.

K	Pole 1	Pole 2
0	-10	0
5	-9.47	-0.53
10	-8.87	-1.13
15	-8.16	-1.84
20	-7.24	-2.76
25	-5	-5
30	-5 + j2.24	-5 - j2.24
35	-5 + j3.16	-5 - j3.16
40	-5 + j3.87	-5 - j3.87
45	-5 + j4.47	-5 - j4.47
50	-5 + j5	-5 - j5

Figure 3: Pole location as a function of gain

The data of figure 3 is graphically displayed in figure 4 which shows each pole and it’s gain.

LECT. 11

One can see that for $K = 0$ poles are $p_1 = -10$, $p_2 = 0$. As K increases, the pole p_1 moves toward the right, while p_2 moves toward the left. For $K = 25$, the poles p_1 and p_2 meet at -5 , break away from the real axis, and move into the complex plane. We see that, if $0 < K < 25$, the poles are real and distinct, and the system is overdamped. For $K = 25$, the poles are real and multiple, and the system is critically damped. For $K > 25$, the poles are complex conjugate, and the system is underdamped.

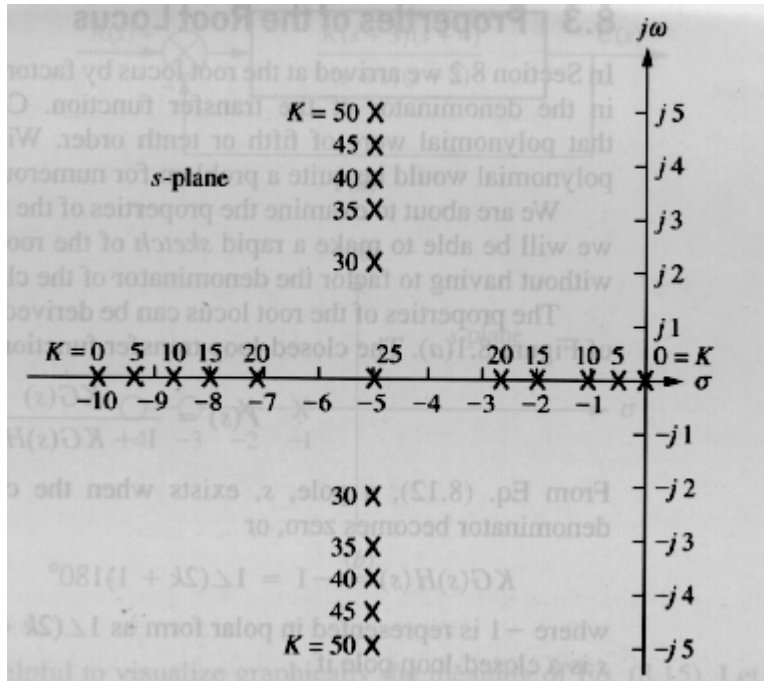


Figure 4: Pole plot

The presented analysis is almost obvious for a second order system. The important point is that the root locus technique allows us to sketch the root locus and make the analogous analysis for systems of order higher than two.

Properties of the Root Locus

In this section we will consider some properties of the root locus. Using these properties, we will be able to sketch the root locus for a higher-order system.

We start from the following observation. A complex number s is a pole of the closed-loop system (1), if and only if

$$KG(s)H(s) = -1.$$

The last equation is equivalent to the following two equations

LECT. 11

$$|KG(s)H(s)| = 1, \quad (2)$$

And

$$\text{Arg } KG(s)H(s) = (2k + 1)180^\circ, \quad k = 0, \pm 1, \pm 2 \dots \quad (3)$$

Now let us recall some properties of complex numbers. Consider a rational function of the form

$$F(s) = \frac{(s + z_1)(s + z_2) \dots (s + z_m)}{(s + p_1)(s + p_2) \dots (s + p_n)}$$

Then the modulus of F(s) can be calculated as follows

$$|F(s)| = \frac{|s + z_1||s + z_2| \dots |s + z_m|}{|s + p_1||s + p_2| \dots |s + p_n|} \quad (4)$$

It is clear that, given a complex number s, then $|s + z_1|$ is the magnitude of the vector drawn from the zero of F(s) at $-z_1$ to the point s, and, analogously, $|s + p_1|$ is the magnitude of the vector from the pole of F(s) at $-p_1$ to the point s.

Analogously, the argument $\theta = \text{Arg}F(s)$ is given by the following formula

$$\text{Arg } F(s) = \sum_{i=1}^m \text{Arg}(s + z_i) - \sum_{j=1}^n \text{Arg}(s + p_j) = \sum \text{zero angles} - \sum \text{pole angles.}$$

Using these properties, one can see from equation (3) that a point s of the complex plane is on the root locus for a particular value of gain K, if

$$\sum \text{zero angles of } G(s)H(s) - \sum \text{pole angles of } G(s)H(s) = (2k + 1)180^\circ, \quad k = 0, \pm 1, \pm 2 \dots \quad (5)$$

On the other hand, suppose a point s is on the root locus. Then the value of the gain K at this point can be found by the formula

$$K = \frac{1}{|G(s)H(s)|} = \frac{|s + p_1||s + p_2| \dots |s + p_n|}{|s + z_1||s + z_2| \dots |s + z_m|}, \quad (6)$$

Where p_j are poles and z_i are zeros of the open-loop transfer function $G(s)H(s)$.

LECT. 11

Example. Consider a system on figure 5.

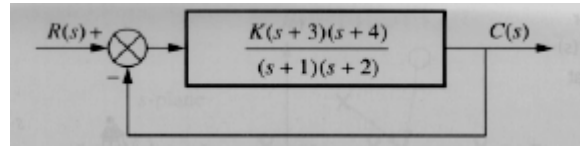


Figure 5: Feedback control system

The transfer function of the open-loop system (i.e. without feedback) is

$$KG(s)H(s) = \frac{K(s+3)(s+4)}{(s+1)(s+2)}$$

The closed-loop transfer function can be found as follows

$$T(s) = \frac{K(s+3)(s+4)}{(1+K)s^2 + (3+7K)s + (2+12K)}$$

If a point s is a pole of the closed-loop system for some value of K then (2) and (3) must be satisfied at this point.

First, let us check the point $s = -2 + j3$. If this point is on the root locus, then the sum of angles of the zeros of the open-loop transfer function minus sum of angles of the poles must be an odd multiple of 180° . From figure 6, we see that

$$\theta_1 + \theta_2 - \theta_3 - \theta_4 = 56.31^\circ + 71.57^\circ - 90^\circ - 108.43^\circ = -70.55^\circ$$

This value is not an odd multiple of 180° , therefore $s = -2 + j3$ is not a point on the root locus, i.e. it is not a pole of the closed-loop system for any gain K .

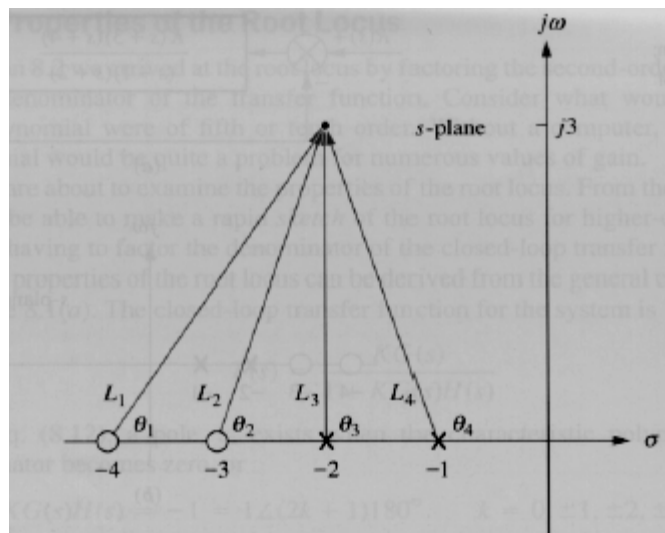


Figure 6: Pole-zero location

LECT. 11

On the other hand, for the point $s = -2 + j(\sqrt{2}/2)$ we have

$$\theta_1 + \theta_2 - \theta_3 - \theta_4 = -180^\circ,$$

Therefore, this point is on the root locus for some value of gain K . To find the value of K , one can use formula (6). We have

$$K = \frac{L_3 L_4}{L_1 L_2} = \frac{\sqrt{2} \times 1.22}{2 \times 2.12 \times 1.22} = 0.33.$$

Therefore, we get that for $K = 0.33$ the point $s = -2 + j(\sqrt{2}/2)$ is a pole of the closed loop system.

End of Lecture eleven

Computer control design and modeling

Lecture 12

Sketching the Root Locus

The following properties of the root locus allow us to sketch the root locus using minimal calculations.

Property 1: Number of branches. Each closed-loop pole moves as the gain is varied. Therefore, number of separate loci (or branches of the root locus) is equal to the number of poles.

Property 2: Symmetry about the real axis. Since the complex poles always appear in complex conjugate pairs, the root locus must be symmetrical about the real axis.

Property 3: Location of the real-axis segment of the root locus. The root locus on the real axis always lies in a section of the real axis to the left of an odd number of (open-loop) poles and zeros.

Property 4: Starting points and ending points. To determine starting and ending points of root locus, denote by $N_G(s)$, $D_G(s)$ the numerator and denominator polynomials of $G(s)$

Correspondingly, i.e.

$$G(s) = \frac{N_G(s)}{D_G(s)}$$

On the other hand, let $N_H(s)$ ($D_H(s)$) be the numerator (denominator) polynomial of $H(s)$. Using these notations, the transfer function of the closed-loop system can be rewritten as follows

$$\frac{C(s)}{R(s)} = \frac{\frac{KN_G(s)}{D_G(s)}}{1 + \frac{KN_G(s)N_H(s)}{D_G(s)D_H(s)}}$$

Thus, the characteristic equation of the system is as follows

$$1 + \frac{KN_G(s)N_H(s)}{D_G(s)D_H(s)} = 0,$$

or

$$KN_G(s)N_H(s) + D_G(s)D_H(s) = 0. \quad (1)$$

LECT. 12

If $K \rightarrow 0$, the equation (1) tends to the following one

$$D_G(s)D_H(s) = 0.$$

Solutions of the last equation are exactly the poles of the open-loop system. On the other hand, for positive $K > 0$ the equation (1) can be rewritten as

$$N_G(s)N_H(s) + \frac{1}{K}D_G(s)D_H(s) = 0. \quad (2)$$

If $K \rightarrow +\infty$, the last equation tends to the following one

$$N_G(s)N_H(s) = 0.$$

Solutions of this last equation are exactly the zeros of the open-loop system. Thus, the following rule is valid: root locus begins at the poles of $G(s)H(s)$ and ends at the zeros of $G(s)H(s)$ as K increases from 0 to $+\infty$.

Property 5: Location of infinite zeros. We will say that a function $F(s)$ has a zero (pole) at infinity if $F(s) \rightarrow 0$ ($F(s) \rightarrow \infty$) as $s \rightarrow \infty$. Every function has an equal number of zeros and poles if we include the infinite poles and zeros as well as finite poles and zeros. Consider, for example, a function

$$KG(s)H(s) = \frac{K}{s(s+1)(s+2)}. \quad (3)$$

This function has three finite poles at $s_1 = 0$, $s_2 = -1$, $s_3 = -2$, and no (finite) zeros. However, if s approaches infinity, the function becomes

$$KG(s)H(s) \approx \frac{K}{s \cdot s \cdot s}.$$

Each s in the denominator causes the function to become zero as s approaches infinity. Therefore, the function has three infinite zeros.

Thus, the root locus for equation (3) will begin at finite poles and end in infinite zeros. Where are the infinite zeros located? In general, if a function has n_p finite poles and n_z finite zeros ($n_p \geq n_z$), then $N = n_p - n_z$ sections (branches) of the root locus will end at infinite zeros. The following rule helps us to locate the infinite zeros.

The branches that end at infinite zeros approach the zeros along linear asymptotes. These asymptotes are centered at the point on the real axis given by

LECT. 12

$$\sigma_0 = \frac{\sum_{j=0} n_p \text{ finite poles} - \sum_{i=0} n_z \text{ finite zeros}}{n_p - n_z}, \quad (4)$$

and their angles are

$$\theta_k = \frac{(2k-1)180^\circ}{n_p - n_z}; \quad k = 1, 2, \dots, n_p - n_z. \quad (5)$$

Property 6: Location of the real-axis breakaway and break-in points. A point on the real axis is called **breakaway point** if the root locus departs from the real axis at this point. On the other hand, a point on the real axis is called **break-in point** if the root locus arrives to the real axis at this point. From the symmetry property it follows that the root loci at the breakaway (break-in) point are symmetrical with respect to the real axis.

How to find the breakaway (break-in) points? Suppose we have two real axis poles which move towards each other as gain increases. One can conclude that the gain must be maximal at the point where breakaway occurs. Thus, the breakaway point is the point of maximum gain between two open-loop real-axis poles.

Analogously, when the complex pair returns to the real axis, the gain will continue to increase as the closed-loop poles move toward the open-loop zeros. Therefore, one can conclude that the break-in point is the point of minimum gain between two real-axis zeros.

These considerations allow us to use the following method to find breakaway and break-in points. As we already know, for any points s of the root locus the following equation

$$K = -\frac{1}{H(s)G(s)} \quad (6)$$

is valid. On the real axis s is real, therefore $H(s)$ and $G(s)$ are real-valued function. To find points of maximum and minimum of K one can simply differentiate the equation (6) with respect to s and set the derivative equal to zero. Let's consider an example.

Example 1. Consider a unity negative feedback system with the following open-loop transfer function

$$KG(s)H(s) = \frac{K(s-3)(s-5)}{(s+1)(s+2)},$$

sketch the root locus.

Solution. The system has two poles at $p_1 = -1$, $p_2 = -2$, and two finite zeros at $z_1 = 3$ and $z_2 = 5$. The root locus has two branches that start from the

LECT. 12

poles, end in the zeros, and symmetrical with respect to real line. Using property 3, we see that the first real-axis segment of the root locus is located between -2 and -1 , and the second one is located between 3 and 5 . Let us calculate breakaway and break-in points. For the points of the root locus we have

$$\frac{K(s-3)(s-5)}{(s+1)(s+2)} = -1.$$

Solving for K yields

$$K = \frac{-(s^2 + 3s + 2)}{(s^2 - 8s + 15)}.$$

Differentiating with respect to s and setting the derivative equal to zero, we get

$$\frac{dK}{ds} = \frac{11s^2 - 26s - 61}{(s^2 - 8s + 15)^2} = 0.$$

Solving for s , we find $s_1 = -1.45$, $s_2 = 3.82$. Clearly, s_1 is the breakaway point, while s_2 is the break-in point. The root locus is shown in figure 1.

Another method to find the breakaway and break-in point is by using the following rule:

- Breakaway and break-in points satisfy the following relationships

$$\sum_{i=1}^m \frac{1}{s - z_i} = \sum_{j=1}^n \frac{1}{s - p_j}, \quad (7)$$

where z_i , and p_j are zeros and poles, respectively, of $G(s)H(s)$.

Example. For the previous example, find the breakaway and break-in points without differentiation.

Solution. Using (7), we get

$$\frac{1}{s-3} + \frac{1}{s-5} = \frac{1}{s+1} + \frac{1}{s+2}.$$

Simplifying, we get

$$11s^2 - 26s - 61 = 0,$$

and solving for s , we obtain $s_1 = -1.45$, $s_2 = 3.82$.

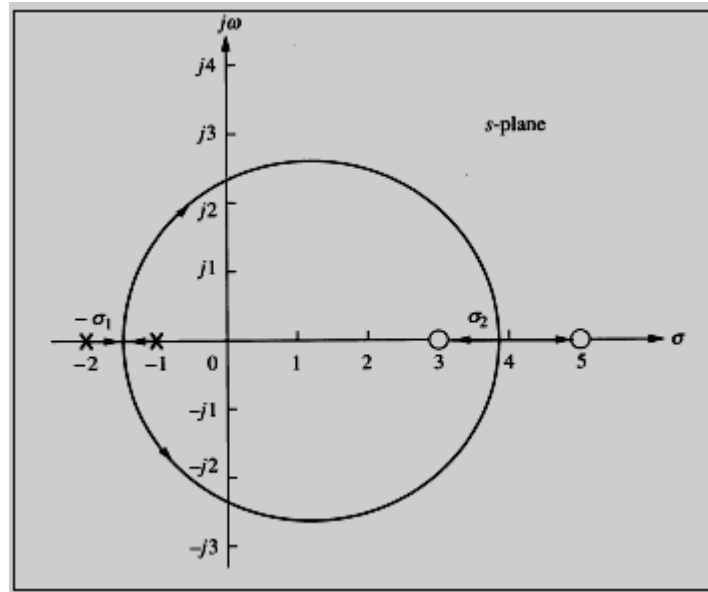


Figure 1: Root locus for Example 1

Property 7: Points of $j\omega$ -axis crossings. The points where the root locus crosses the $j\omega$ axis are important because they separate stable parts of root locus from unstable ones. To find the $j\omega$ -axis crossing, we can use the Routh-Hurwitz criterion as follows: forcing a row of zeros in the Routh table will yield the gain: going back one row to the even polynomial equation and solving for the roots (if possible) yields the points of the imaginary axis crossing.

