Xi'an Jiaotong-Liverpool University
西交利物浦大学

MEC208 Instrumentation and Control System

2023-24 Semester 2

Dr. Yuqing Chen

Email: Yuqing.chen@xjtlu.edu.cn

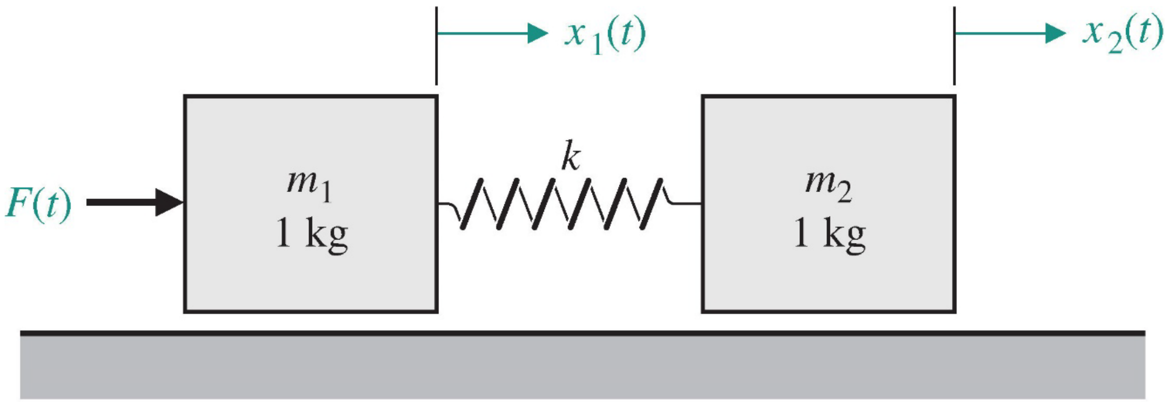
Office: EB330

Department of Mechatronics and Robotics

School of Advanced Technology

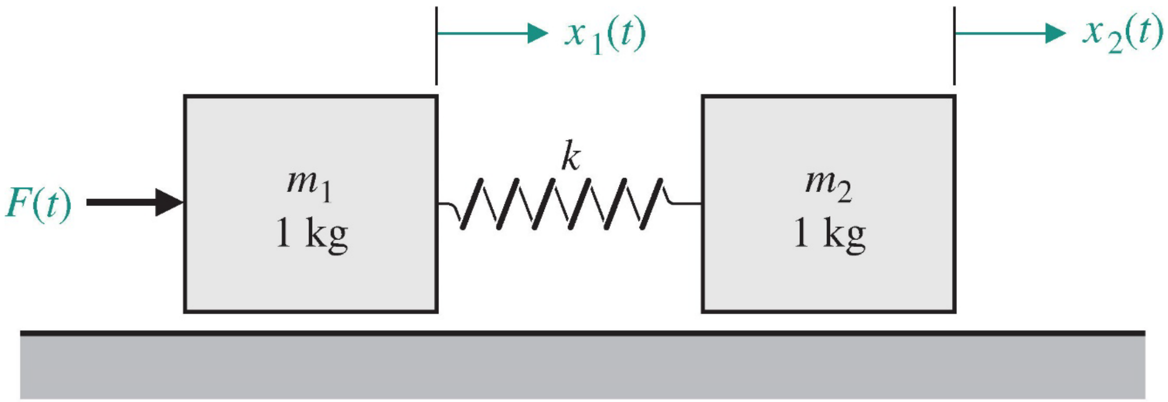
Quiz 7.1

Determine the transfer function between $x_2(t)$ and $F(t)$ for the following system (assume k=1):



Quiz 7.1

Determine the transfer function between $x_2(t)$ and $F(t)$ for the following system (assume k=1):



Solution:

For m_2:

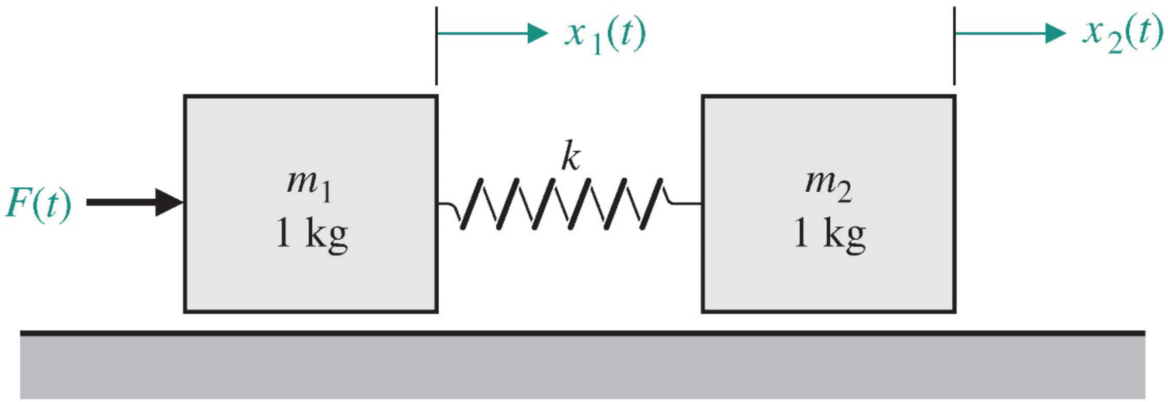
$$k(x_1 - x_2) = m_2\ddot{x}_2$$

For m_1:

$$F - k(x_1 - x_2) = m_1\ddot{x}_1$$

Quiz 7.1

Determine the transfer function between $x_2(t)$ and $F(t)$ for the following system (assume k=1):



Solution:

$$k[X_1(s) - X_2(s)] = m_2 s^2 X_2(s)$$

$$F(s) - k[X_1(s) - X_2(s)] = m_1 s^2 X_1(s)$$

Notice $m_1 = m_2 = 1kg, k = 1,$

$$\frac{X_2(s)}{F(s)} = \frac{1}{s^4 + 2s^2}$$

Quiz 7.2

A system has the following transfer function:

$$\frac{Y(s)}{R(s)} = \frac{4(s+50)}{s^2+30s+200}$$

(a) If $r(t)$ is a unit step input, find the output $y(t)$;

(b) What is the final value of $y(t)$?

Solution:

(a) $r(t) = 1$, $R(s) = \frac{1}{s}$, then

$$Y(s) = \frac{4(s+50)}{s(s^2+30s+200)} = \frac{1}{s} + \frac{-1.6}{s+10} + \frac{0.6}{s+20}$$

$$y(t) = \mathcal{L}^{-1}\big(Y(s)\big) = 1 - 1.6e^{-10t} + 0.6e^{-20t}$$

Quiz 7.2

A system has the following transfer function:

$$\frac{Y(s)}{R(s)} = \frac{4(s+50)}{s^2+30s+200}$$

(a) If $r(t)$ is a unit step input, find the output $y(t)$;

(b) What is the final value of $y(t)$?