Example 2. Consider a unity negative feedback system with the following open-loop transfer function

$$G(s) = \frac{K(s+3)}{s(s+1)(s+2)(s+4)}$$

Sketch the root locus and find the range of K such that the closed-loop system is stable.

Solution. The open-loop system has four poles $p_1 = 0$, $p_2 = -1$, $p_3 = -2$, $p_4 = -4$, and one finite zero $z_1 = -3$, therefore, there are three infinite zeros. Using property 3, we see that there are three real axis segments of the root locus: first between 0 and -1 , second between -2 and -3 , and the third one is to the left of -4 . To find the location of infinite zeros, let us calculate the asymptotes using formulas (4), (5). The asymptotes are centered at the point

$$\sigma_0 = \frac{\sum_{j=0} n_p \text{ finite poles} - \sum_{i=0} n_z \text{ finite zeros}}{n_p - n_z} = \frac{(-1 - 2 - 4) - (-3)}{4 - 1} = -\frac{4}{3}$$

LECT. 12

The angles of asymptotes that intersect at $-\frac{4}{3}$ are

$$\theta_k = \frac{(2k - 1)180^\circ}{n_p - n_z}; \quad k = 1, 2, \dots, n_p - n_z,$$

i.e.

$$\theta_1 = 60^\circ,$$

$$\theta_2 = 180^\circ,$$

and

$$\theta_3 = 300^\circ.$$

A breakaway point must exist between the poles -1 and 0 , and the corresponding branches tend to infinite zeros along the asymptotes with angles θ_1 and θ_3 . Now, let us calculate the points where the root locus crosses the imaginary axis. First, one can find the transfer function of the closed loop system as follows

$$T(s) = \frac{K(s + 3)}{s^4 + 7s^3 + 14s^2 + (8 + K)s + 3K}. \quad (8)$$

The Routh table for the system (8) is given in figure 2.

s^4	1	14	$3K$
s^3	7	$8 + K$	
s^2	$90 - K$	$21K$	
s^1	$\frac{-K^2 - 65K + 720}{90 - K}$		
s^0	$21K$		

Figure 2: Routh table for Example 2

We know that a complete row of zeros yields the possibility for imaginary axis roots. For this system, only the s^1 row can yield a row of zeros. This happens if

$$-K^2 - 65K + 720 = 0.$$

Solving for K , we get

$$K = 9.65.$$

LECT. 12

Going one row back, one can form the following even polynomial

$$(90 - K)s^2 + 21K = 0,$$

and for $K = 9.65$ we get

$$80.35s^2 + 202.65 = 0.$$

Solving for s , we get $s_{1,2} = \pm j1.59$. Thus, the root locus crosses the imaginary axis at $\pm j1.59$ at a gain $K = 9.65$. The system is stable for $0 \leq K < 9.65$. The root locus is sketched in figure 3.

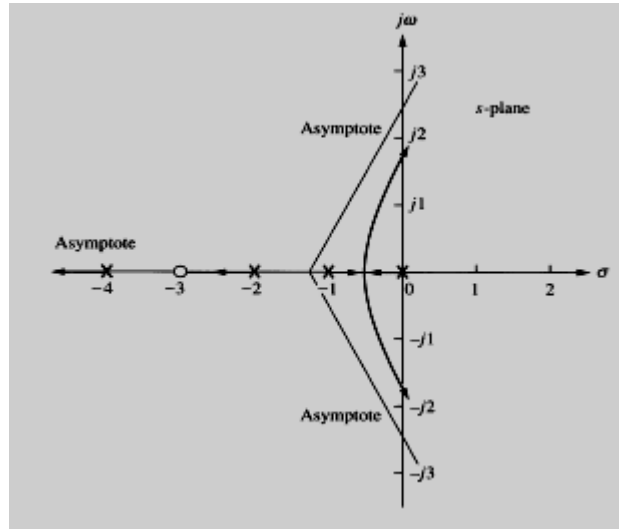


Figure 3: Root locus for Example 2

Property 8: Angles of Departure and Arrival. One can find angles of departure from the complex poles as well as angles of arrival to the complex zeros as follows. Consider figure 4, which shows the open-loop poles and zeros, some of them are complex. Take a point ϵ of the root locus close to a complex pole, then the sum of angles drawn from all finite poles and zeros to this pole is equal to an odd multiple of 180° . Assume that all angles from all other poles and zeros are drawn directly to the pole that is near the point. The only unknown angle is the angle drawn from the pole that is close to ϵ . We can solve for this unknown angle, which is actually the angle of departure from this complex pole. For example, for the pole-zero plot in figure 4, we have

$$-\theta_1 + \theta_2 + \theta_3 - \theta_4 - \theta_5 + \theta_6 = (2k + 1) 180^\circ,$$

which implies

$$\theta_1 = \theta_2 + \theta_3 - \theta_4 - \theta_5 + \theta_6 - (2k + 1) 180^\circ.$$

LECT. 12

Angle of arrival to a complex zero can be calculated analogously. For example (for figure 5)

$$\theta_2 = \theta_1 - \theta_3 + \theta_4 + \theta_5 - \theta_6 + (2k + 1) 180^\circ.$$

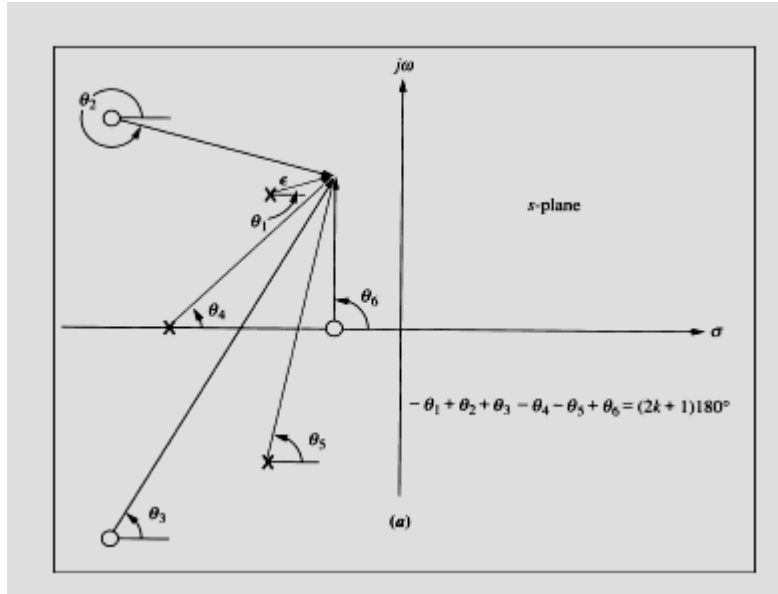


Figure 4: Calculation of angle of departure

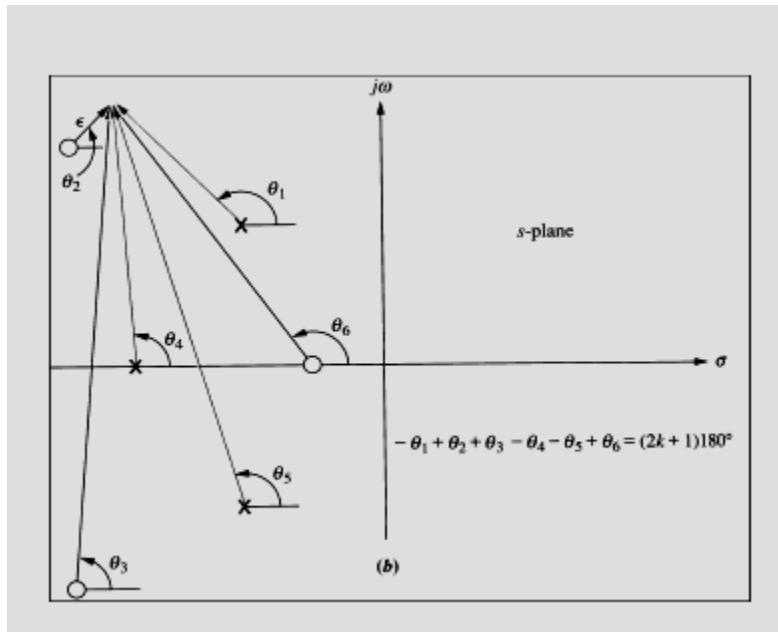


Figure 5: Calculation of angle of arrival

LECT. 12

Example 3. Consider a unity negative feedback system with the following open-loop transfer function

$$G(s) = \frac{K(s+2)}{s^2 + 2s + 3}$$

Sketch the root locus.

Solution. The open-loop system has two poles $p_1 = -1+j\sqrt{2}$ and $p_2 = -1-j\sqrt{2}$, and one finite zero $z_1 = -2$, therefore, there is one infinite zero. Using property 3, we see that there is one real axis segment of the root locus between -2 and $-\infty$. To find the location of infinite zero, let us calculate the asymptotes using formulas (4), (5). The asymptotes are centered at the point

$$\sigma_0 = \frac{-1 + j\sqrt{2} - 1 - j\sqrt{2} + 2}{1} = 0$$

The angles of asymptotes are

$$\theta_k = \frac{(2k-1)180}{n_p - n_z}; k = 1, 2, \dots, n_p - n_z$$

i.e.

$$\theta_1 = 180^\circ$$

To determine the angle of departure from the complex-conjugate open-loop poles, we use the following equation:

$$\theta_1 = 180 - \theta_2 + \theta_3$$

Where

θ_1 : The angle of departure

θ_2 : The angle between p_1 and p_2

θ_3 : The angle between p_1 and z_1

Then

$$\theta_1 = 180 - 90 + 55 = 145^\circ$$

And since the root locus is symmetric about the real axis, the angle of departure from the pole at $s = -p_2$ is -145° .

Let us calculate breakaway and break-in points. For the points of the root locus we have

LECT. 12

$$k = -\frac{(s^2 + 2s + 3)}{s + 2}$$

We have

$$\frac{dk}{ds} = -\frac{(2s + 2)(s + 2) - (s^2 + 2s + 3)}{(s + 2)^2}$$

which gives

$$s^2 + 4s + 1 = 0$$

$$\text{or } s = -3.73 \quad \text{or } s = -0.268$$

It is clear that points $s = -3.73$ is on the root locus. Hence this point is an actual break-in point. Then the root locus is shown in figure 6.

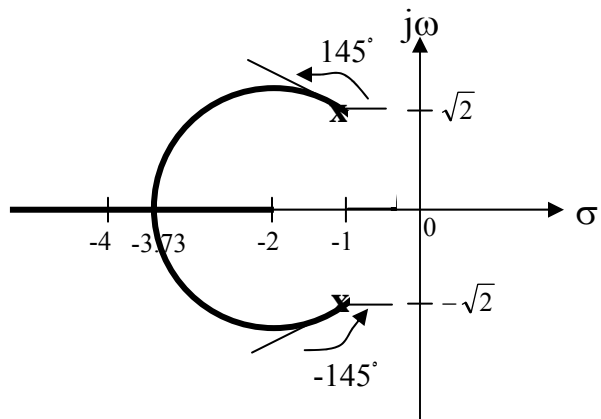


Figure 6: Root locus of Example 3

End of Lecture twelve

Computer control design and modeling

Lecture 13

Design of Feedback Control Systems

The design of a control system is concerned with the arrangement of the system structure and the selection of suitable components and parameters to meet requirements of stability and performance specifications. The alteration or adjustment of a control system in order to provide a suitable performance is called compensation. To alter the system response, an additional component is inserted within the structure of the feedback system. This additional component is often called a compensator or a controller.

Two basic configurations of compensation are used in feedback control system design. They are a) cascade compensation, and b) feedback compensation. With cascade compensation, the compensator $G_1(s)$ is placed at the low-power end of the forward path in cascade with the plant (figure 1, (a)). In feedback compensation scheme, the compensator $H_1(s)$ is placed in the feedback path (figure 1, (b)).

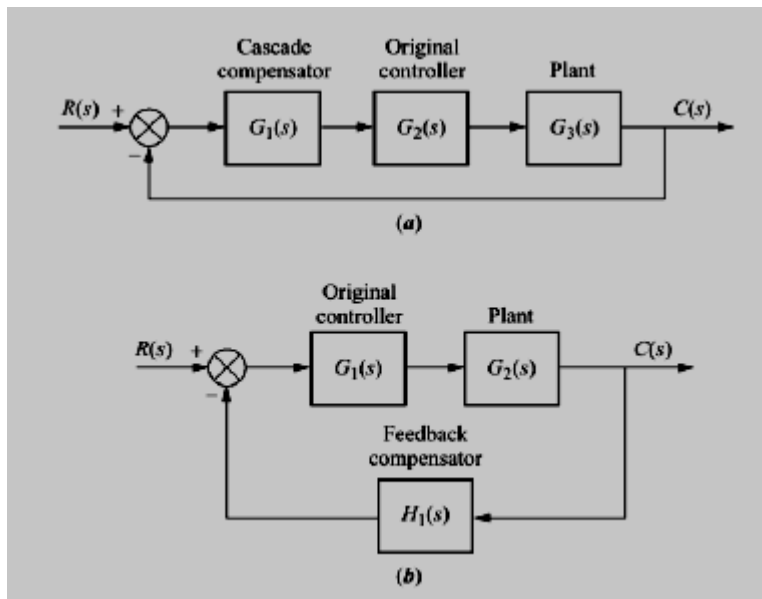


Figure 1: Compensation configurations: a) cascade; b) feedback.

Improving steady-state error via cascade compensation

There are two ways to improve the steady-state error of a feedback control system using cascade compensation. These are

- ideal integral compensation;
- lag compensation.

Ideal integral compensation

Steady-state error can be improved by placing an open-loop pole at the origin. For example, if the uncompensated system does not have pure integrations, then its response to a step input has finite steady-state error. After adding a pole at the origin, the system responds to a step input with zero steady-state error. To see how to improve the steady-state error without affecting the transient response, look at figures 2 and 3. Suppose the uncompensated system has desirable transient response generated by the closed-loop pole at A. If we add a pole at the origin, then the angular contribution of the open-loop poles at the point A is no longer 180° and, therefore, A is no longer a pole of the closed-loop system (figure 2, (b)). To solve the problem, one should add a zero very close to the pole at the origin (figure 3)). Now, the angular contribution of the compensator zero and compensator pole cancel out, and the point A is still on the root locus. Thus, we have improved the steady-state error without affecting the transient response.

A method of implementing an ideal integral compensator is shown in figure 4.

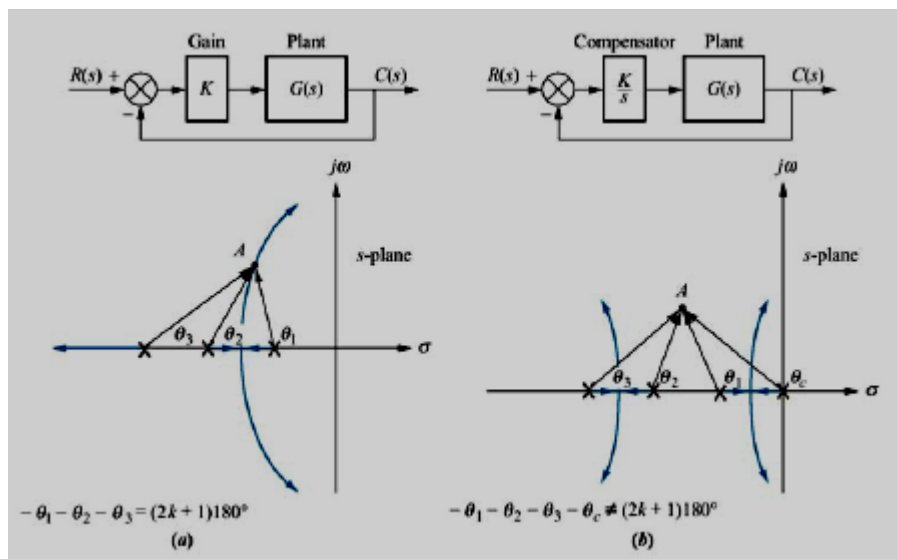


Figure 2: Pole at A is: a) on the root locus without compensator; b) not on the root locus with compensator pole added.

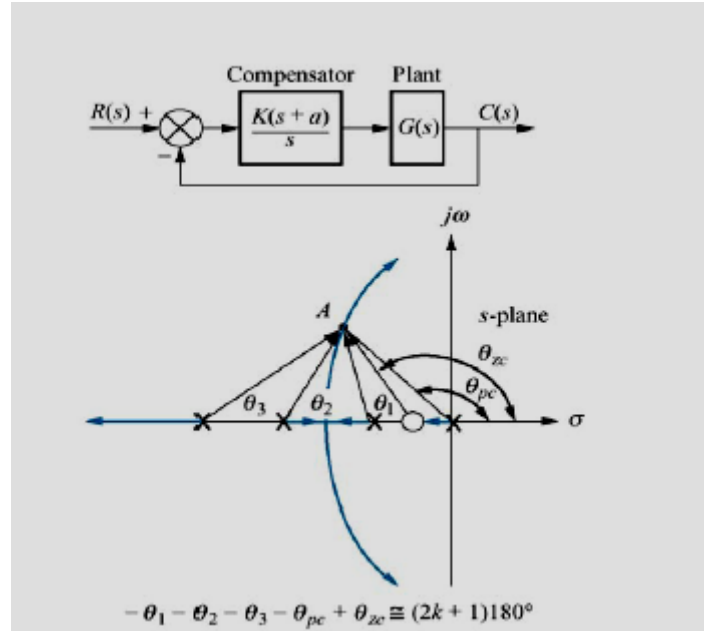


Figure 3: Pole at A is approximately on the root locus with compensator pole and zero added

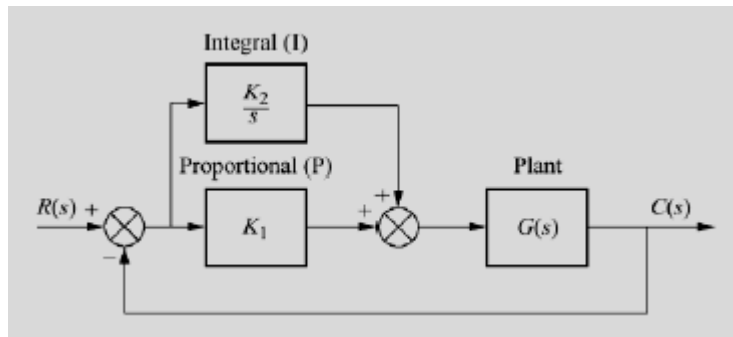


Figure 4: PI controller

The compensator consists of proportional K_1 and integral K_2/s blocks. The transfer function of the compensator is

$$G_c(s) = K_1 + \frac{K_2}{s} = \frac{K_1 \left(s + \frac{K_2}{K_1} \right)}{s},$$

so that the value of an additional zero can be adjusted by varying K_2/K_1 . Since the compensator has both proportional and integral control, the ideal

integral controller (compensator) is given the alternate name PI (Proportional-Integral) Controller.

Lag compensation

The drawback of ideal compensators is that they must be implemented with active network, which require the use of active amplifiers and additional power sources. If we use passive networks (which consist of resistors and capacitors and do not require additional power sources), additional pole and zero are moved to the left, close to origin, as shown in figure 5. This placement of the pole, although it does not increase the number of integrators, does yield the improvement in the static error constant over an uncompensated system.

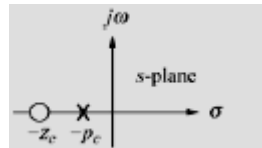


Figure 5: Compensator pole-zero plot

It is important to note that, to keep transient response characteristics unchanged, the compensator zero and compensator pole must be very close to each other. Thus, to obtain the improvement in steady-state error, one can place the compensator's zero-pole pair very close to origin.

Lag compensation techniques based on the root-locus approach

Consider the problem of finding a suitable compensation network for the case where the system exhibits satisfactory transient response characteristics but unsatisfactory steady-state error characteristics. Compensation in this case essentially consists of increasing the open loop gain without appreciably changing the transient response characteristics. This means that the root locus in the neighborhood of the dominant closed-loop poles should not be changed appreciably, but the open-loop gain should be increased as much as needed. This can be accomplished if a lag compensator is put in cascade with the given feedforward transfer function.

Consider a lag compensator $G_c(s)$, where

$$G_c(s) = \hat{K}_c \beta \frac{Ts + 1}{\beta Ts + 1} = \hat{K}_c \frac{s + \frac{1}{T}}{s + \frac{1}{\beta T}} \quad (1)$$

LECT. 13

An increase in the gain means an increase in the static error constants. If the open-loop transfer function of the uncompensated system is $G(s)$. Then the static velocity error constant K_v of the uncompensated system is

$$K_v = \lim_{s \rightarrow 0} sG(s)$$

If the compensator is chosen as given by Equation (1), then for the compensated system with the open-loop transfer function $G_c(s)G(s)$ the static velocity error constant \hat{K}_v becomes

$$\begin{aligned}\hat{K}_v &= \lim_{s \rightarrow 0} sG_c(s)G(s) \\ &= \lim_{s \rightarrow 0} G_c(s)K_v \\ &= \hat{K}_c \beta K_v\end{aligned}$$

Thus if the compensator is given by Equation (1), then the static velocity error constant is increased by a factor of $\hat{K}_c \beta$, where \hat{K}_c is approximately unity.

Design procedures for lag compensation by the root locus method

The procedure for designing lag compensators for the system shown in figure (6) by the root locus method may be stated as follows:

1. Draw the root locus plot for the uncompensated system whose open-loop transfer function is $G(s)$. Based on the transient response specifications, locate the dominant closed-loop poles on the root locus.
2. Assume the transfer function of the lag compensator to be

$$G_c(s) = \hat{K}_c \beta \frac{Ts + 1}{\beta Ts + 1} = \hat{K}_c \frac{s + \frac{1}{T}}{s + \frac{1}{\beta T}}$$

Then the open-loop transfer function of the compensated system becomes $G_c(s)G(s)$.

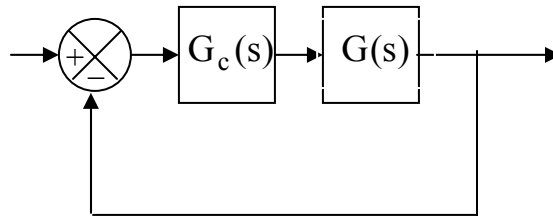


Figure 6: Control system

LECT. 13

3. Evaluate the particular static error constant specified in the problem.
4. Determine the amount of the increase in the static error constant necessary to satisfy the specifications.
5. Determine the pole and zero of the lag compensator that produce the necessary increase in the particular static error constant without appreciably altering the original root loci.

NOTE: the ratio of the value required in the specifications and the gain found in the uncompensated system is the required ratio between the distance of the zero from the origin and that of the pole from the origin.

6. Draw a new root locus plot for the compensated system. Locate the desired dominant closed-loop poles on the root locus.
7. Adjust gain \hat{K}_c of the compensator from the magnitude condition so that dominant closed-loop poles lie at the desired location.

Example: Consider the system with the following feedforward transfer function is:

$$G(s) = \frac{1.06}{s(s+1)(s+2)}$$

It is desired to increase the static velocity error constant K_v to about 5 sec^{-1} using lag compensator.

Solution:

The closed-loop transfer function becomes

$$\begin{aligned} \frac{C(s)}{R(s)} &= \frac{1.06}{s(s+1)(s+2) + 1.06} \\ &= \frac{1.06}{(s + 0.3307 - j0.5864)(s + 0.3307 + j0.5864)(s + 2.3386)} \end{aligned}$$

The dominant closed loop poles are

$$s = -0.3307 \pm j0.5864$$

To find the damping ratio and the natural frequency of the dominant closed loop poles, we must use the following plot:

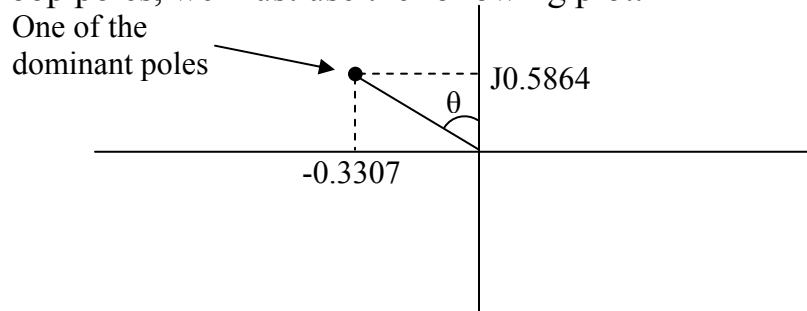


Figure 7:
Illustration of using dominant pole location to calculate ζ and ω_n

LECT. 13

Then

$$\zeta = \sin(\theta) = \sin(\tan^{-1} \frac{0.3307}{0.5864}) = 0.491.$$

$$\omega_n = \sqrt{(0.3307)^2 + (0.5864)^2} = 0.673.$$

$$\text{And } K_v = \lim_{s \rightarrow 0} sG(s) = \frac{1.06}{2} = 0.53 \text{ sec}^{-1}.$$

To increase the static error constant by a factor of 10, let us choose $\beta = 10$ and place the zero and the pole of the lag compensator at $s = -0.05$ and $s = -0.005$, respectively. The transfer function of the lag compensator becomes:

$$G_c = \hat{K}_c \frac{s + 0.05}{s + 0.005}$$

The open loop transfer function of the compensated system then becomes

$$\begin{aligned} G_c(s)G(s) &= \hat{K}_c \frac{s + 0.05}{s + 0.005} \frac{1.06}{s(s + 1)(s + 2)} \\ &= \frac{K(s + 0.05)}{s(s + 0.005)(s + 1)(s + 2)} \end{aligned}$$

where

$$K = 1.06 \hat{K}_c$$

The block diagram of the compensated system is shown in figure (8)

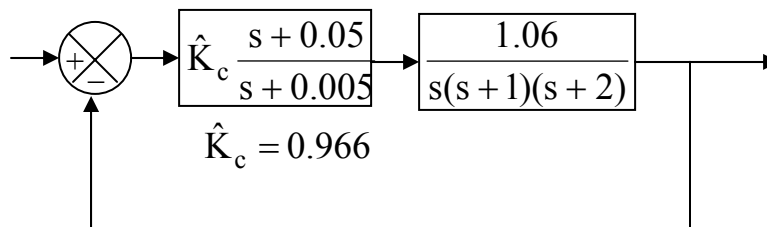


Figure 8: Compensated system

If the damping ratio of the new dominant closed-loop poles is kept the same, then the poles are obtained as given below:

$$s_1 = -0.31 + j 0.55, \quad s_2 = -0.31 - j 0.55$$

The open-loop gain K is

$$\begin{aligned} K &= \left| \frac{s(s + 0.005)(s + 1)(s + 2)}{s + 0.05} \right|_{s=-0.31+j0.55} \\ &= 1.0235 \end{aligned}$$

LECT. 13

Then the lag compensator gain \hat{K}_c is determined as

$$\hat{K}_c = \frac{K}{1.06} = \frac{1.0235}{1.06} = 0.9656$$

Thus the transfer function of the lag compensator designed is

$$G_c(s) = 0.9656 \frac{s + 0.05}{s + 0.005} = 9.656 \frac{20s + 1}{200s + 1}$$

Then the compensated system has the following open-loop transfer function:

$$\begin{aligned} G_1(s) &= \frac{1.0235(s + 0.05)}{s(s + 0.005)(s + 1)(s + 2)} \\ &= \frac{5.12(20s + 1)}{s(200s + 1)(s + 1)(0.5s + 1)} \end{aligned}$$

The static velocity error constant \hat{K}_v is

$$\hat{K}_v = \lim_{s \rightarrow 0} sG_1(s) = 5.12 \text{ sec}^{-1}$$

In the compensated system, the static velocity error constant has increased to 5.12 sec^{-1} . $5.12/0.53 = 9.66$ times the original value. (The steady-state error with ramp inputs has decreased to about 10 % of that of the original system). We have essentially accomplished the design objective of increasing the static velocity error constant to about 5 sec^{-1} .

End of Lecture thirteen

Computer control design and modeling

Lecture 14

Improving transient response

When dealing with improving of transient response, the objective typically is to design a response that has a desirable percent overshoot and a shorter settling time than the uncompensated system. As well as in the case of improving of steady-state error characteristics, there exists two ways to improve the transient response of a feedback control system using cascade compensation:

- ideal derivative compensation;
- lead compensation.

Ideal derivative compensation

One way to speed up the response of the system is to add a single zero to the forward path. This zero can be represented by a compensator with transfer function

$$G_c(s) = s + z_c.$$

This function is a sum of differentiator and a pure gain. It is called an ideal derivative or PD (proportional-derivative) controller. The structure of PD controller is shown in figure 1.

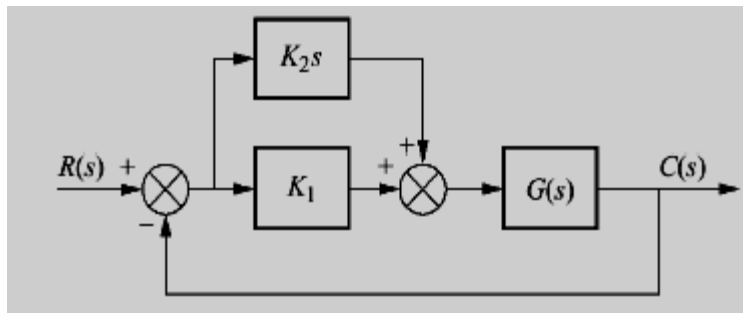


Figure 1: PD controller

In this figure, the transfer function of the controller is

$$G_c(s) = K_2s + K_1 = K_2 \left(s + \frac{K_1}{K_2} \right).$$

LECT. 14

Hence, K_1/K_2 is chosen to equal the negative of the compensator zero, and K_2 is chosen to contribute to the required loop-gain value. Transient response unattainable by a simple gain adjustment can be obtained by augmenting the systems poles and zeros with PD compensator.

While the ideal derivative compensator can improve the transient response of the system, it has two drawbacks. First, it requires an active circuit to perform the differentiation. Second, differentiation is a noisy process: if the input noise has low level but high frequency, the result of differentiating may lead to large unwanted signals. The lead compensator is a passive network used to overcome the disadvantages of ideal differentiation and still retain the ability to improve the transient response.

Lead compensation

When passive networks are used, a single zero cannot be produced; rather, a compensator zero and pole result. However, the advantages of passive networks are: a) no additional power supplies are required, and b) noise due to differentiation is reduced.

Lead compensation techniques based on the root-locus approach

Consider a design problem in which the original system either is unstable for all values of gain or is stable but has undesirable transient response characteristics. In such a case, the reshaping of the root locus is necessary in the broad neighborhood of the $j\omega$ axis and the origin in order that the dominant closed-loop poles be at desired locations in the complex plane. This problem may be solved by inserting an appropriate lead compensator in cascade with the feedforward transfer function.

The procedures for designing a lead compensator for the system shown in figure 2 by the root locus method may be stated as follows:

1. From the performance specifications, determine the desired location for the dominant closed-loop poles.
2. By drawing the root locus plot, ascertain whether or not the gain adjustment alone can yield the desired closed-loop poles. If not, calculate the angle deficiency ϕ . This angle must be contributed by the lead compensator if the new root locus is to pass through the desired locations for the dominant closed-loop poles.
3. Assume the lead compensator $G_c(s)$ to be

LECT. 14

$$G_c(s) = K_c \alpha \frac{Ts + 1}{\alpha Ts + 1} = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}} \quad (0 < \alpha < 1) \quad (1)$$

Where α and T are determined from the angle deficiency. K_c is determined from the requirement of the open-loop gain.

4. If static error constants are not specified, determine the location of the pole and zero of the lead compensator so that the lead compensator will contribute the necessary angle ϕ . If no other requirements are imposed on the system, try to make the value of α as large as possible. A larger value of α generally results in a larger value of K_v which is desirable.

5. Determine the open-loop gain of the compensated system from the magnitude condition.

Once a compensator has been designed, check to see whether all performance specifications have been met. If the compensated system does not meet the performance specifications, then repeat the design procedure by adjusting the compensator pole and zero until all such specifications are met. If a large error constant is required, cascade a lag network or alter the lead compensator to a lag-lead compensator.

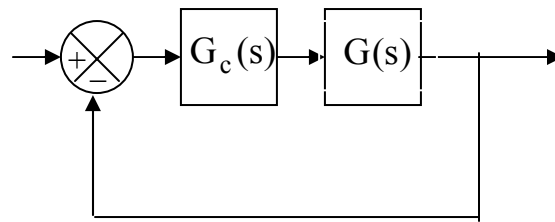


Figure 2: Control system

Example: Consider the system shown in figure 3.