Solution:

(b) Two ways to compute steady state value:

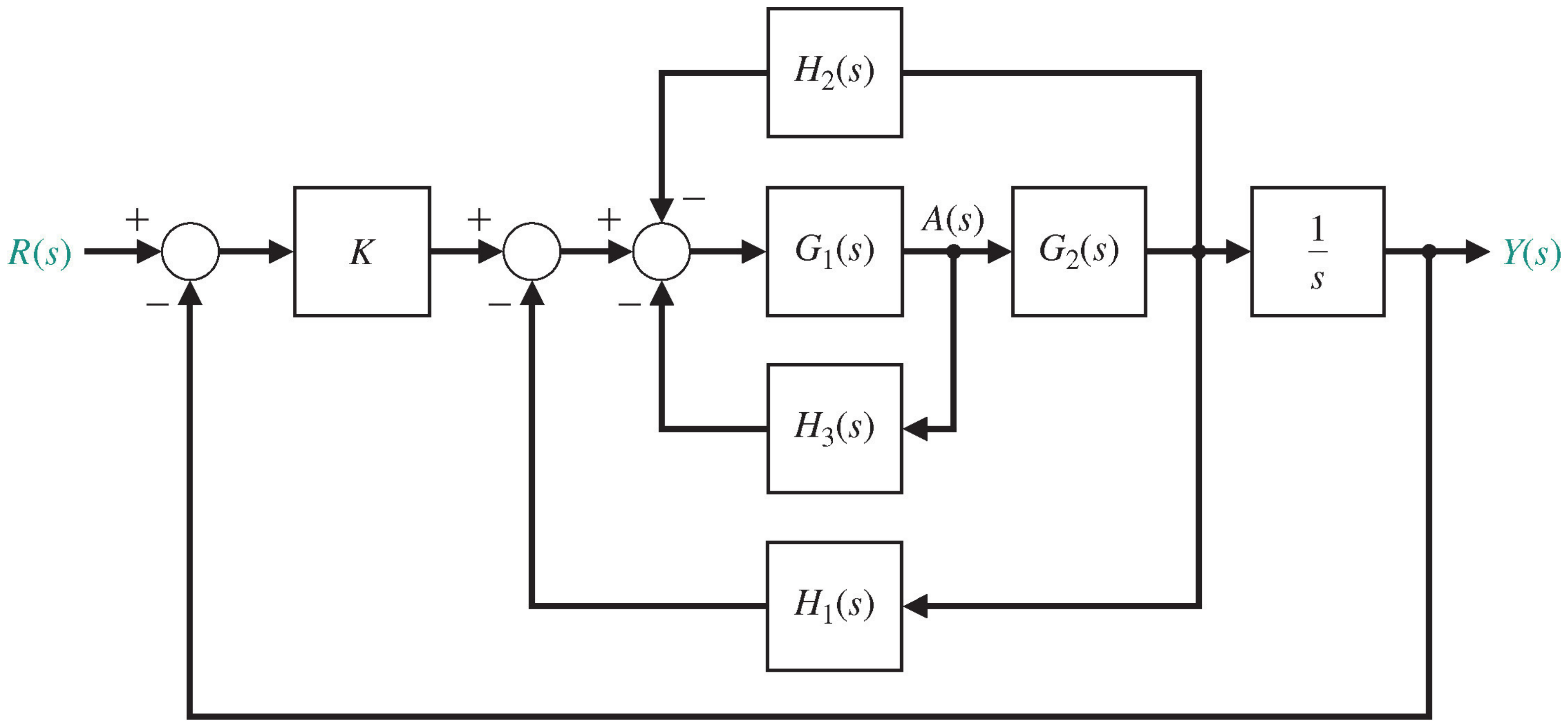
$$y(\infty) = \lim_{s \to 0} sY(s) = \frac{4(s+50)}{(s^2+30s+200)} = \frac{200}{200} = 1$$

Or

$$y(\infty) = \lim_{t \to \infty} 1 - 1.6e^{-10t} + 0.6e^{-20t} = 1$$

Quiz 7.3

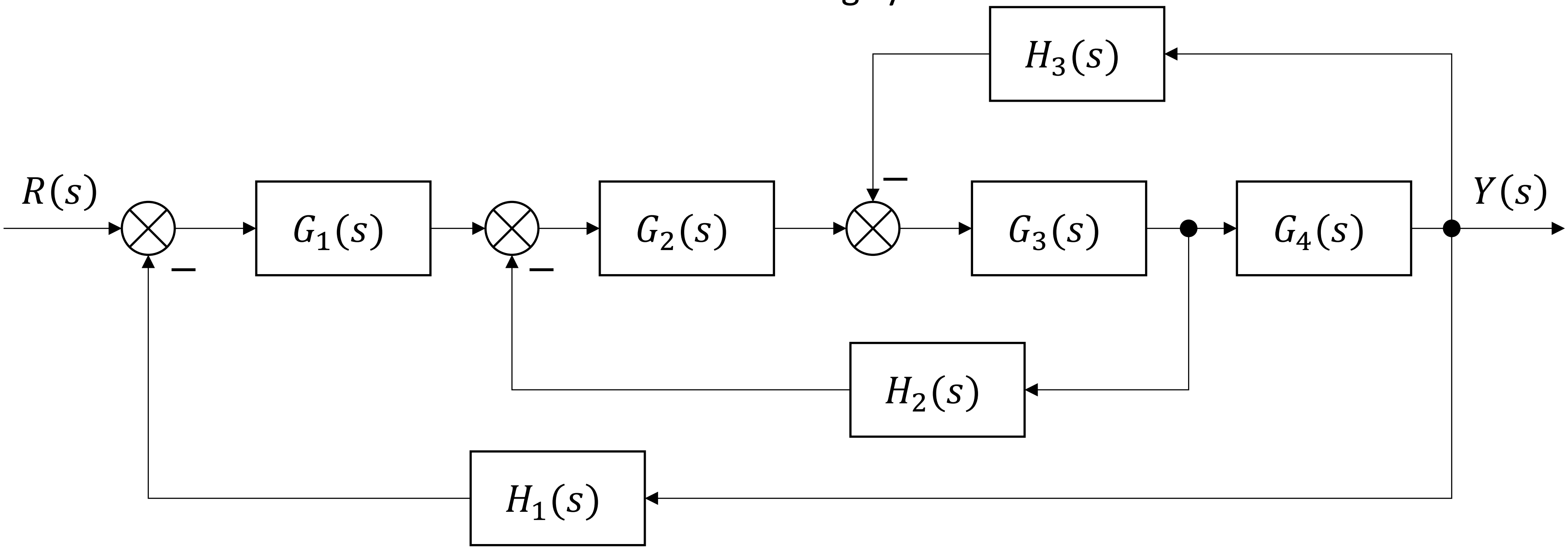
Determine the transfer function for the following system:



$$\frac{Y(s)}{R(s)} = \frac{KG_1(s)G_2(s)/s}{1 + G_1(s)H_3(s) + G_1(s)G_2(s)[H_1(s) + H_2(s)] + KG_1(s)G_2(s)/s}$$

Quiz 7.4

Determine the transfer function for the following system:



$$\frac{Y(s)}{R(s)} = \frac{G_1G_2G_3G_4}{1 + G_2G_3H_2 + G_3G_4H_3 + G_1G_2G_3G_4H_1}$$

Lecture 8

Outline

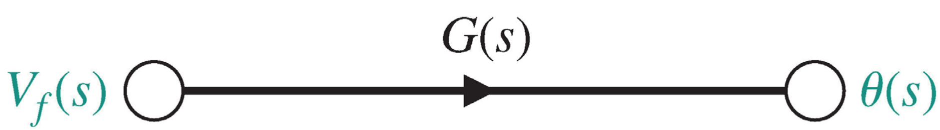
Control Systems:

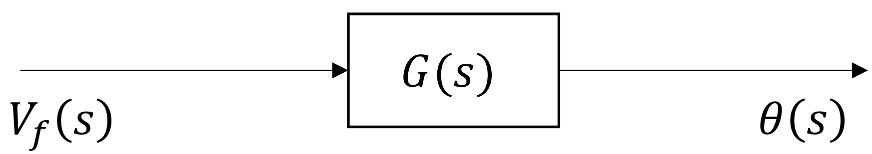
Mathematical Models of Systems

- ❑ Differential Equations of Physical Systems
- ❑ Linear Approximation of Physical Systems
- ❑ The Laplace Transform
- ❑ The Transfer Function of Linear Systems
- ❑ Block Diagram Models
- ❑ *Signal-Flow Graph Models*
- ❑ Simulation Tool

Signal-Flow Graph Models

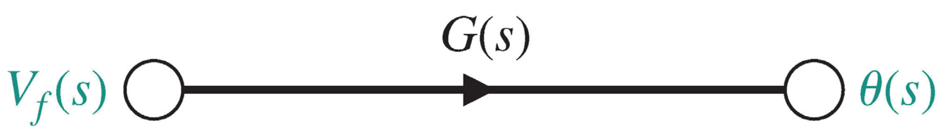
- A **signal-flow graph** is a diagram consisting of **nodes** that are connected by several **directed branches** and is a graphic representation of a set of linear relations;
- Signal-flow graph is particularly useful for feedback control systems because feedback theory is primarily concerned with the flow and processing of signals in the system;

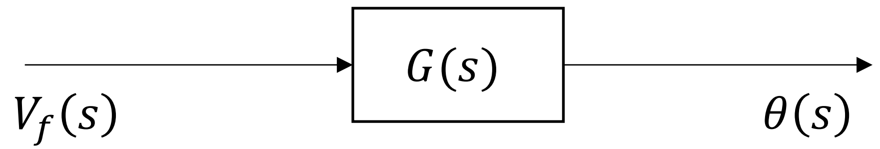




Signal-Flow Graph Models

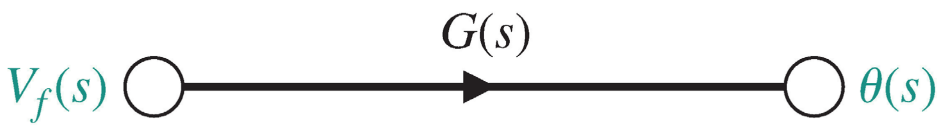
- The basic element of a signal-flow graph is a unidirectional path segment called a **branch**, which relates the dependency of input and an output variable in a manner equivalent to a **block** of a block diagram.
- For complex systems, the block diagram method can become difficult to complete. By using the signal-flow graph model, the reduction procedure (used in the block diagram method) is not necessary to determine the relationship between system variables.





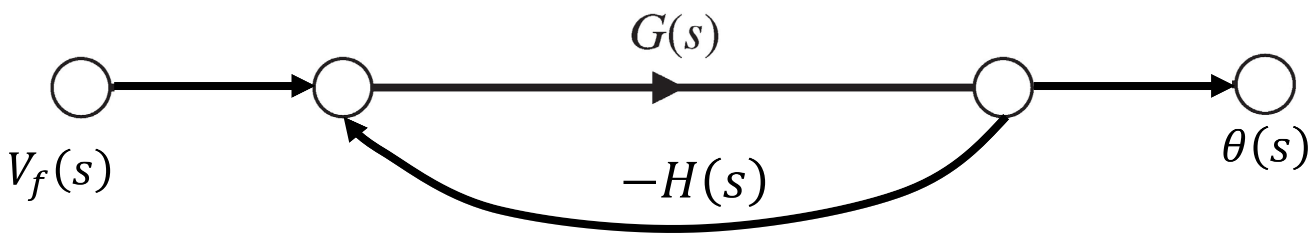
Basic Concepts

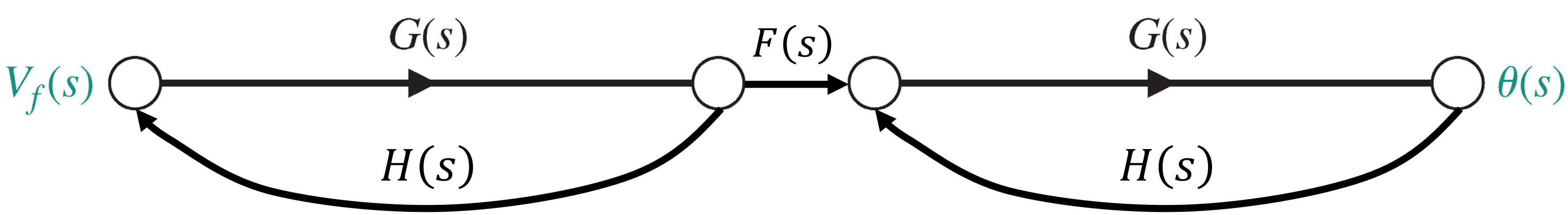
- **Nodes**: are the input and output points or junctions; all branches leaving a node will pass the nodal signal to the output node of each branch (unidirectional); the summation of all signals entering a node is equal to the node variable;
- A **Path**: is a branch or a continuous sequence of branches that can be traversed from one node (signal) to another node (signal);



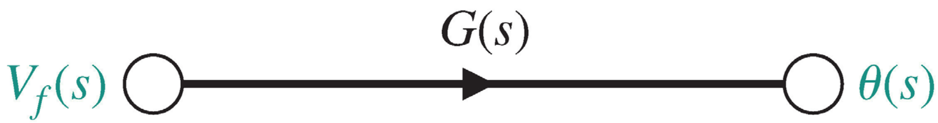
Basic Concepts

- A **Loop**: is a closed path that originates and terminates on the same node, with no node been met twice along the path;
- Two loops are said to be **nontouching** if they don't have a common node; two touching loops share one or more common nodes;