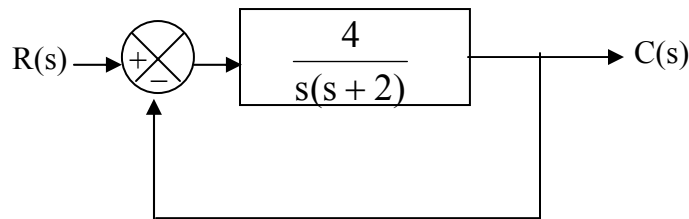


Figure 3: Control system example

It is desired to modify the closed-loop poles so that $\omega_n = 4$ rad/sec is obtained, without changing the value of ζ using lead compensator.

LECT. 14

Solution:

The root-locus plot for this system is shown in figure 4.

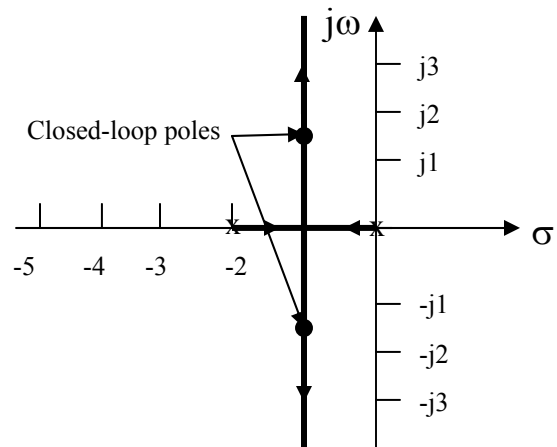


Figure 4: Root-locus of the system shown in figure 3

The closed-loop transfer function becomes

$$\begin{aligned}\frac{C(s)}{R(s)} &= \frac{4}{s^2 + 2s + 4} \\ &= \frac{4}{(s + 1 + j\sqrt{3})(s + 1 - j\sqrt{3})}\end{aligned}$$

The closed-loop poles are located at

$$s = -1 \pm j\sqrt{3}$$

The damping ratio of the closed-loop poles is 0.5. The undamped natural frequency of the closed-loop poles is 2 rad/sec. The static velocity error constant is 2 sec⁻¹.

In the present example, the desired locations of the closed-loop poles are

$$s = -2 \pm j2\sqrt{3}$$

In some cases, after the root loci of the original system have been obtained, the dominant closed-loop poles may be moved to the desired location by simple gain adjustment. This is however, not the case for the present system. Therefore, we shall insert a lead compensator in the feedforward path.

LECT. 14

A general procedure for determining, the lead compensator is as follows: First, find the sum of the angles at the desired location of one of the dominant closed-loop poles with the open-loop poles and zeros of the original system, and determine the necessary angle ϕ to be added so that the total sum of the angles is equal to $\pm 180^\circ(2k + 1)$. The lead compensator must contribute this angle ϕ . (If the angle ϕ is quite large, then two or more lead networks may be needed rather than a single one.)

If the original system has the open-loop transfer function $G(s)$, then the compensated system will have the open-loop transfer function:

$$G_c(s)G(s) = \left(K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}} \right) G(s) \quad \text{(Use equation (1))}$$

It is clear that there are many possible values for T and α that will yield the necessary angle contribution at the desired closed-loop poles.

The next step is to determine the locations of the zero and pole of the lead compensator. There are many possibilities for the choice of such locations. In what follows, we shall introduce a procedure to obtain the largest possible value for α (a larger value of α will produce a larger value of K_v . In most cases, the larger the K_v is, the better the system performance.) First, draw a horizontal line passing through point P , the desired location for one of the dominant closed-loop poles. This is shown as line PA in figure 5. Draw also a line connecting point P and the origin. Bisect the angle between the lines PA and PO , as shown in figure 5. Draw two lines PC and PD that make angles $\pm \phi/2$ with the bisector PB . The intersections of PC and PD with the negative real axis give the necessary location for the pole and zero of the lead network. The compensator thus designed will make point P a point on the root locus of the compensated system. The open-loop gain is determined by use of the magnitude condition.

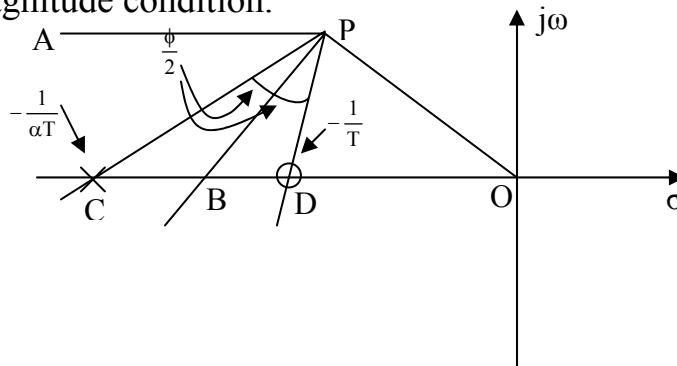


Figure 5: Determination of the pole and zero of a lead network

LECT. 14

In the present system, the angle of $G(s)$ at the desired closed-loop pole is

$$\left| \frac{4}{s(s+2)} \right|_{s=-2+j2\sqrt{3}} = -210^\circ$$

Thus, if we need to force the root locus to go through the desired closed-loop pole, the lead compensator must contribute $\phi = 30^\circ$ at this point. By following the foregoing design procedure, we determine the zero and pole of the lead compensator, as shown in figure 6, to be

$$\text{Zero at } s = -2.9, \quad \text{Pole at } s = -5.4$$

Or

$$T = \frac{1}{2.9} = 0.345, \quad \alpha T = \frac{1}{5.4} = 0.185$$

Thus $\alpha = 0.537$.

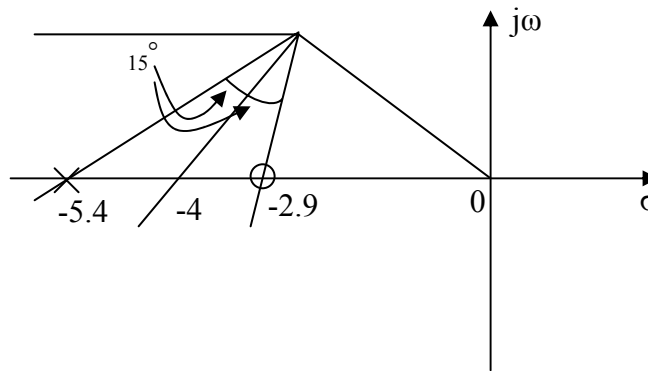


Figure 6: Determination of the pole and zero of the lead network for the above example

The open-loop transfer function of the compensated system becomes

$$G_c(s)G(s) = K_c \frac{s+2.9}{s+5.4} \frac{4}{s(s+2)} = \frac{K(s+2.9)}{s(s+2)(s+5.4)}$$

Where $K=4K_c$. The gain K is evaluated from the magnitude condition as follows:

$$\left| \frac{K(s+2.9)}{s(s+2)(s+5.4)} \right|_{s=-2+j2\sqrt{3}} = 1$$

Or

$$K = 18.7$$

LECT. 14

It follows that

$$G_c(s)G(s) = \frac{18.7(s+2.9)}{s(s+2)(s+5.4)}$$

The constant K_c of the lead compensator is

$$K_c = \frac{18.7}{4} = 4.68$$

Hence, $K_c\alpha = 2.51$. The lead compensator, therefore, has the transfer function

$$G_s(s) = 2.51 \frac{0.345s+1}{0.185s+1} = 4.68 \frac{s+2.9}{s+5.4}$$

The static velocity error constant K_v is obtained from the expression

$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} sG_c(s)G(s) \\ &= \lim_{s \rightarrow 0} \frac{s18.7(s+2.9)}{s(s+2)(s+5.4)} \\ &= 5.02 \text{ sec}^{-1} \end{aligned}$$

End of Lecture fourteen

Computer control design and modeling

Lecture 15

Improving Transient response and Steady-State Error PID controllers and Lag-Lead Compensators

To obtain simultaneous improvement in steady-state error and transient response, one can combine the methods described in the previous lectures.

PID controller design

A PID controller is a combination of PI and PD controllers. Its structure is shown in figure 1. A PID controller is described by the following formula

$$G_c(s) = K_1 + \frac{K_2}{s} + K_3s = \frac{K_3 \left(s^2 + \frac{K_1}{K_3}s + \frac{K_2}{K_3} \right)}{s}$$

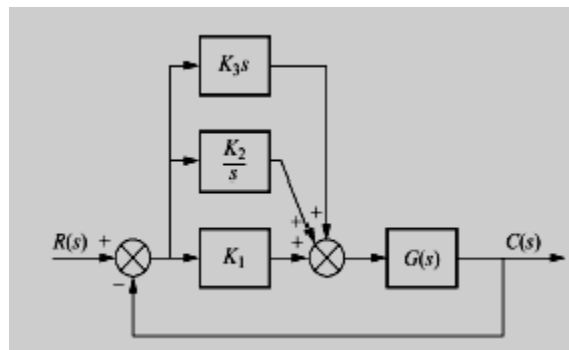


Figure 1: PID controller

The procedure for designing PID controller consists of the following steps:

- Evaluate the performance of the uncompensated system to determine how much improvement in transient response is required.
- Design the PD controller to meet the transient response specifications. The design includes the zero location and the loop gain.
- Design the PI controller to yield the required steady-state error.
- Determine the gains, K_1 , K_2 , and K_3 .
- Simulate the system to be sure that all requirements have been met.
- Redesign if simulation shows that requirements have not been met.

Lag-lead compensators

When passive networks are used, simultaneous improvement of transient response and steady-state error can be obtained using combination of a lag

compensator and a lead compensator rather than the ideal PID compensator. The resulting compensator is called lag-lead compensator.

Lag-lead compensation techniques based on the root-locus approach

Consider the system shown in figure 2.

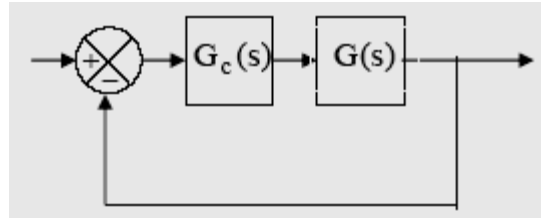


Figure 2: Control system

Assume that we use the lag-lead compensator:

$$G_c(s) = K_c \frac{\beta}{\gamma} \frac{(T_1s + 1)(T_2s + 1)}{\left(\frac{T_1}{\gamma}s + 1\right)(\beta T_2s + 1)} = K_c \left(\frac{s + \frac{1}{T_1}}{s + \frac{\gamma}{T_1}} \right) \left(\frac{s + \frac{1}{T_2}}{s + \frac{1}{\beta T_2}} \right) \quad (1)$$

Where $\beta > 1$ and $\gamma > 1$. (K_c is considered to belong to the lead portion of the lag-lead compensator.)

Case 1. $\gamma \neq \beta$. In this case, the design process is a combination of the design of the lead compensator and that of the lag compensator. The design procedure for the lag-lead compensator is as follows:

1. From the given performance specifications, determine the desired location for the dominant closed-loop poles.
2. Using the uncompensated open-loop transfer function $G(s)$, determine the angle deficiency ϕ if the dominant closed-loop poles are to be at the desired location. The lead portion of the lag-lead compensator must contribute this angle ϕ .
3. Assuming that we later choose T_2 sufficiently large so that the magnitude of the lag portion

$$\left| \frac{s_1 + \frac{1}{T_2}}{s_1 + \frac{1}{\beta T_2}} \right|$$

LECT. 15

is approximately unity, where $s = s_1$ is one of the dominant closed-loop poles, choose the values of T_1 and γ from the requirement that

$$\frac{s_1 + \frac{1}{T_1}}{s_1 + \frac{\gamma}{T_1}} = \phi$$

The choice of T_1 and γ is not unique. (Infinitely many sets of T_1 and γ are possible.) Then determine the value of K_c from the magnitude condition:

$$\left| K_c \frac{s_1 + \frac{1}{T_1}}{s_1 + \frac{\gamma}{T_1}} G(s_1) \right| = 1$$

4. If the static velocity error constant K_v is specified, determine the value of β to satisfy the requirements for K_v . The static velocity error constant K_v is given by

$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} s G_c(s) G(s) \\ &= \lim_{s \rightarrow 0} s K_c \left(\frac{s + \frac{1}{T_1}}{s + \frac{\gamma}{T_1}} \right) \left(\frac{s + \frac{1}{T_2}}{s + \frac{1}{\beta T_2}} \right) G(s) \\ &= \lim_{s \rightarrow 0} s K_c \frac{\beta}{\gamma} G(s) \end{aligned}$$

Where K_c and γ are already determined in step 3. Hence, given the value of K_v , the value of β can be determined from this last equation. Then, using the value of β thus determined, choose the value of T_2 such that

$$\left| \frac{s_1 + \frac{1}{T_2}}{s_1 + \frac{1}{\beta T_2}} \right| \approx 1$$

$$-5^\circ \left\langle \frac{s_1 + \frac{1}{T_2}}{s_1 + \frac{1}{\beta T_2}} \right\rangle 0^\circ$$

Case 2. $\gamma = \beta$. If $\gamma = \beta$ is required in equation (1), then the preceding design procedure for the lag-lead compensator may be modified as follows:

1. From the given performance specifications, determine the desired location for the dominant closed-loop poles.
2. The lag-lead compensator given by equation (1) is modified to

$$G_c(s) = K_c \frac{(T_1s + 1)(T_2s + 1)}{\left(\frac{T_1}{\beta}s + 1\right)(\beta T_2s + 1)} = K_c \left(\frac{s + \frac{1}{T_1}}{s + \frac{\beta}{T_1}} \right) \left(\frac{s + \frac{1}{T_2}}{s + \frac{1}{\beta T_2}} \right) \quad (2)$$

Where $\beta > 1$. The open-loop transfer function of the compensated system is $G_c(s)G(s)$. If the static velocity error constant K_v is specified, determine the value of constant K_c from the following equation:

$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} s G_c(s)G(s) \\ &= \lim_{s \rightarrow 0} s K_c G(s) \end{aligned}$$

3. To have the dominant closed-loop poles at the desired location, calculate the angle contribution ϕ needed from the lead portion of the lag-lead compensator.
4. For the lag-lead compensator, we later choose T_2 sufficiently large so that

$$\left| \frac{s_1 + \frac{1}{T_2}}{s_1 + \frac{1}{\beta T_2}} \right|$$

is approximately unity, where $s = s_1$ is one of the dominant closed-loop poles. Determine the values of T_1 and β from the magnitude and angle conditions:

$$\left| K_c \left(\frac{s_1 + \frac{1}{T_1}}{\beta} \right) G(s_1) \right| = 1$$

$$\frac{s_1 + \frac{1}{T_1}}{s_1 + \frac{\beta}{T_1}} = \phi$$

5. Using the value of β just determined, choose T_2 so that

$$\left| \frac{s_1 + \frac{1}{T_2}}{s_1 + \frac{1}{\beta T_2}} \right| \approx 1$$

$$-5^\circ < \frac{s_1 + \frac{1}{T_2}}{s_1 + \frac{1}{\beta T_2}} < 0^\circ$$

The value of βT_2 , the largest time constant of the lag-lead compensator, should not too large to be physically realized.

Example 1: Consider the control system with the following feedforward transfer function:

$$G(s) = \frac{4}{s(s + 0.5)}$$

It is desired to make ξ of the dominant closed-loop poles to 0.5 and $\omega_n = 5$ rad/sec and $K_v = 80 \text{ sec}^{-1}$. Design an appropriate compensator to meet all the performance specifications.

LECT. 15

Solution:

The system has closed-loop poles at

$$s = -0.25 \pm j1.9843$$

And $\xi = 0.125$, $\omega_n = 2$ rad/sec, and $K_v = 8 \text{ sec}^{-1}$.

Let us assume that we use lag-lead having the transfer function:

$$G_c(s) = K_c \left(\frac{s + \frac{1}{T_1}}{s + \frac{\gamma}{T_1}} \right) \left(\frac{s + \frac{1}{T_2}}{s + \frac{1}{\beta T_2}} \right) (\gamma > 1, \beta > 1)$$

Where $\gamma \neq \beta$. Then the compensated system will have the transfer function:

$$G_c(s)G(s) = K_c \left(\frac{s + \frac{1}{T_1}}{s + \frac{\gamma}{T_1}} \right) \left(\frac{s + \frac{1}{T_2}}{s + \frac{1}{\beta T_2}} \right) G(s)$$

From the performance specifications, the dominant closed-loop poles must be at:

$$s = -2.5 \pm j4.33$$

Since

$$\frac{4}{s(s+0.5)} \Big|_{s=-2.50+j4.33} = -235^\circ$$

the lead portion of the lag-lead compensator must contribute 55° so that the root locus passes through the desired location of the dominant closed-loop poles.