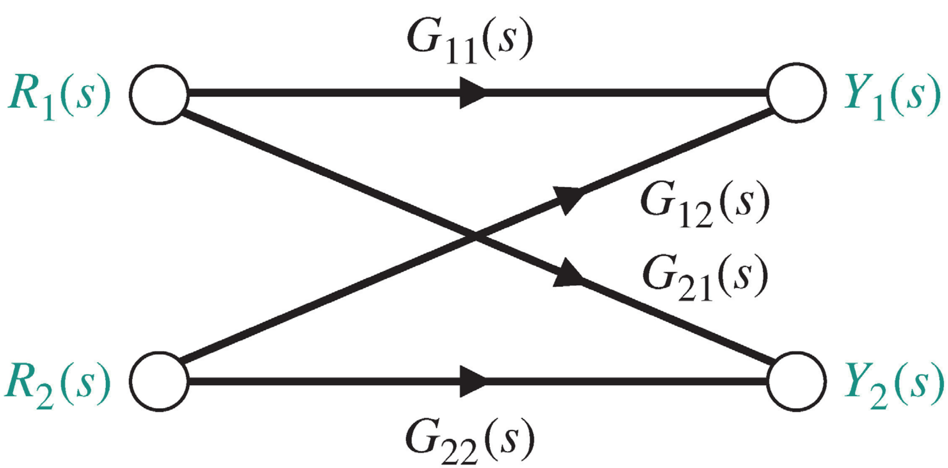




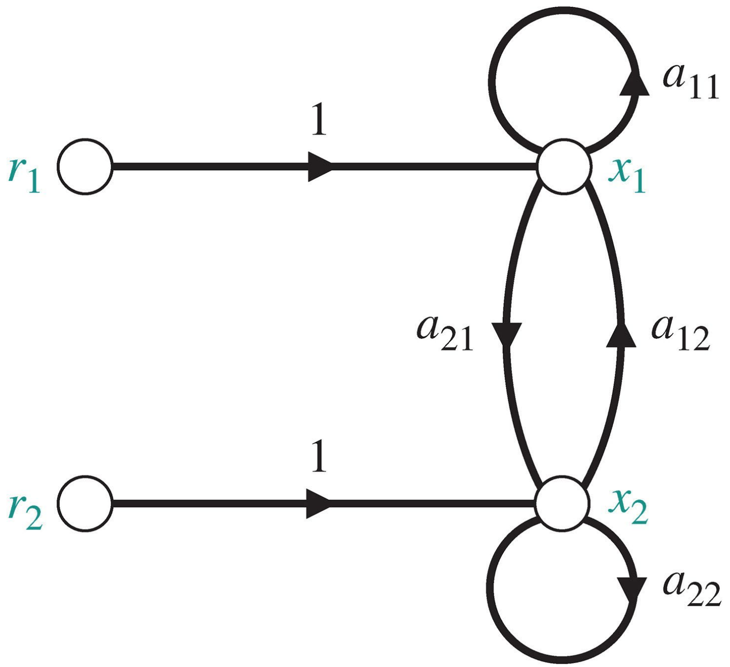
Examples



(a) Signal-flow graph of a DC motor.



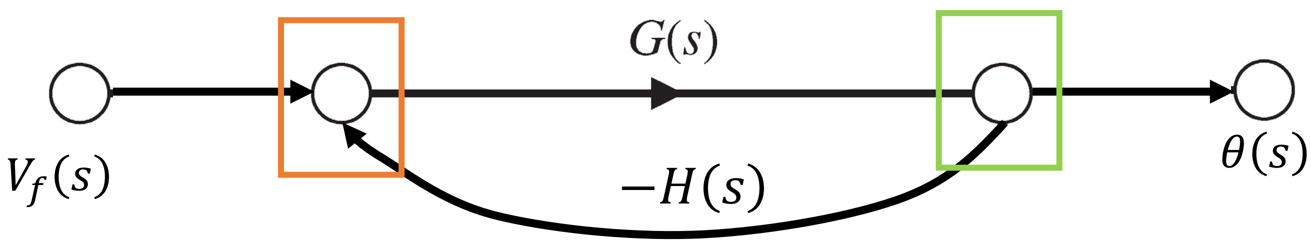
(b) Signal-flow graph of two-input, two-output interconnected system.

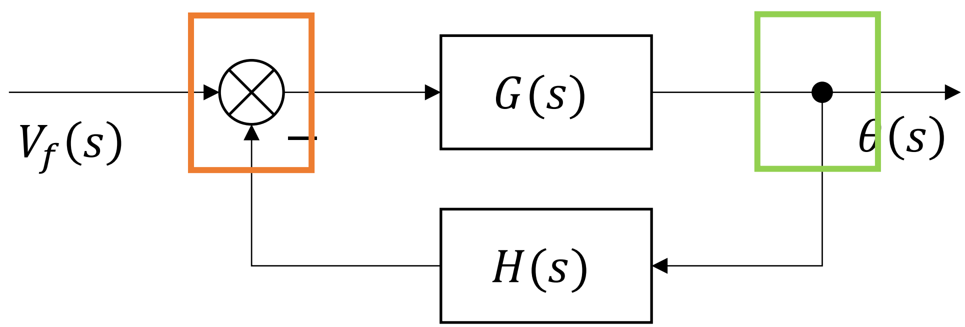


(c) Signal-flow graph of two algebraic equations.

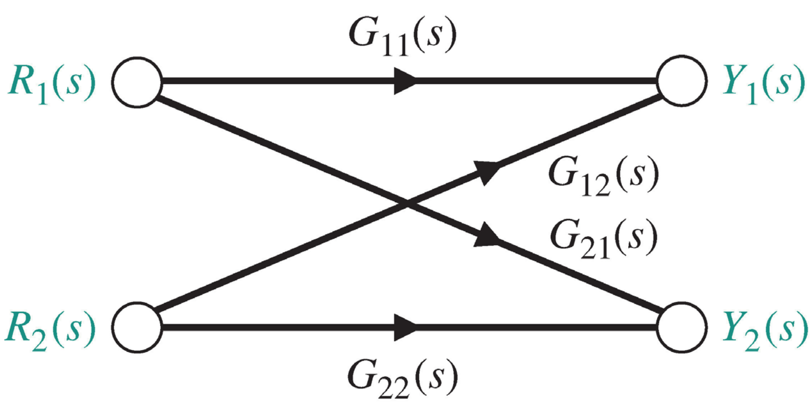
Equivalence to Block Diagram

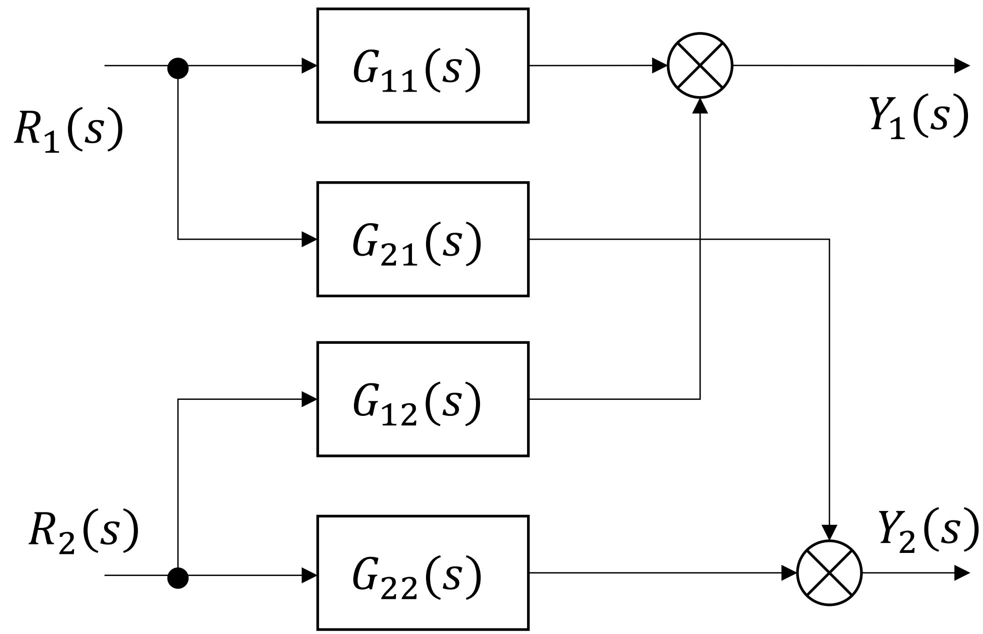
- Signal flow graph and the block diagram are equivalent.
- Below is an example for typical feedback system.
- Signals flow out of a node denote pickoff point; signals flow into a node denote summing point.