To design the lead portion of the compensator, we first determine the location of the zero and pole that will give 55° contribution. There are many possible choices, but we shall here choose the zero at $s = -0.5$ so that this zero will cancel the pole at $s = -0.5$ of the plant. Once the zero is chosen, the pole can be located such that the angle contribution is 55° . By simple calculation or graphical analysis, the pole must be located at $s = -5.021$. Thus, the lead portion of the lag-lead compensator becomes

$$K_c \frac{s + \frac{1}{T_1}}{s + \frac{\gamma}{T_1}} = K_c \frac{s + 0.5}{s + 5.021}$$

LECT. 15

Thus

$$T_1 = 2, \quad \gamma = \frac{5.021}{0.5} = 10.04$$

Next we determine the value of K_c from the magnitude condition:

$$\left| K_c \frac{s + 0.5}{s + 5.021} \frac{4}{s(s + 0.5)} \right|_{s=-2.5+j4.33} = 1$$

Hence,

$$K_c = \left| \frac{(s + 5.021)s}{4} \right|_{s=-2.5+j4.33} = 6.26$$

The lag portion of the compensator can be designed as follows: First the value of β is determined to satisfy the requirement on the static velocity error constant:

$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} s G_c(s) G(s) = \lim_{s \rightarrow 0} s K_c \frac{\beta}{\gamma} G(s) \\ &= \lim_{s \rightarrow 0} s (6.26) \frac{\beta}{10.04} \frac{4}{s(s + 0.5)} = 4.988\beta = 80 \end{aligned}$$

Hence, β is determined as

$$\beta = 16.04$$

Finally, we choose the value of T_2 large enough so that

$$\left| \frac{s + \frac{1}{T_2}}{s + \frac{1}{16.04T_2}} \right|_{s=-2.5+j4.33} \approx 1$$

and

$$-5^\circ < \left| \frac{s + \frac{1}{T_2}}{s + \frac{1}{16.04T_2}} \right|_{s=-2.5+j4.33} < 0^\circ$$

LECT. 15

Since $T_2 \approx 5$ (or any number greater than 5) satisfies the above two requirements, we may choose

$$T_2 = 5$$

Now the transfer function of the designed lag-lead compensator is given by

$$\begin{aligned} G_c(s) &= (6.26) \left(\frac{s + \frac{1}{2}}{s + \frac{10.04}{2}} \right) \left(\frac{s + \frac{1}{5}}{s + \frac{1}{16.04 \times 5}} \right) \\ &= 6.26 \left(\frac{s + 0.5}{s + 5.02} \right) \left(\frac{s + 0.2}{s + 0.01247} \right) \\ &= \frac{10(2s + 1)(5s + 1)}{(0.1992s + 1)(80.19s + 1)} \end{aligned}$$

The compensated system will have the open-loop transfer function

$$G_c(s)G(s) = \frac{25.04(s + 0.2)}{s(s + 5.02)(s + 0.01247)}$$

Example 2: Consider the control system of example (1). Suppose that we use a lag-lead compensator of the form given by equation (2), or

$$G_c(s) = K_c \frac{\left(s + \frac{1}{T_1} \right) \left(s + \frac{1}{T_2} \right)}{\left(s + \frac{\beta}{T_1} \right) \left(s + \frac{1}{\beta T_2} \right)} \quad (\beta > 1)$$

Assuming the specifications are the same as those given in example (1), design a compensator $G_c(s)$

Solution:

The desired locations for the dominant closed-loop poles are at

$$s = -2.5 \pm j 4.33$$

The open-loop transfer of the compensated system is

$$G_c(s)G(s) = K_c \frac{\left(s + \frac{1}{T_1} \right) \left(s + \frac{1}{T_2} \right)}{\left(s + \frac{\beta}{T_1} \right) \left(s + \frac{1}{\beta T_2} \right)} \cdot \frac{4}{s(s + 0.5)}$$

LECT. 15

Since the requirement on the static velocity error constant K_v is 80 sec^{-1} , we have

$$K_v = \lim_{s \rightarrow 0} s G_c(s) G(s) = \lim_{s \rightarrow 0} K_c \frac{4}{0.5} = 8K_c = 80$$

Thus

$$K_c = 10$$

The time constant T_1 and the value of β are determined from

$$\left| \frac{s + \frac{1}{T_1}}{s + \frac{\beta}{T_1}} \right| \bigg|_{s=-2.5+j4.33} \frac{40}{s(s+0.5)} \bigg|_{s=-2.5+j4.33} = \left| \frac{s + \frac{1}{T_1}}{s + \frac{\beta}{T_1}} \right| \frac{8}{4.77} = 1$$

$$\left| \frac{s + \frac{1}{T_1}}{s + \frac{\beta}{T_1}} \right| \bigg|_{s=-2.5+j4.33} = 55^\circ$$

As shown in figure 3,

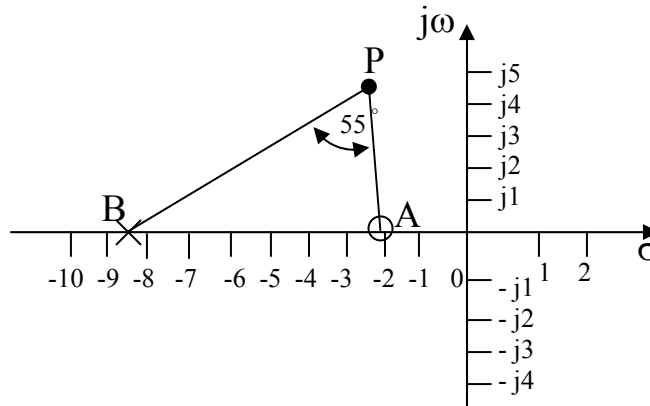


Figure 3: determination of the desired pole-zero location

We can easily locate points A and B such that

$$\angle APB = 55^\circ, \frac{\overline{PA}}{\overline{PB}} = \frac{4.77}{8}$$

By using a graphical approach or trigonometric approach, the result is:

$$\overline{AO} = 2.38, \overline{BO} = 8.34$$

LECT. 15

or
$$T_1 = \frac{1}{2.38} = 0.42, \beta = 8.34T_1 = 3.503$$

The lead portion of the lag-lead network thus becomes

$$10 \frac{s + 2.38}{s + 8.34}$$

For the lag portion, we may choose:

$$T_2 = 10$$

Then

$$\frac{1}{\beta T_2} = \frac{1}{3.503 \times 10} = 0.0285$$

Thus, the lag-lead compensator becomes

$$G_c(s) = (10) \left(\frac{s + 2.38}{s + 8.34} \right) \left(\frac{s + 0.1}{s + 0.0285} \right)$$

The compensated system will have the open-loop transfer function:

$$G_c(s)G(s) = \frac{40(s + 2.38)(s + 0.1)}{s(s + 8.34)(s + 0.0285)(s + 0.5)}$$

End of Lecture fifteen

Computer control design and modeling

Lecture 16

Analog Simulation

Introduction

The electronic analog computer consists of a set of computing components that perform the functions of addition, scaling, integration, etc., required in representing differential equations. The voltages of the analog computer represent the continuous solution of the differential equation programmed by the interconnection of the components (functions). The differential equation is a model of a physical (or natural) dynamic system. An analogy is made between the physical variables of the model and the voltages that represent them on the computer (thus the term “analog” computer). The analogy defines scale factors relating the units of the physical variables to volts in the computer variables. Since the solution of the physical system may be faster or slower than desired, the computer solution can be scaled in both time and magnitude.

The underlying concept of analog computation is to interconnect or wire together those components (summers, scalars, integrators, etc.) required to represent the differential equations of interest. As discussed earlier, these differential equations can be expressed in classical form (a single n th-order differential equation) or state space form (n first-order differential equations), although both representations will result in essentially the same wiring diagram. It is easy to perform a single integration at a time on the analog computer, so the state space format is a more natural representation of dynamic systems for analog computation. Therefore, the setup procedure for analog computation will be developed in the state space format; the classical approach, however, will be briefly illustrated as well.

Computing Components

The basic computing element of the analog computer is a high-gain differential DC amplifier with feedback, called an operational amplifier or “op-amp.” The op-amp has extremely high input impedance, low output impedance, and high gain. The high input impedance results in negligible required input current into the amplifier. The low output impedance results in negligible loading effect upon the output voltage. The high gain G of the amp, accompanied by a sign change or inversion, is required in the computing equation.

LECT. 16

The op-amp (shown in figure 1(a)) open-loop transfer equation (input-output relationship) is

$$e_o = -Ge_a \quad (1)$$

where e_a is the amplifier input voltage, G is the gain, and e_o is the output voltage.

The op-amp is always used with input and feedback impedances, as shown in figure 1(b). Since a common ground line is always present in electronic analog computer circuits, and the op-amp is normally supplied with positive and negative voltage sources, these details can be omitted to reduce clutter in the circuit, as shown by figure 1(c).

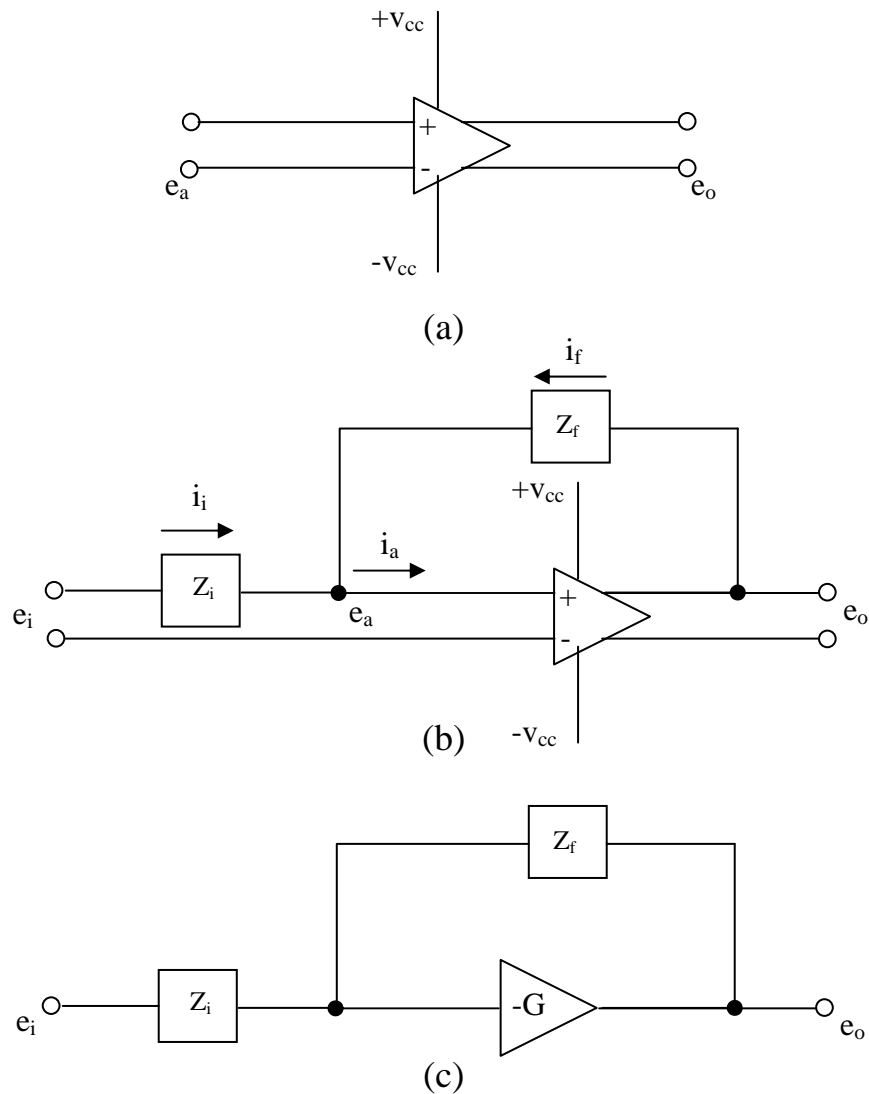


Figure 1: Representations of an operational amplifier circuit.
 (a) Isolated op-amp. (b) Op-amp with input and feedback impedances.
 (c) Simplified schematic of analog computer circuit.

LECT. 16

The circuit of figure 1(b) can be analyzed by the following component equations (neglecting the output impedance of the amplifier):

$$i_i = \frac{e_i - e_a}{Z_i} \quad (2)$$

$$i_f = \frac{e_o - e_a}{Z_f} \quad (3)$$

The node equation at the amplifier may be simplified by taking into account the fact that I_a is negligibly small due to the high input impedance of the op-amp and the low input voltage to the amplifier:

$$i_i + i_f = i_a \approx 0 \quad (4)$$

Substituting the component equations into the node equation yields

$$e_o = -\frac{e_o}{G} \left(1 + \frac{Z_f}{Z_i} \right) - \left(\frac{Z_f}{Z_i} \right) e_i \quad (5)$$

Since the gain G is extremely large compared to $1 + (Z_f / Z_i)$, the first term can be neglected, yielding the basic computing equation

$$e_o = -\frac{Z_f}{Z_i} e_i \quad (6)$$

This equation indicates that the closed-loop gain between input and output is Z_f / Z_i , accompanied by a sign change. Different combinations of the types of circuit elements used for Z_f and Z_i give rise to the various computing components discussed next. These computing components are as given below:

1. Inverter or sign change

If both the feedback and the input impedances are resistive and of equal magnitude, (e.g., $Z_f = R_f$, $Z_i = R_i$, where $R_f = R_i$), then the basic computing equation reduces to a sign change or inverter:

$$e_o = -e_i \quad (7)$$

An inverter is commonly used to change the sign of a signal or variable; it is often used for subtraction in a summing junction.

2. Scaler or Fixed-Gain Amplifier

If both impedances are resistive, but not necessarily of equal magnitude, the basic computing equation reduces to a fixed-gain relation, or scaling, between the input and output:

$$e_o = -\frac{R_f}{R_i} e_i \quad (8)$$

This equation represents multiplication by a constant or simulates a coefficient in the differential equation. Most analog computers utilize fixed, or hard-wired, resistors in multiples of ten for scaling; thus, gains of 0.1, 1.0, 10.0 are possible from typical fixed-gain amplifiers for determining the order of magnitude of coefficients. The setting of precise values for the coefficients is done with potentiometers.

3. Potentiometer

A variable electronic resistive voltage divider, or “pot,” is a passive element that can be used to adjust the exact numerical value of the coefficients of the differential equation. Mathematically,

$$e_o = k e_i \quad \text{where } 0 \leq k \leq 1.0 \quad (9)$$

It is clear that the gain of a pot is less than unity, so that a fixed-gain amplifier with a gain of 10.0 or 100.0 is required in conjunction with a pot to obtain coefficients greater than unity. A bank of several pots is always supplied with analog computers to allow system coefficients to be adjusted.

4. Summing Junction

Using more than one input resistor allows the addition, or summation, of several independent input signals. The computing equation reduces to a linear summation of the signals, with the possibility of different gains for each signal:

$$e_o = -\frac{R_f}{R_1} e_1 - \frac{R_f}{R_2} e_2 - \frac{R_f}{R_3} e_3 \quad (10)$$

Summing junctions are ordinarily used to sum together the various terms required to form the differential equation.

5. Integrator

The use of a capacitive feedback impedance with a resistive input impedance results in the integration function. The transfer gain can be stated, in operator notation, with $D = d / dt$, as

$$e_o = -\frac{1}{R_i C_f D} e_i \quad (11)$$

LECT. 16

Writing this equation in integral form and considering the initial condition $e_o(0)$ yields

$$e_o(t) = - \left[\frac{1}{R_i C_f} \int_0^t e_i dt + e_o(0) \right] \quad (12)$$

The integration gain is $1 / (R_i C_f)$. The sign change across the amplifier requires that the initial condition be entered with a sign change. It is clear that by using multiple input resistors, a summing integrator is possible.

6. Differentiator

If a resistive feedback impedance and a capacitive input impedance are used, the resulting equation implies the function of differentiation. Specifically

$$e_o = -R_f C_i D e_i \quad (13)$$

Or

$$e_o = -R_f C_i \frac{de_i}{dt} \quad (14)$$

Differentiators amplify high-frequency signals more than low-frequency signals and hence produce more noise on the output. For this reason, differentiators are seldom used in analog computation.

Table 1 illustrates the electronic circuit and the simplified analog computer block diagram symbol for each of the common linear operations just discussed.

End of Lecture sixteen

Table 1 Basic analog computer operations.

Operation	Computer symbol	Computer circuit
Amplification (with sign change) $e_0 = -Ge_1$		$G = \frac{R_f}{R_1}$
Summation (only two inputs shown) $e_0 = -G_1e_1 - G_2e_2$		$G_1 = \frac{R_f}{R_1} \quad G_2 = \frac{R_f}{R_2}$
Integration (shown with summation) $e_0 = - \int (G_1e_1 + G_2e_2) dt - I.C.$		$G_1 = \frac{1}{R_1 C} \quad G_2 = \frac{1}{R_2 C}$
Multiplication (by a constant < 1.0) $e_0 = ke_1$ $0 < k < 1.0$		$k = \frac{R_0}{R_T}$
Multiplication (by a constant > 1.0) $e_0 = -10ke_1$		$k = \frac{R_0}{R_T} \quad 10 = \frac{R_f}{R_1}$

Computer control design and modeling

Lecture 17

Classical analog computer diagrams

The goal of classical analog computation is to solve the classical differential equation for the highest-order derivative of an nth-order equation and integrate n times to yield the solution. The equation that results for the nth derivative is formed at the input of the first integrator by summing the lower order derivatives with appropriate coefficients (gains).

For example, consider the second-order differential equation

$$\ddot{e} + 0.5\dot{e} + 2.0e = v(t) \quad (1)$$

with initial conditions

$$e(0) = e_{\text{int}} \quad (2)$$

and

$$\dot{e}(0) = \dot{e}_{\text{int}} \quad (3)$$

Solving for the highest order derivative yields

$$\ddot{e} = v(t) - 0.5\dot{e} - 2.0e \quad (4)$$

Assuming that this equation is formed at the input of an integrator, the output would be $-\dot{e}$, and the initial condition $\dot{e}(0)$ would have the opposite sign of $-\dot{e}$ as shown in Figure (1). An integration of the $-\dot{e}$ signal yields e . The initial condition of e would be entered with the opposite sign of $+e$.

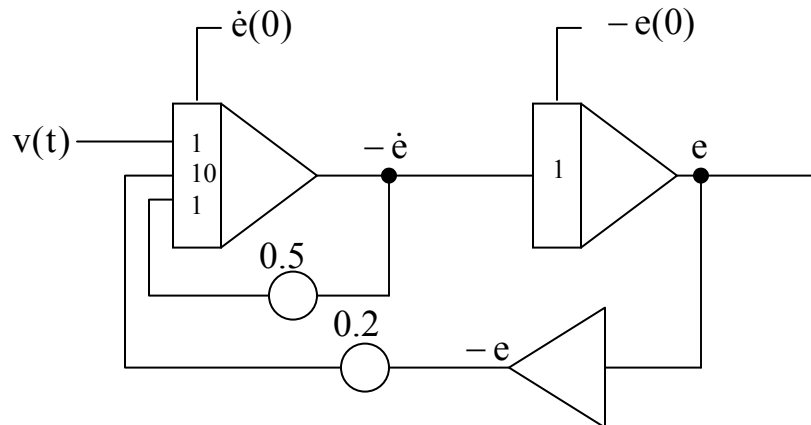


Figure 1: Classical analog computer diagram

Now the equation for \ddot{e} must be wired at the input to first integrator. Scaling $-\dot{e}$ with a pot set at 0.5 yields one term. The e signal must be inverted, scaled by 0.2, and then entered into the summing integrator with a gain of 10

to obtain the proper sign and coefficient of the term. The system input $v(t)$ must be available from a function generator or some similar source, depending upon the input signal. Summation of these three terms with the gains shown yields equation (4) for \ddot{e} .

State space Computer Diagrams

As mentioned earlier, the analog computer is designed to conveniently perform single integrations at a time; thus, having an n th-order dynamic system model of n first-order equations in state space format is advantageous, since single integrations of the derivative state vector $\dot{\mathbf{x}}$ yields the solution of the system. The equations for the derivatives of each of the state variables are formed or wired at the input of the integrators. This format is further attractive, since the coefficients in the \mathbf{A} and \mathbf{B} matrices will result in integrator gains when scaled in magnitude and time.

Consider a state space system (with a single input) that has not magnitude scaled, namely,

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (5)$$

or

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & & & & \\ \cdot & & & & \\ a_{n1} & a_{n2} & \cdot & \cdot & a_{nn} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{bmatrix} [u_1] \quad (6)$$

with

$$\mathbf{x}(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \\ \cdot \\ \cdot \\ x_n(0) \end{bmatrix} \quad (7)$$

This system simulated by the analog computer diagram shown in figure 2.

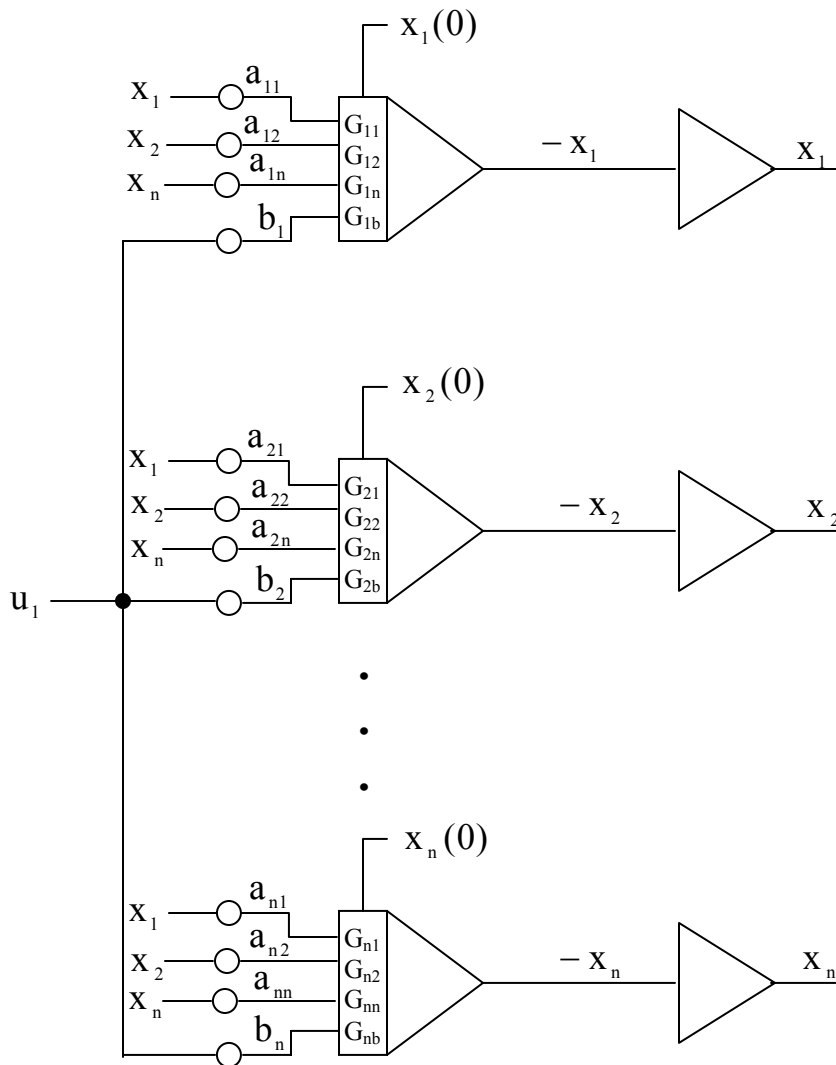


Figure 2: General state space analog computer diagram

Now, the input to each integrator forms the equation of the derivative of a state, and the output will be that state variable with a sign change; an inverter often (but not always) will be required for computation or monitoring. It is clear that the initial condition of an integrator must be entered with a sign opposite to that of the output, which in this case allows the initial conditions to be entered with the same sign as that of the physical problem.

For example, consider the second-order system of equation (1), which is used to illustrate the classical analog computer diagram. This system can be converted to state space form, yielding

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [u_1] \quad (8)$$

with

$$\mathbf{x}(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \quad (9)$$

The latter system is programmed on the analog computer as shown in figure 3. It is clear that the solution for x_1 is the output of the upper inverter, and the solution for x_2 is the output of the lower inverter. The equation for \dot{x}_2 contains $-2x_1$, that was obtained with a pot set at 0.2 and an integrator gain of 10.

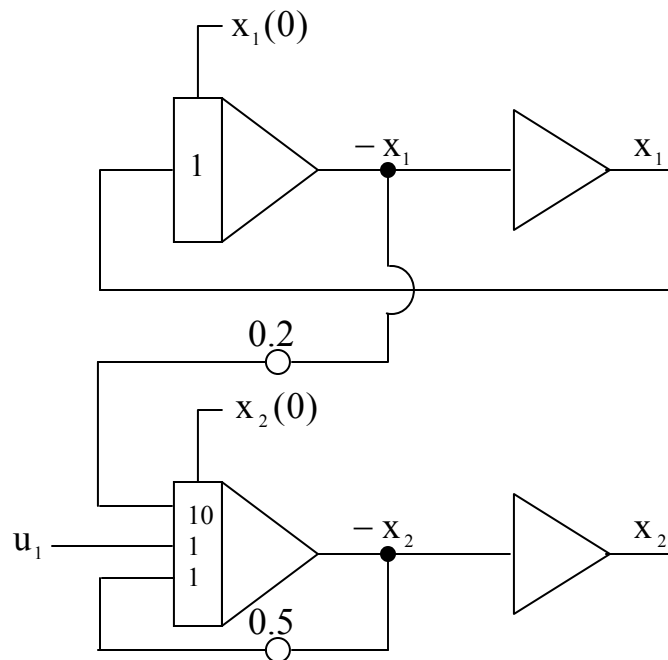


Figure 3: State space analog computer diagram for a second-order system

Magnitude and time scaling

The purpose of analog computation is to develop an analogy between the physical system variables and the voltages that represent those variables on the analog computer; thus, a conversion must be made from physical variables to the analog computer variables. The analogy provides these conversion factors, or magnitude scale factors. A magnitude scale factor must be associated with each state variable. It is clear that this scaling must

be done in either the classical or the state space format. It is easily handled in the state space format with scaling matrices.

A. Magnitude scaling

Consider the set of n th-order linear state space equations with m inputs, denoted by

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (10)$$

With initial conditions

$$\mathbf{x}(0) = \mathbf{x}_{\text{int}} \quad (11)$$

where \mathbf{x} and \mathbf{u} are the variables and input vectors representing the physical system. Any system response or output y may be found by

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \quad (12)$$

The physical system state variable \mathbf{x} can be scaled to analog computer state variable \mathbf{e} by the transformation

$$\mathbf{x} = \boldsymbol{\alpha} \mathbf{e} \quad (13)$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \alpha_2 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \alpha_3 & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \alpha_n \end{bmatrix} \quad (14)$$

is a constant n th-order diagonal scaling matrix (in physical units/volt).

In a similar fashion, the inputs \mathbf{u} to the physical system can be scaled to inputs \mathbf{v} to the analog computer system by the transformation

$$\mathbf{u} = \boldsymbol{\beta} \mathbf{v} \quad (15)$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \beta_2 & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ 0 & 0 & \cdot & \cdot & \cdot & \beta_m \end{bmatrix} \quad (16)$$

LECT. 17

is a constant m th-order diagonal scaling matrix (usually $m = 1$).

Thus, observing that $\dot{\mathbf{x}} = \boldsymbol{\alpha}\dot{\mathbf{e}}$ and substituting the preceding expressions for \mathbf{x} and \mathbf{u} into equation (10), we obtain, after a bit of algebraic rearrangement,

$$\dot{\mathbf{e}} = \boldsymbol{\alpha}^{-1}\mathbf{A}\boldsymbol{\alpha}\mathbf{e} + \boldsymbol{\alpha}^{-1}\mathbf{B}\boldsymbol{\beta}\mathbf{v} \quad (17)$$

or

$$\dot{\mathbf{e}} = \tilde{\mathbf{A}}\mathbf{e} + \tilde{\mathbf{B}}\mathbf{v} \quad (18)$$

with initial conditions

$$\mathbf{e}(0) = \boldsymbol{\alpha}^{-1}\mathbf{x}(0) \quad (19)$$

and outputs

$$\mathbf{y} = \mathbf{C}\boldsymbol{\alpha}\mathbf{e} + \mathbf{D}\boldsymbol{\beta}\mathbf{v} \quad (20)$$

or

$$\mathbf{y} = \tilde{\mathbf{C}}\mathbf{e} + \tilde{\mathbf{D}}\mathbf{v} \quad (21)$$

Now it is clear that the inverse of a diagonal matrix is merely a diagonal matrix with the scalar inverse of the respective elements; that is,

$$\boldsymbol{\alpha}^{-1} = \begin{bmatrix} \frac{1}{\alpha_1} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\alpha_2} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\alpha_3} & \dots & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & 0 & \dots & \frac{1}{\alpha_n} \end{bmatrix} \quad (22)$$

Thus, matrices become

$$\tilde{\mathbf{A}} = \boldsymbol{\alpha}^{-1}\mathbf{A}\boldsymbol{\alpha} = \begin{bmatrix} \frac{\alpha_1}{\alpha_1} a_{11} & \frac{\alpha_2}{\alpha_1} a_{12} & \frac{\alpha_3}{\alpha_1} a_{13} & \dots & \frac{\alpha_n}{\alpha_1} a_{1n} \\ \frac{\alpha_1}{\alpha_2} a_{21} & \frac{\alpha_2}{\alpha_2} a_{22} & \frac{\alpha_3}{\alpha_2} a_{23} & \dots & \frac{\alpha_n}{\alpha_2} a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\alpha_1}{\alpha_n} a_{n1} & \frac{\alpha_2}{\alpha_n} a_{n2} & \frac{\alpha_3}{\alpha_n} a_{n3} & \dots & \frac{\alpha_n}{\alpha_n} a_{nn} \end{bmatrix} \quad (23)$$

in which the diagonal elements are unchanged, and

$$\tilde{\mathbf{B}} = \alpha^{-1} \mathbf{B} \beta = \begin{bmatrix} \frac{\beta_1}{\alpha_1} b_{11} & \frac{\beta_2}{\alpha_1} b_{12} & \frac{\beta_3}{\alpha_1} b_{13} & \dots & \frac{\beta_m}{\alpha_1} b_{1m} \\ \frac{\beta_1}{\alpha_2} b_{21} & \frac{\beta_2}{\alpha_2} b_{22} & \frac{\beta_3}{\alpha_2} b_{23} & \dots & \frac{\beta_m}{\alpha_2} b_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\beta_1}{\alpha_n} b_{n1} & \frac{\beta_2}{\alpha_n} b_{n2} & \frac{\beta_3}{\alpha_n} b_{n3} & \dots & \frac{\beta_m}{\alpha_n} b_{nm} \end{bmatrix} \quad (24)$$

or, in general, the element in the i th row and j th column becomes

$$\tilde{b}_{ij} = \frac{\beta_j}{\alpha_i} b_{ij} \quad (25)$$

And similarly for \tilde{a}_{ij} .

For k outputs,

$$\tilde{\mathbf{C}} = \mathbf{C} \alpha = \begin{bmatrix} c_{11} \alpha_1 & c_{12} \alpha_2 & \dots & c_{1n} \alpha_n \\ c_{21} \alpha_1 & c_{22} \alpha_2 & \dots & c_{2n} \alpha_n \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ c_{k1} \alpha_1 & c_{k2} \alpha_2 & \dots & c_{kn} \alpha_n \end{bmatrix} \quad (26)$$

and

$$\tilde{\mathbf{D}} = \mathbf{D} \beta = \begin{bmatrix} d_{11} \beta_1 & d_{12} \beta_2 & \dots & d_{1m} \beta_m \\ d_{21} \beta_1 & d_{22} \beta_2 & \dots & d_{2m} \beta_m \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ d_{k1} \beta_1 & d_{k2} \beta_2 & \dots & d_{km} \beta_m \end{bmatrix} \quad (27)$$

B. Time scaling

Often, the response of the physical system model proceeds at a slower or faster pace than is practical for recording or plotting purposes. That is, the response of the physical system may be complete in milliseconds or may take hours to arrive at steady state. For this reason, it is frequently desirable

LECT. 17

to slow down or speed up the simulated response on the computer. This can conveniently be done by making the transformation

$$t = \gamma \tau \quad (28)$$

where γ is a time scaling factor, t is the real time of the physical system, and τ is computer time ($\gamma > 1$ accelerates the solution of the system, while $\gamma < 1$ retards the solution). This transformation implies that

$$dt = \gamma d\tau \quad (29)$$

then

$$\frac{d}{d\tau} = \gamma \frac{d}{dt} \quad (30)$$

Thus, the analog computer state variable system becomes

$$\frac{d}{d\tau} \mathbf{e} = \gamma \tilde{\mathbf{A}}\mathbf{e} + \gamma \tilde{\mathbf{B}}\mathbf{v} \quad (31)$$

with initial conditions

$$\mathbf{e}(0) = \boldsymbol{\alpha}^{-1}\mathbf{x}(0) \quad (32)$$

and outputs

$$\mathbf{y} = \tilde{\mathbf{C}}\mathbf{e} + \tilde{\mathbf{D}}\mathbf{v} \quad (33)$$

It is clear from the forgoing description of the system that time scaling affects only the $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ matrices and thus the integrator gains; it does not affect initial conditions or outputs. Time scaling affects only the time required to obtain a response (you will know if time scaling is necessary if all of the integrator gains are very large (> 100) or very small (< 0.01)).