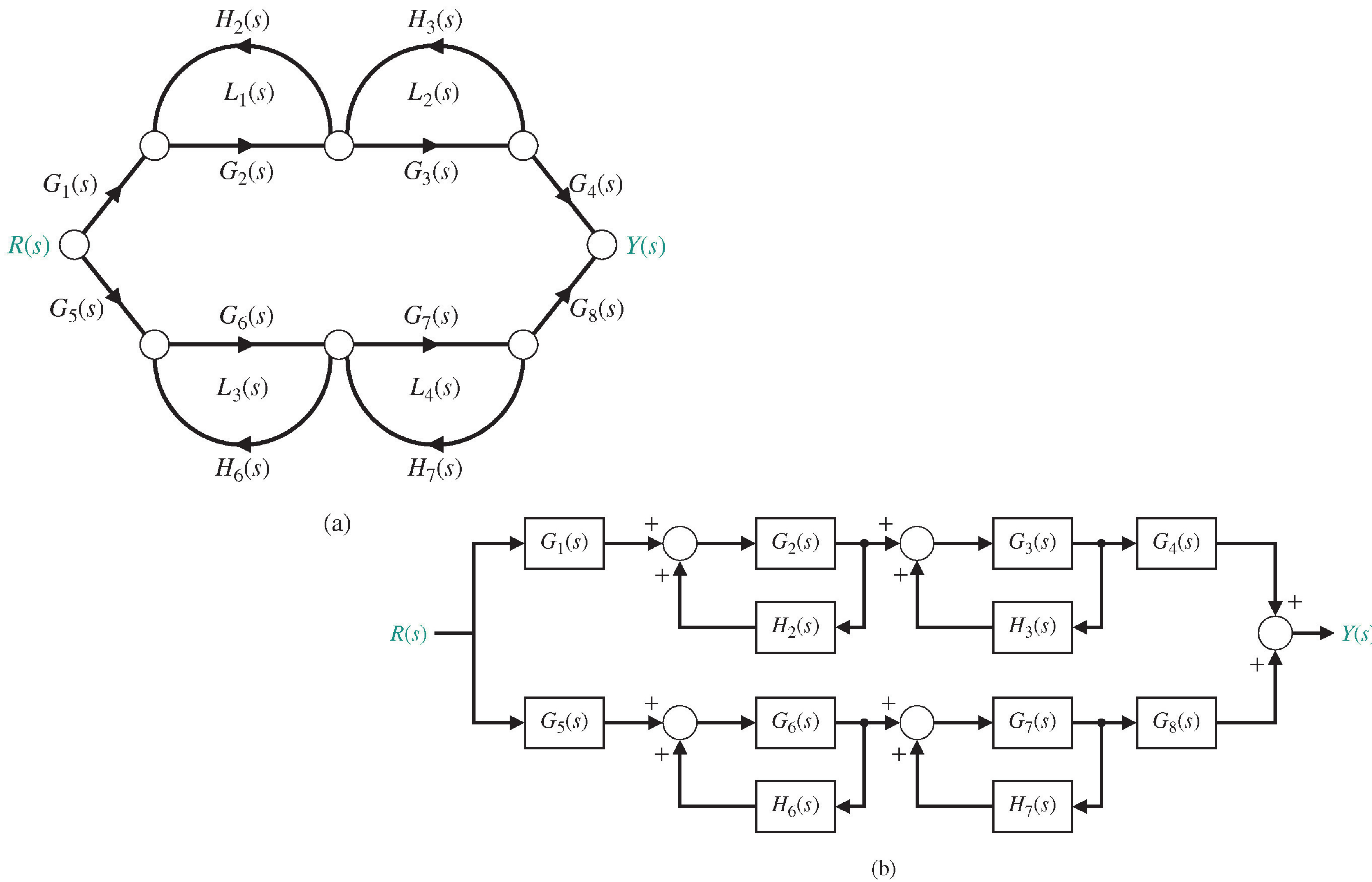


Equivalence to Block Diagram

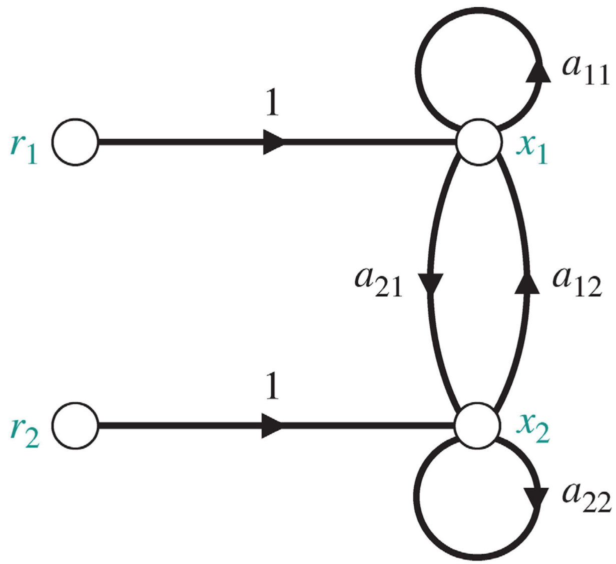




Equivalence to Block Diagram



Determine Transfer Function from the Graph



$$a_{11}x_1 + a_{12}x_2 + r_1 = x_1$$

$$a_{21}x_1 + a_{22}x_2 + r_2 = x_2.$$

$$x_1(1 - a_{11}) + x_2(-a_{12}) = r_1,$$

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

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

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

Mason's Signal-flow Gain Formula

In general, the linear dependence $T_{ij}(s)$ between the independent variable x_i (often called the input variable) and a dependent variable x_j (output variable)is given by:

$$T_{ij}(s) = \frac{\sum_{k=1}^{n} P_{ijk}(s)\Delta_{ijk}(s)}{\Delta}$$

The summation is taken over all possible k paths from x_i to x_j.

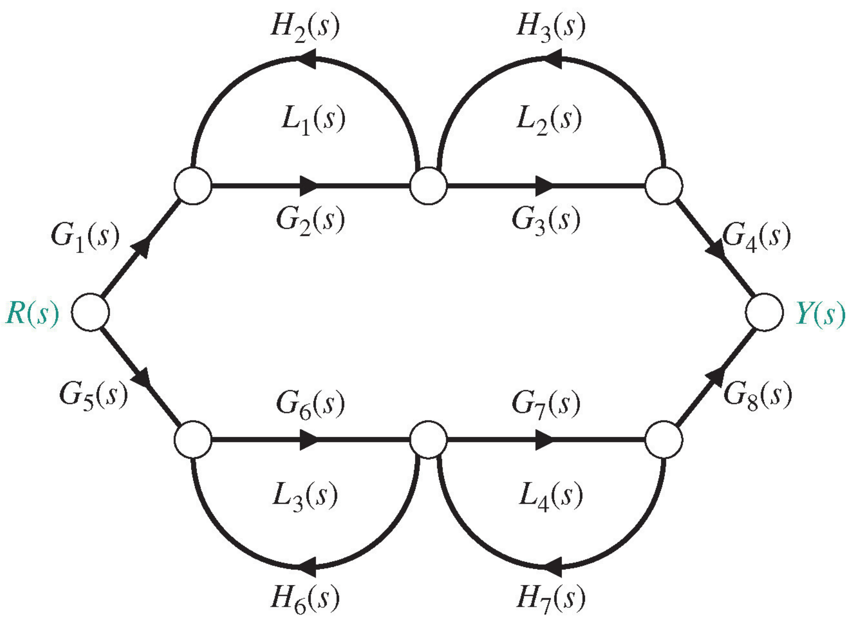
- P_{ijk}: is the k^{th} path gain from x_i to x_j, defined as the product of gains of the branches on the k^{th} path, traversed in the direction of the arrows with no node encountered more than once;
- Δ_{ijk}: cofactor, is the determinant with the loops touching the kth path removed.
- Δ: the determinant, is:

$$\Delta = 1 - \sum_{n=1}^{N} L_n + \sum_{\substack{n,\, m \\ \text{nontouching}}} L_n L_m - \sum_{\substack{n,\, m,\, p \\ \text{nontouching}}} L_n L_m L_p + \cdots,$$

Where L_q equals the value of qth loop transmittance, therefore:

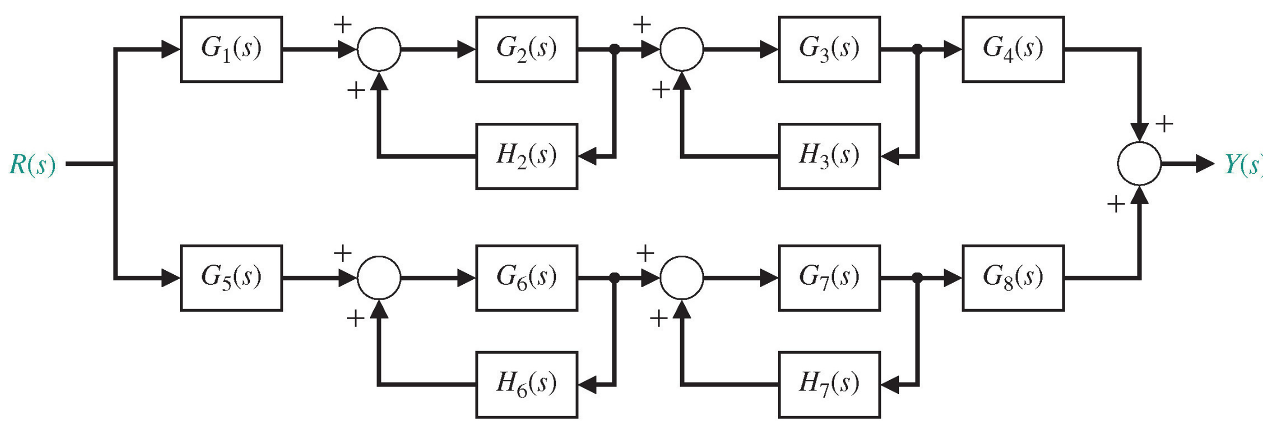
$$\begin{aligned} \Delta = 1 &- (\text{sum of all different loop gains}) \\ &+ (\text{sum of the gain products of all combinations of two nontouching loops}) \\ &- (\text{sum of the gain products of all combinations of three nontouching loops}) \\ &+ \cdots. \end{aligned}$$

Mason’s Rule: Application



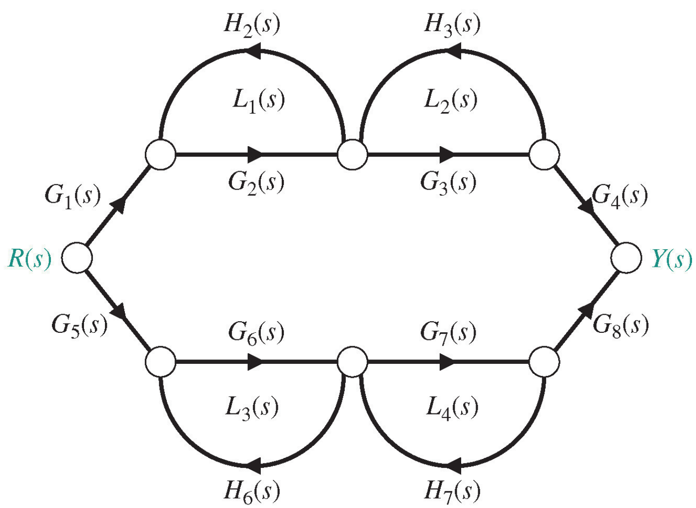
(a)

How to obtain Transfer Function between Input R(s) and Output Y(s)?



(b)

Example 10.1



$$T_{ij}(s) = \frac{\sum_{k=1}^{n} P_{ijk}(s)\Delta_{ijk}(s)}{\Delta}$$

1. How many possible paths? $P_1 = G_1G_2G_3G_4$ (path 1) and $P_2 = G_5G_6G_7G_8$ (path 2).

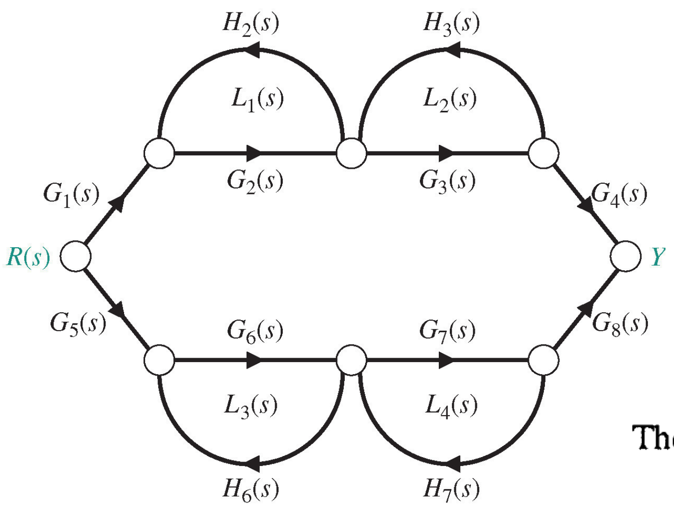
2. How many loops? There are four self-loops:

$$L_1 = G_2H_2, \quad L_2 = H_3G_3, \quad L_3 = G_6H_6, \quad \text{and} \quad L_4 = G_7H_7.$$

3. How many groups of 2 nontouching loops? Loops L_1 and L_2 do not touch L_3 and L_4.

4. How many groups of 3 nontouching loops? None.

Example 10.1



$$T_{ij}(s) = \frac{\sum_{k=1}^{n} P_{ijk}(s)\Delta_{ijk}(s)}{\Delta}$$

$\Delta = 1 -$ (sum of all different loop gains)
$+$ (sum of the gain products of all combinations of two nontouching loops)
$-$ (sum of the gain products of all combinations of three nontouching loops)
$+ \cdots .$

There are four self-loops:

$$L_1 = G_2H_2, \quad L_2 = H_3G_3, \quad L_3 = G_6H_6, \quad \text{and} \quad L_4 = G_7H_7.$$