Selection of Magnitude and Time Scale Factors

The most difficult task in analog simulation is selecting the magnitude scale factors for each state variable. The scale factors α establish the analogy between the physical variables and the analog computer variables. They are selected such that a maximum response in the physical system will result in a maximum response in the computer variables that will be within certain limits.

A. Selection of Magnitude Scale Factors

The magnitude scale factors are selected in such a manner that the maximum (absolute) value of the physical response of a state variable x_i will cause a maximum response of the computer variable e_i to be about (but not greater than) 10 volts. Thus,

$$\alpha_i = \left| \frac{x_{i\max}}{e_{i\max}} \right| \quad (34)$$

If the computed value of α is not a convenient number, it should be rounded off higher (rather than lower) to prevent overloading ($e_i > \pm 10$ volts). For example, if $x_{i\max}$ is predicted to be 1.37 cm, then we would select

$$\alpha_i = \frac{1.37\text{cm}}{10\text{volt}} = 0.137 \frac{\text{cm}}{\text{volt}} \quad (35)$$

which would be rounded off to 0.15 or 0.20 cm/volt. The input scale factors are selected in a similar manner as

$$\beta_j = \frac{|u_{j\max}|}{|v_{j\max}|} \quad (36)$$

The preceding discussion specifies how to select the scale factors, but the problem of predicting $x_{i\max}$ and $u_{j\max}$ still faces us. Determining $u_{j\max}$ is relatively easy, since the system input is usually known (e.g., a step, a ramp, sinusoidal, etc). While the determination of $x_{i\max}$ is slightly more difficult, it is greatly simplified by the approach set out next.

In some systems, the largest value of a state variable occurs at its initial condition, and the response decays thereafter. In other instances, the maximum response occurs at or near the steady-state value, as shown in figure 4

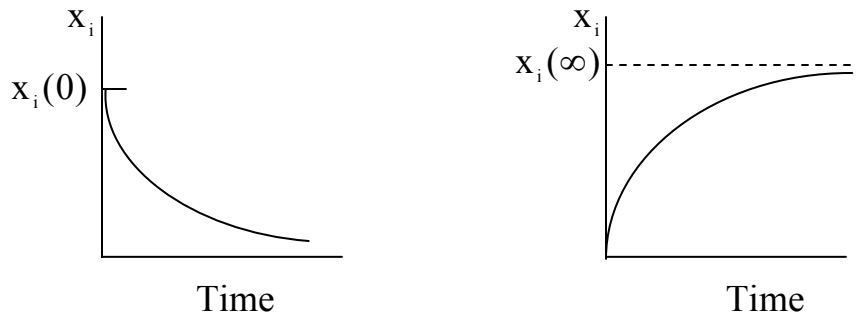


Figure 4: Typical responses illustrating maximum values

LECT. 17

The initial conditions are known, and the steady-state maxima can be estimated. To do the estimation, we observe that the steady-state values occur when $\dot{\mathbf{x}} = \mathbf{0}$, and thus,

$$\mathbf{Ax}(\infty) = -\mathbf{Bu} \quad (37)$$

This yields a set of algebraic equations in which the energy or effort supplied by the inputs is put into the state variables. If we solve each of the equations for successive state variables, then assuming that all of the input effort goes into these variables entirely, we can observe the ultimate steady-state maxima of the state variables. For example, consider a second-order system with one input; Equation (37) becomes

$$a_{11}x_1(\infty) + a_{12}x_2(\infty) = -b_1u_1 \quad (38)$$

$$a_{21}x_1(\infty) + a_{22}x_2(\infty) = -b_2u_1 \quad (39)$$

Now, assuming that all of the input effort goes into x_1 and none goes into x_2 we obtain the ultimate (indicated by *) steady-state maxima. From equation (38) with $x_2(\infty)$ assumed to be zero, it follows that

$$x_1^*(\infty) = \frac{-b_1}{a_{11}}u_{1\max} \quad (40)$$

From equation (39)

$$x_1^*(\infty) = \frac{-b_2}{a_{21}}u_{1\max} \quad (41)$$

Whichever of these two equations yields the largest absolute value for x_1 will be used as the steady-state maximum. Thus,

$$x_1^*(\infty)_{\max} = \text{MAX of } \left\{ \left| \frac{b_1}{a_{11}}u_{1\max} \right| \text{ or } \left| \frac{b_2}{a_{21}}u_{1\max} \right| \right\} \quad (42)$$

And similarly,

$$x_2^*(\infty)_{\max} = \text{MAX of } \left\{ \left| \frac{b_1}{a_{12}}u_{1\max} \right| \text{ or } \left| \frac{b_2}{a_{22}}u_{1\max} \right| \right\} \quad (43)$$

We are now ready to estimate the maximum value we expect from a state variable. The maximum will occur either at the initial condition or at the value estimated by the steady-state maximum method; that is

$$x_{i\max} = \text{MAX of } \left\{ |x_i(0)| \text{ or } |x_i^*(\infty)_{\max}| \right\} \quad (44)$$

The method just outlined should predict all the maximum expected values of the state variables necessary to allow the selection of magnitude scale factors α and β . Then, having selected the scale factors, we can compute the

matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$. The elements in those matrices represent the integrator gains to be used in wiring the computer diagram and thus have units of 1/second or 1/time. It is desirable for these gains to be in the range 0.1 to 10 (or possibly 100). If all of the coefficients or gains are within this range, the solution of the system can be obtained directly. If all of the gains are in this range except for one or so, trade-offs can be made by raising one scale factor. It is clear from equation (25) that increasing a scale factor will decrease the gains in one row and raise the gains in one column. Since the \mathbf{A} matrix usually has numerous zero entries, the trade-off of lowering one gain while raising its symmetrical element can be effective.

B. Selection of Time Scale Factors

If all of the gains are either too low or too high, time scaling will be necessary. The time scale factor γ alters all gains by a common amount in the $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ matrices. The time scale factor is usually selected in multiples of 10.

For example, if the matrix

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 250 \\ 100 & 75 \end{bmatrix} \quad (45)$$

were obtained, a time scale factor of 1/100 would reduce the coefficients to a usable value. After time scaling:

$$\gamma\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 2.5 \\ 1.0 & 0.75 \end{bmatrix} \quad (46)$$

Example 1: Fluid RLC System

As an example of the scaling process, consider the series RLC fluid system with definitions of state variables

$u_1 =$ Pressure input (kPa)

$x_1 =$ Pressure in capacitor (kPa)

$x_2 =$ Flow in inductor (cm^3/s)

state-space differential equations

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \frac{1}{\text{s}} & 5 \frac{\text{kPa}}{\text{cm}^3} \\ -10 \frac{\text{cm}^3}{\text{kPa s}^2} & -2.5 \frac{1}{\text{s}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \frac{1}{\text{s}} \\ 10 \frac{\text{cm}^3}{\text{kPa s}^2} \end{bmatrix} [u_1] \quad (47)$$

LECT. 17

and initial conditions

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} -20\text{kPa} \\ 0\text{cm}^3/\text{s} \end{bmatrix} \quad (48)$$

Let the system be subjected to a step input

$$u_1 = 10 \text{ kPa} \quad \text{for } t \geq 0 \quad (49)$$

The first step in solving the system is to select the magnitude scale factors. This requires knowledge of $x_{i\max}$. The steady-state maximums can be predicted from equation (47) by setting $\dot{\mathbf{x}} = 0$. The first equation yields no information, since the input has no direct influence on \dot{x}_1 . The second equation results in

$$10x_1(\infty) + 2.5x_2(\infty) = 10u_1 \quad (50)$$

By setting $x_2(\infty) = 0$ and allowing all of the input effort to drive x_1 , we find that

$$x_1^*(\infty) = \frac{10}{10} u_{1\max} = u_{1\max} \quad (51)$$

$$x_1^*(\infty) = 10\text{kPa} \quad (52)$$

By setting $x_1(\infty) = 0$ and allowing all of the input to go into the second state variable, we find that

$$x_2^*(\infty) = \frac{10}{2.5} u_{1\max} = 4u_{1\max} \quad (53)$$

$$x_2^*(\infty) = 40\text{cm}^3/\text{s} \quad (54)$$

The maximum values for \mathbf{x} can now be estimated from the maximum of either the initial conditions or the steady-state maxima:

$$\begin{aligned} x_{1\max} &= \text{MAX of } \{ |x_1(0)| \text{ or } |x_1^*(\infty)_{\max}| \} \\ &= \text{MAX of } \{ 20 \text{ or } 10 \} = 20\text{kPa} \end{aligned} \quad (55)$$

$$\begin{aligned} x_{2\max} &= \text{MAX of } \{ |x_2(0)| \text{ or } |x_2^*(\infty)_{\max}| \} \\ &= \text{MAX of } \{ 0 \text{ or } 40 \} = 40\text{cm}^3/\text{s} \end{aligned} \quad (56)$$

The magnitude scale factors can now be computed:

$$\alpha_1 = \frac{x_{1\max}}{e_{1\max}} = \frac{20\text{kPa}}{10\text{volt}} = 2.0 \frac{\text{kPa}}{\text{volt}} \quad (57)$$

$$\alpha_2 = \frac{x_{2\max}}{e_{2\max}} = \frac{40\text{cm}^3/\text{s}}{10\text{volt}} = 4.0 \frac{\text{cm}^3/\text{s}}{\text{volt}} \quad (58)$$

$$\beta_1 = \frac{u_{1\max}}{v_{1\max}} = \frac{10\text{kPa}}{10\text{volt}} = 1.0 \frac{\text{kPa}}{\text{volt}} \quad (59)$$

Next, we can compute the $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ matrices:

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_{11} & \frac{\alpha_2}{\alpha_1} a_{12} \\ \frac{\alpha_1}{\alpha_2} a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ -5 & -2.5 \end{bmatrix} \quad (60)$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} \frac{\beta_1}{\alpha_1} b_1 \\ \frac{\beta_1}{\alpha_2} b_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2.5 \end{bmatrix} \quad (61)$$

The initial conditions of \mathbf{e} are computed from the following equation:

$$\mathbf{e}(0) = \boldsymbol{\alpha}^{-1} \mathbf{x}(0) \quad (62)$$

or

$$e_1(0) = \frac{1}{\alpha_1} x_1(0) = \frac{\text{volt}}{2\text{kPa}} (-20\text{kPa}) = -10\text{volt} \quad (63)$$

$$e_2(0) = \frac{1}{\alpha_2} x_2(0) = \frac{\text{volt}}{4\text{cm}^3/\text{s}} (0\text{cm}^3/\text{s}) = 0\text{volt} \quad (64)$$

The final simulation problem to be wired on the analog computer can be stated as

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ -5 & -2.5 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2.5 \end{bmatrix} [v_1] \quad (65)$$

with initial conditions

$$\begin{bmatrix} e_1(0) \\ e_2(0) \end{bmatrix} = \begin{bmatrix} -10\text{volt} \\ 0 \end{bmatrix} \quad (66)$$

The system input is a step input, or a constant for $t \geq 0$. The magnitude of this constant is

$$v_1 = \frac{u_1}{\beta_1} = 10\text{volt} \quad (67)$$

LECT. 17

The elements in $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ represent integrator gains. The analog computer diagram for this system is shown in Figure 5

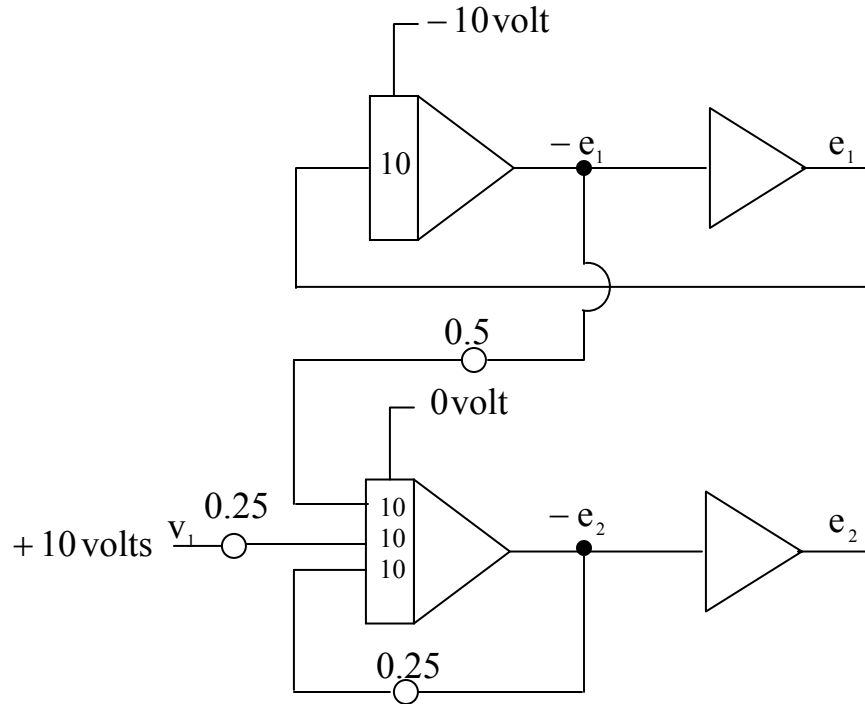


Figure 5: Analog computer diagram for Example 1

End of Lecture seventeen

Computer control design and modeling

Lecture 18

Digital Simulation

Introduction

While many low-order linear systems can be solved using standard differential equation solution procedures, numerical methods are advantageous when a large number of differential equations is necessary to describe the system or when the physical situation must be represented by nonlinear differential equations. In these cases, digital simulation is an indispensable tool. It is assumed that higher order systems can be transformed into an equivalent set of first-order, state space equations, and only first order equations are discussed here.

Figure 1 shows the solution to a differential equation (solid line) and an approximate solution to the same equation obtained by a numerical method (dots). While a closed-form solution consists of an analytic function relating the dependent and independent variables (x and t , respectively), a numerical solution consists of a series of points x_j obtained at a discrete set of points t_j of time. The distance between the time points (the time step) is denoted by h . The figure makes clear the strong dependence of the quality of the numerical solution on the size of h . That is, toward the end of the record of time given, the analytic solution begins an oscillation that the discrete point spacing is unable to represent. A shorter time step is obviously needed in this region of the solution.

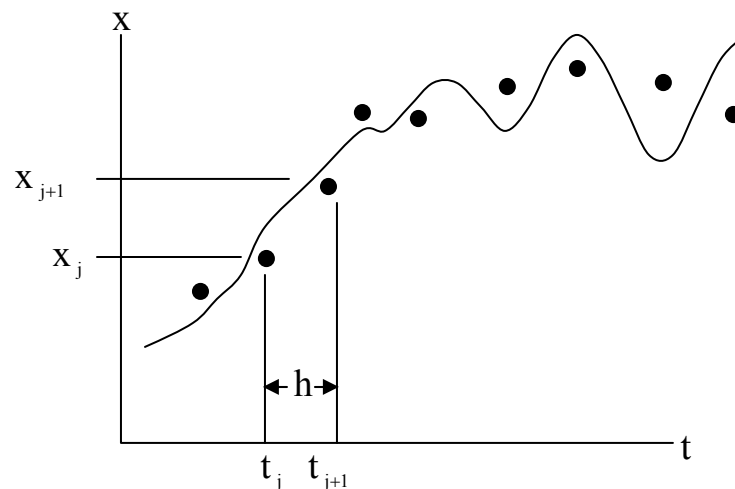


Figure 1: Numerical integration of a dynamic system

Euler's method

Suppose a single first-order differential equation is given in which $f(x,t)$ may be a linear or a nonlinear function, depending on the problem at hand:

$$\frac{dx}{dt} = f(x,t) \text{ with initial condition } x(t_0) = x_0 \quad (1)$$

This is an initial-value problem, since $x(t)$ is given at some initial time t_0 , and we desire the solution values at succeeding times t . Equation (1) may be written as

$$\int_{x_j}^{x_{j+1}} dx = \int_{t_j}^{t_{j+1}} f(x,t) dt \quad (2)$$

Suppose the time step $h = t_{j+1} - t_j$ is small enough so that the slope $\dot{x} = f(x,t)$ may be taken to be the constant $f(x_j, t_j)$ over the interval h . Then

$$x_{j+1} - x_j = f(x_j, t_j) \int_{t_j}^{t_{j+1}} dt = hf(x_j, t_j) = hf_j \quad (3)$$

Thus, the new value of x , (x_{j+1}) may be computed from previous values by the relation

$$x_{j+1} = x_j + hf_j \quad (4)$$

This simple technique of integration is called **Euler's method**.