Loops L_1 and L_2 do not touch L_3 and L_4. Therefore, the determinant is

$$\Delta = 1 - (L_1 + L_2 + L_3 + L_4) + (L_1L_3 + L_1L_4 + L_2L_3 + L_2L_4).$$

The cofactor of the determinant along path 1 is evaluated by removing the loops that touch path 1 from Δ. Hence, we have

$$L_1 = L_2 = 0 \quad \text{and} \quad \Delta_1 = 1 - (L_3 + L_4).$$

Similarly, the cofactor for path 2 is

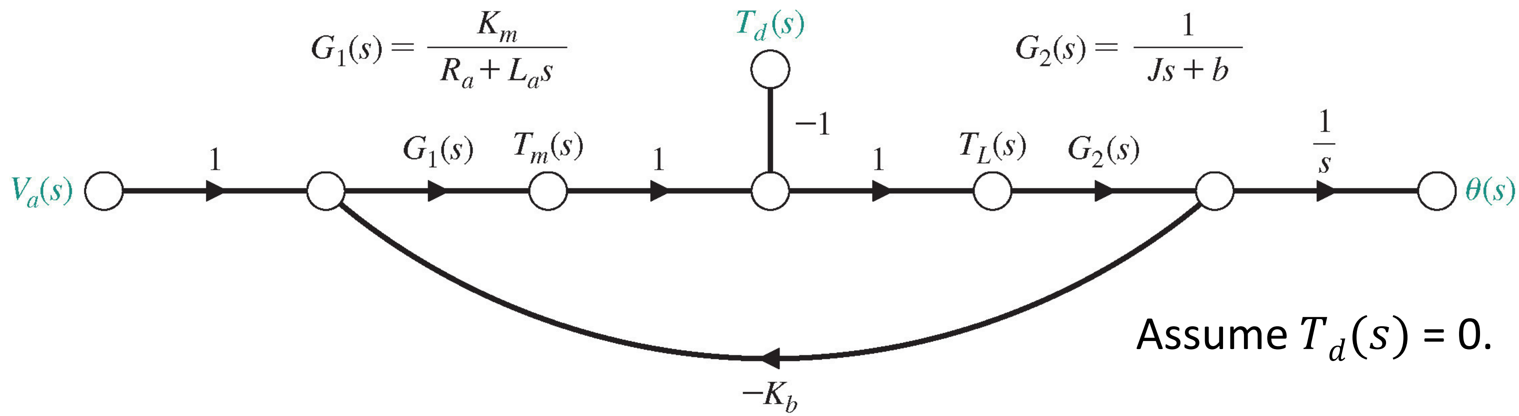
$$\Delta_2 = 1 - (L_1 + L_2).$$

Therefore, the transfer function of the system is

$$\frac{Y(s)}{R(s)} = T(s) = \frac{P_1\Delta_1 + P_2\Delta_2}{\Delta}$$

$$= \frac{G_1G_2G_3G_4(1 - L_3 - L_4) + G_5G_6G_7G_8(1 - L_1 - L_2)}{1 - L_1 - L_2 - L_3 - L_4 + L_1L_3 + L_1L_4 + L_2L_3 + L_2L_4}$$

Example 10.2



Assume $T_d(s) = 0$.

Signal-flow graph of the armature-controlled DC motor

There is only 1 forward path, which touches one loop:

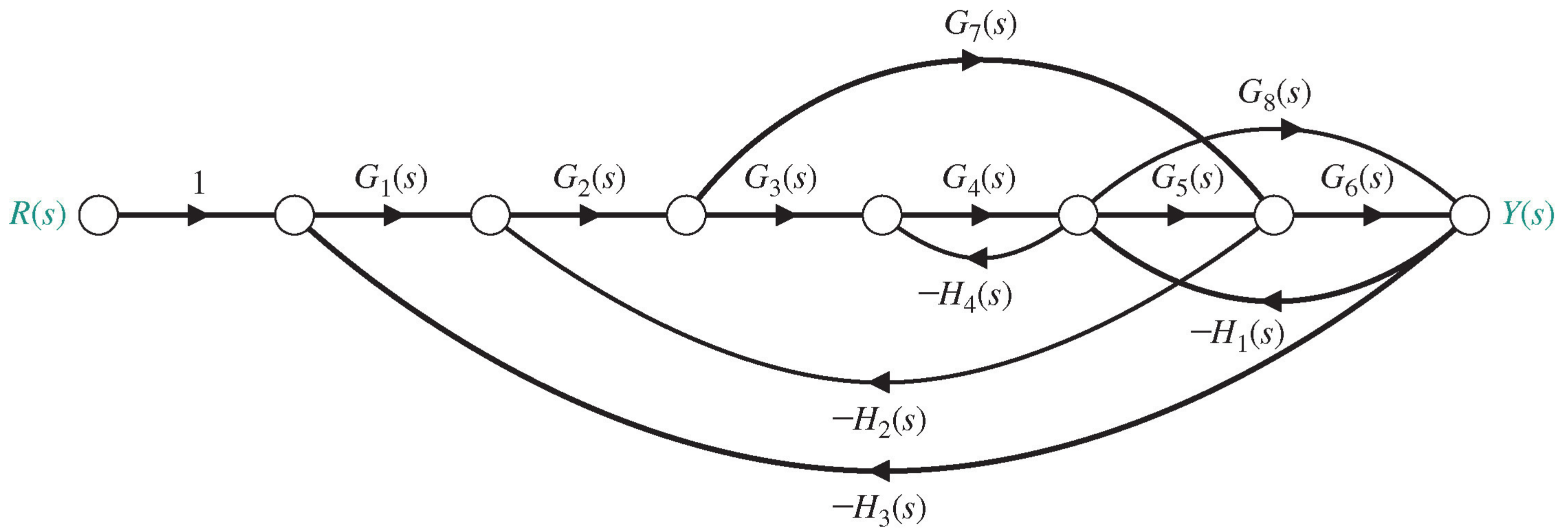
$$P_1(s) = \frac{1}{s} G_1(s) G_2(s) \quad \text{and} \quad L_1(s) = -K_b G_1(s) G_2(s). \qquad \Delta_1 = 1$$

Therefore, the transfer function is

$$T(s) = \frac{P_1(s)}{1 - L_1(s)} = \frac{(1/s) G_1(s) G_2(s)}{1 + K_b G_1(s) G_2(s)} = \frac{K_m}{s[(R_a + L_a s)(Js + b) + K_b K_m]}$$