Geometrically, the Euler process assumes that the slope $dx/dt = f(x,t)$ is constant over the interval h . This assumption has obvious limitations. The error involved in using Euler's method can be examined by considering a Taylor series expansion of $x(t)$. Recalling the Taylor series expansion discussed in calculus, we write

$$x(t_{j+1}) = x(t_j) + h \left. \frac{dx}{dt} \right|_j + \frac{h^2}{2!} \left. \frac{d^2x}{dt^2} \right|_j + \dots \quad (5)$$

Replacing dx/dt with $f(x,t)$ we obtain

$$x_{j+1} = x_j + hf_j + \frac{h^2}{2!} \left. \frac{d^2x}{dt^2} \right|_j + \dots \quad (6)$$

In Euler's method, we neglect terms beyond hf_j ; hence, the error caused at each step is proportional to h^2 , and is called the truncation error.

Example 1: Euler's Method

Use Euler's method to solve the following first order equations. The first displays exponential behavior; the second has a harmonic solution.

LECT. 18

$$\dot{x} = -x \text{ with initial condition } x(0) = 1, 0 \leq t \leq 5 \text{ s,} \tag{7}$$

And

$$\dot{y} = -3y + 10\sin t, \text{ with initial condition } y(0) = -1, 0 \leq t \leq 12 \text{ s.} \tag{8}$$

Solution:

For Equation (7) we select a step size of $h = 0.5$ s then

$$\text{At } t = 0.5: \text{ then } x_1 = x_0 + 0.5(-1) = 1 - 0.5 = 0.5 \tag{9}$$

$$\text{At } t = 1: \text{ then } x_2 = x_1 + 0.5(-0.5) = 0.5 - 0.25 = 0.25 \tag{10}$$

And etc.

This recursive process is continued until we reach the desired final time point, $t = 5$. The exact solution to Equation (7) can be found using classical methods as:

$$x_e(t) = e^{-t} \tag{11}$$

It is clear that the exact solution approaches zero with increasing time.

Table 1 gives a comparison of the Euler solution with the exact solution to this problem for step sizes, $h = 0.5$, $h = 0.1$, and $h = 0.0125$. The error in the calculated result at some specific time t is called the global error. It is clear from Table 2 that as the step size is halved; the global error of Euler's method is approximately halved also. Thus, even though the pre-step (truncation) error is proportional to h^2 , the global error is approximately proportional to h .

HOMEWORK: Solve the second differential equation.

TABLE 1 SOLUTIONS TO $\dot{x} = -x, x(0) = 1$

h = 0.5			
t	x (Euler)	x (exact)	error (%)
0.5	0.500000	0.606531	17.56
1.0	0.250000	0.367879	32.04
1.5	0.125000	0.223130	43.98
2.0	0.062500	0.135335	53.82
2.5	0.031250	0.082085	61.93
3.0	0.015625	0.049787	68.62
3.5	0.007813	0.030197	74.13
4.0	0.003906	0.018316	78.67
4.5	0.001953	0.011109	82.42
5.0	0.000977	0.006738	85.51

LECT. 18

h = 0.1			
t	x (Euler)	x (exact)	error (%)
0.5	0.590490	0.606531	2.64
1.0	0.348678	0.367879	5.22
1.5	0.205891	0.223130	7.73
2.0	0.121577	0.135335	10.17
2.5	0.071790	0.082085	12.54
3.0	0.042391	0.049787	14.86
3.5	0.025032	0.030197	17.11
4.0	0.014781	0.018316	19.30
4.5	0.008728	0.011109	21.43
5.0	0.005154	0.006738	23.51

h = 0.0125			
t	x (Euler)	x (exact)	error (%)
0.5	0.604622	0.606531	0.31
1.0	0.365568	0.367880	0.63
1.5	0.221031	0.223130	0.94
2.0	0.133640	0.135335	1.25
2.5	0.080802	0.082085	1.56
3.0	0.048855	0.049787	1.87
3.5	0.029539	0.030197	2.18
4.0	0.017860	0.018315	2.49
4.5	0.010798	0.011109	2.80
5.0	0.006529	0.006738	3.10

TABLE 2 FINAL GLOBAL ERRORS USING EULER’S METHOD TO SOLVE $\dot{x} = -x, x(0) = 1$

Step size h(s)	0.5	0.25	0.1	0.05	0.025	0.0125
Global error in x(t=2), (percent)	53.8	26.0	10.2	5.04	2.51	1.25

More Accurate Methods

Euler’s method uses only two items of information to propagate the solution from t_j to t_{j+1} : the current solution estimate x_j and the function value or slope f_j . If more information concerning the character of the solution is used, the numerical results will obviously be more accurate for a given step size; that is, they will have smaller per-step truncation errors. Many such methods have been devised over the years. Two are discussed here: Adams method and Runge-Kutta methods.

The Adams method is typical of methods employing three pieces of information to advance the solution. It uses the equation

$$x_{j+1} = x_j + \frac{h}{2}(3f_j - f_{j-1}) \tag{12}$$

which employs a weighted combination of the slopes at time t_j and time t_{j-1} , to propagate the solution. With the Adams method, the per-step error term is proportional to h^3 , and hence, the global error is proportional to h^2 . It is clear that the method is not a **self-starting** process, since solutions must be known at t_j and at t_{j-1} before the values at t_{j+1} can be determined. When starting the process, the initial value is used for the t_0 point but a self-starting technique such as Euler method would have to be used to generate an estimate of the solution at t_1 before the Adams method could be initiated.

Runge-Kutta methods

A technique that employs five information values and is also self-starting is Runge-Kutta scheme. The Runge-Kutta approach is based on retaining higher order terms in the Taylor series expansion of the dependent variable. The fourth-order method, utilizes information about the derivative at four points within the time-step interval. This provides a much more accurate estimate of the solution than the methods discussed previously, since curvature of the solution over the time step is now accounted for. Runge-Kutta methods are widely used in system simulation studies because of their accuracy and ease of implementation.

A frequently used fourth-order Runge-Kutta recursion process is defined by the equation

$$x_{j+1} = x_j + \frac{h}{6}(A + 2B + 2C + D) \quad (13)$$

where

$$A = f(x_j, t_j) \quad (14)$$

$$B = f\left(x_j + \frac{h}{2}A, t_j + \frac{h}{2}\right) \quad (15)$$

$$C = f\left(x_j + \frac{h}{2}B, t_j + \frac{h}{2}\right) \quad (16)$$

$$D = f(x_j + hC, t_j + h) \quad (17)$$

It is clear that the arguments of the functions defined in B, C, and D are not necessarily points on the solution curve $x(t)$. The per-step and global errors of this method are proportional to h^5 and h^4 , respectively.

Example 2: Runge-Kutta Method

Solve the first-order problem of equation (7) using the fourth-order Runge-Kutta method.

Solution: The problem is solved using the Runge-Kutta method as just outlined, and the results are presented in Table 3. It is clear that for comparable step sizes, the Runge-Kutta method produces results that are much more accurate than does the Euler method.

**TABLE 3 RUNGE-KUTTA (R-K)
SOLUTION TO $\dot{x} = -x, x(0) = 1$**

h = 0.5			
t	x (R-K)	x (exact)	error (%)
0.5	0.606771	0.606531	-0.040
1.0	0.368171	0.367879	-0.079
1.5	0.223395	0.223130	-0.119
2.0	0.135550	0.135335	-0.158
2.5	0.082248	0.082085	-0.198
3.0	0.049905	0.049787	-0.238
3.5	0.030281	0.030197	-0.278
4.0	0.018374	0.018316	-0.317
4.5	0.011149	0.011109	-0.357
5.0	0.006765	0.006738	-0.397

This problem is solved several times using the Runge-Kutta method with different step sizes, and the effect of the step size on the global error is shown in Table 4. It is clear that as the step size is halved; the error is reduced by a factor of about 16.

TABLE 4 RUNGE-KUTTA SOLUTION TO $\dot{x} = -x, x(0) = 1$

Step size h(s)	0.5	0.25	0.125
Global error in $x(t=2)$, (percent)	0.158	0.008	0.00046

End of Lecture eighteen

Computer control design and modeling

Lecture 19

Systems of Equations

The methods discussed in lecture 18 may be used for more than one equation if we regard the symbols as vectors. That is, \mathbf{x} and \mathbf{f} are vectors, and we have a set of equations of the state-space form:

$$\dot{\mathbf{x}} = \mathbf{D}\mathbf{x} = \mathbf{f}(\mathbf{x}, t), \text{ with initial-condition vector } \mathbf{x}(t_0) = \mathbf{x}_0 \quad (1)$$

There is a separate of the derivative for each variable; that is, for n equations,

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \dots, x_n, t) \\ \dot{x}_2 &= f_2(x_1, x_2, \dots, x_n, t) \\ &\cdot \\ &\cdot \\ &\cdot \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n, t) \end{aligned} \quad (2)$$

The functions f_i express the coupling between the solution variables x_i . In applying the fourth order Runge-Kutta method to a system of equations, we obtain the numerical solution for the vector of dependent variables, \mathbf{x} , as given by the expression

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \frac{h}{6}(\mathbf{A} + 2\mathbf{B} + 2\mathbf{C} + \mathbf{D}) \quad (3)$$

where

$$\mathbf{A} = \mathbf{f}(\mathbf{x}_j, t_j) \quad (4)$$

$$\mathbf{B} = \mathbf{f}\left(\mathbf{x}_j + \frac{h}{2}\mathbf{A}, t_j + \frac{h}{2}\right) \quad (5)$$

$$\mathbf{C} = \mathbf{f}\left(\mathbf{x}_j + \frac{h}{2}\mathbf{B}, t_j + \frac{h}{2}\right) \quad (6)$$

$$\mathbf{D} = \mathbf{f}(\mathbf{x}_j + h\mathbf{C}, t_j + h) \quad (7)$$

Here \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} are vectors corresponding to the function evaluations shown in equations 3 through 7 above.

LECT. 19

Example 1: Given a system with the following state-space equations.

$$\dot{x}_1 = -0.5x_1 - 4.0x_2 \text{ with } x_1(0) = 0 \quad (8)$$

and

$$\dot{x}_2 = x_1 \text{ with } x_2(0) = 2 \quad (9)$$

Calculate the values of x_1 and x_2 for $t = 0$ to 5 second ($h = 0.5$) by using the fourth-order Runge-Kutta method.

Solution:

To calculate the values of x_1 and x_2 at $t = 0.5$, for example, we use the following equations:

$$x_1(0.5) = x_1(0) + \frac{h}{6} (A_1 + 2B_1 + 2C_1 + D_1) \quad (10)$$

and

$$x_2(0.5) = x_2(0) + \frac{h}{6} (A_2 + 2B_2 + 2C_2 + D_2) \quad (11)$$

where

$$A_1 = f_1(x_1(0), x_2(0), t) = -8 \quad (12)$$

$$A_2 = f_2(x_1(0), x_2(0), t) = 0 \quad (13)$$

$$B_1 = f_1(x_1(0) + \frac{h}{2} A_1, x_2(0) + \frac{h}{2} A_2, t + \frac{h}{2}) = -7 \quad (14)$$

$$B_2 = f_2(x_1(0) + \frac{h}{2} A_1, x_2(0) + \frac{h}{2} A_2, t + \frac{h}{2}) = -2 \quad (15)$$

$$C_1 = f_1(x_1(0) + \frac{h}{2} B_1, x_2(0) + \frac{h}{2} B_2, t + \frac{h}{2}) = -5.125 \quad (16)$$

$$C_2 = f_2(x_1(0) + \frac{h}{2} B_1, x_2(0) + \frac{h}{2} B_2, t + \frac{h}{2}) = -1.75 \quad (17)$$

$$D_1 = f_1(x_1(0) + hC_1, x_2(0) + hC_2, t + h) = -3.21875 \quad (18)$$

$$D_2 = f_2(x_1(0) + hC_1, x_2(0) + hC_2, t + h) = -2.5625 \quad (19)$$

Then

$$x_1(0) = 0 + \frac{0.5}{6} (-8 - 2 \times 7 - 2 \times 5.125 - 3.21875) = -2.9557 \quad (20)$$

$$x_2(0) = 2 + \frac{0.5}{6} (0 - 2 \times 2 - 2 \times 1.75 - 2.5625) = 1.1615 \quad (21)$$

By the same approach we can calculate other values. The values of x_1 and x_2 for $t = 0$ to $t = 5$ second ($h = 0.5$) are given in Table 1.

TABLE 1 SOLUTION OF EXAMPLE 1

t	x₁	x₂
0	0	2.0000
0.5000	-2.9557	1.1615
1.0000	-2.8869	-0.4175
1.5000	-0.5261	-1.3091
2.0000	1.7263	-0.9546
2.5000	2.0944	0.0834
3.0000	0.7061	0.8223
3.5000	-0.9356	0.7384
4.0000	-1.4617	0.0831
4.5000	-0.7017	-0.4918
5.0000	0.4489	-0.5448

Selection of the Step Size

It is clear that the accuracy with which the numerical solution is computed depends upon the step size employed. A fixed step size that is too large can give results either with large errors or cause the numerical process to become unstable. On the other hand, too small an h causes an inordinately large number of steps to be taken, which, in the extreme, can cause large computational errors due to round-off. This happens because computations are normally performed with a limited number of digits.

1. Selection of A fixed Step Size

In selecting the proper step size for a numerical method of solving a system equation, the primary goal is to divide the resulting waveform into enough pieces to be able to resolve the character of the wave accurately, without resorting to the use of an excessive number of points and thereby incurring significant round-off errors. Two different dynamic effects must be considered in the selection of the step size:

- a) The speed of the response, determined by the natural dynamics of the system.
- b) The waveform of the input signal.

2. Selection of output interval and final time

The output interval is the time between two distinct recordings of data, and of course, it must be an integer multiple of the step size. Although there are no definite rules, suggested output intervals are as follows:

h for a first-order system

LECT. 19

2h to 5h	for a second-order system
4h to 10h	for a third-order system
10h to 20h	for a high-order system

The final time is the ending time of the simulation and may also generally be a function of the natural dynamics of the system. Thus, it, too, may be based on the solution step size. The time required to achieve a desired response or reach a steady state is generally different for each simulation; however in the absence of other information, the following guidelines may be used for estimating the required final time:

50h	for a first-order system
50h to 200h	for a second-order system
100h to 500h	for a third-order system
200h to 1000h	for a high-order system

Variable step size methods

In some instances, the step size dictated by the requirements outlined in the previous section may not be required, except over a portion of the period during which the solution is formed. As time advances, the frequency content of the input function may change, or, in nonlinear systems, the system dynamic characteristics may change.

In order to be able to change the integration time step as the solution progresses, some estimate of the current accuracy of the calculated results must be made, and different methods of doing this are available, depending on the particular numerical integration scheme that is being employed.

One of the most straightforward approaches involves calculating the response at the next point in time in two ways. First, the new solution point is calculated in the usual way by advancing from t to $t + h$ using a step of size h . Let this solution be x_a . Then another approximation to the solution at $t + h$ is found by taking two small steps of size $h/2$. Call this second solution estimate x_c .

An estimate of the truncation error that occurs during the step can be computed from the two separate solutions calculated for the value of $x(t + h)$:

$$E_T \cong \frac{x_c - x_a}{2^1 - 1} \quad (22)$$

LECT. 19

Since the Euler method is a first-order method, the constant 2 appears to the first power in the above equation. If the analysis were repeated for a method of order p , we would obtain:

$$E_T \cong \frac{X_c - X_a}{2^p - 1} \quad (23)$$

for an estimate of the pre-step truncation error.

For the fourth-order Runge-Kutta method, we could estimate the pre-step error by

$$E_T \cong \frac{X_c - X_a}{2^4 - 1} = \frac{X_c - X_a}{15} \quad (24)$$

If, during the solution process, it is found that the pre-step error estimate is larger than some preselected value, the step size may be reduce at that point in the computation and the time advance continued. On the other hand, if the error is very small, the step size may safely be increased.

An alternative method of estimating the error involves using integration methods of different orders to calculate the estimate over a given time step. For example, we might use a fourth-order Runge-Kutta method to advance the solution one time step and then repeat the calculation for that step using a fifth-order Runge-Kutta method. An estimate of the error can then be found from the difference in the two calculated values at the endpoint and a decision made about whether to change the step size.

End of Lecture nineteen

End of Lectures

End of Lectures