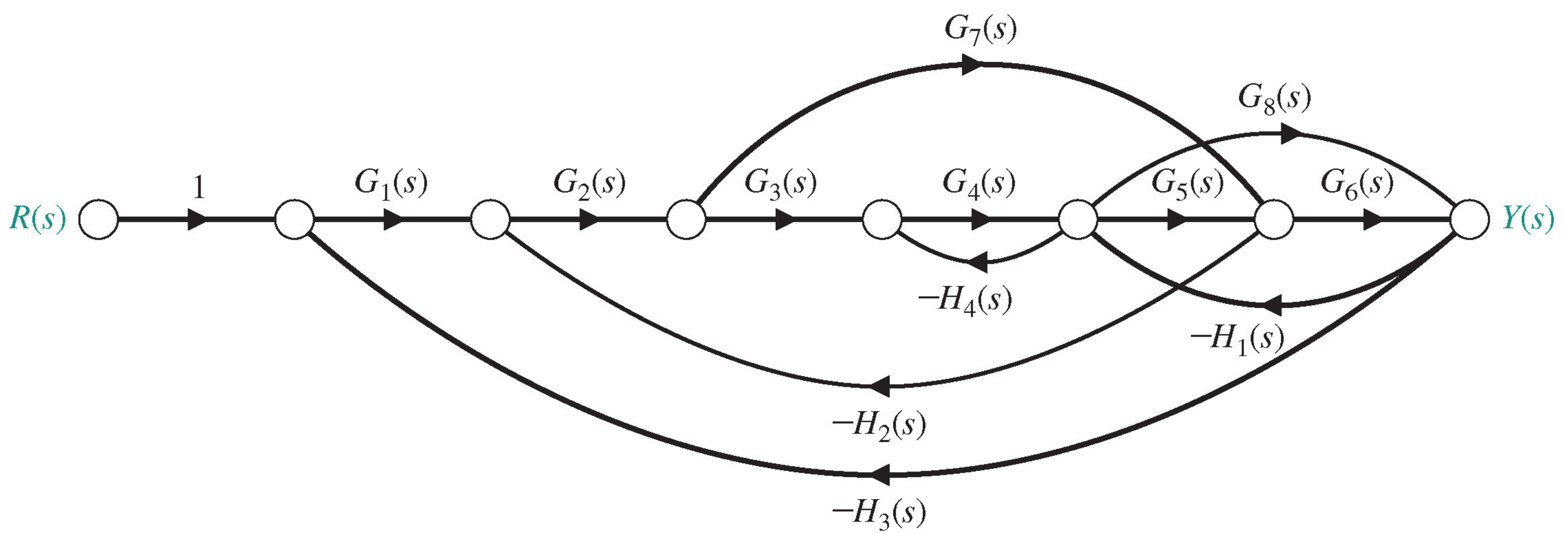
Example 10.3

Consider a reasonably complex system that would be difficult to reduce by block diagram techniques:



Example 10.3

Consider a reasonably complex system that would be difficult to reduce by block diagram techniques:

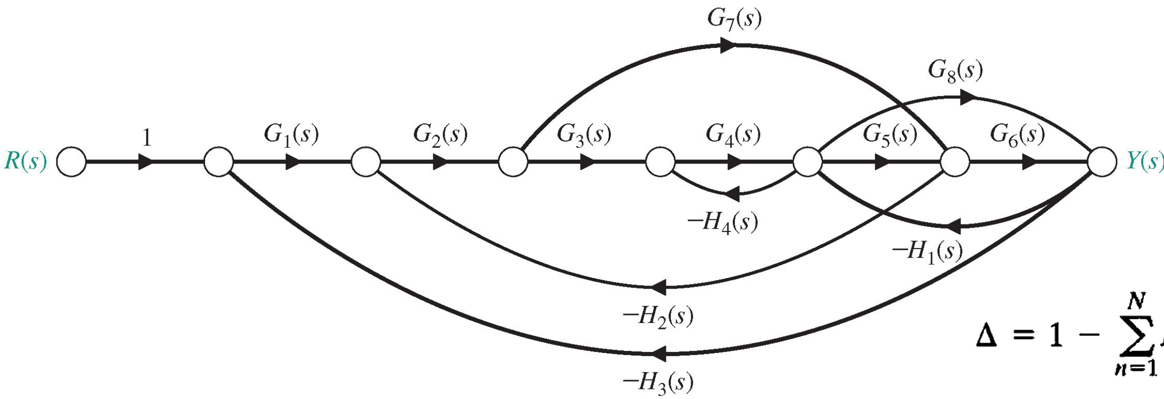


$P_1 = G_1G_2G_3G_4G_5G_6, \quad P_2 = G_1G_2G_7G_6, \quad \text{and} \quad P_3 = G_1G_2G_3G_4G_8.$

The feedback loops are

$L_1 = -G_2G_3G_4G_5H_2, \quad L_2 = -G_5G_6H_1, \quad L_3 = -G_8H_1, \quad L_4 = -G_7H_2G_2,$
$L_5 = -G_4H_4, \quad L_6 = -G_1G_2G_3G_4G_5G_6H_3, \quad L_7 = -G_1G_2G_7G_6H_3, \quad \text{and}$
$L_8 = -G_1G_2G_3G_4G_8H_3.$

Example 10.3



$$T_{ij}(s) = \frac{\sum_{k=1}^{n} P_{ijk}(s)\Delta_{ijk}(s)}{\Delta}$$

$$\Delta = 1 - \sum_{n=1}^{N} L_n + \sum_{\substack{n,\,m \\ \text{nontouching}}} L_n L_m - \sum_{\substack{n,\,m,\,p \\ \text{nontouching}}} L_n L_m L_p + \cdots,$$

Loop L_5 does not touch loop L_4 or loop L_7, and loop L_3 does not touch loop L_4; but all other loops touch. Therefore, the determinant is

$$\Delta = 1 - (L_1 + L_2 + L_3 + L_4 + L_5 + L_6 + L_7 + L_8) + (L_5L_7 + L_5L_4 + L_3L_4).$$

The cofactors are

$$\Delta_1 = \Delta_3 = 1 \quad \text{and} \quad \Delta_2 = 1 - L_5 = 1 + G_4H_4.$$

Finally, the transfer function is

$$T(s) = \frac{Y(s)}{R(s)} = \frac{P_1 + P_2\Delta_2 + P_3}{\Delta}.$$

Outline

Control Systems:

Mathematical Models of Systems

- ❑ Differential Equations of Physical Systems
- ❑ Linear Approximation of Physical Systems
- ❑ The Laplace Transform
- ❑ The Transfer Function of Linear Systems
- ❑ Block Diagram Models
- ❑ Signal-Flow Graph Models
- ❑ *Simulation Tool*

Laplace and Inverse Laplace Transform

- Function laplace:

```
>> help laplace
--- help for sym/laplace ---

 laplace Laplace transform.
    L = laplace(F) is the Laplace transform of the sym F with default
    independent variable t.  The default return is a function of s.
    If F = F(s), then laplace returns a function of z:  L = L(z).
    By definition, L(s) = int(F(t)*exp(-s*t),t,0,inf).
```

- Function ilaplace:

```
>> help ilaplace
--- help for sym/ilaplace ---

 ilaplace Inverse Laplace transform.
    F = ilaplace(L) is the inverse Laplace transform of the sym L
    with default independent variable s.  The default return is a
    function of t.  If L = L(t), then ilaplace returns a function of x:
    F = F(x).
```

Example

- Compute the Laplace transform of $f(t) = t^2 + 2t + 2$
- Compute the inverse Laplace transform of

$$F(s) = \frac{s+6}{(s^2+4s+3)(s+2)}$$

- Check the code file provided in LMO

Solve ODE

- Function dsolve:

```
>> help dsolve
 dsolve Symbolic solution of ordinary differential equations.
    dsolve will not accept equations as strings in a future release.
    Use symbolic expressions or sym objects instead.
    For example, use syms y(t); dsolve(diff(y)==y) instead of  dsolve('Dy=y').

    dsolve(eqn1,eqn2, ...) accepts symbolic equations representing
    ordinary differential equations and initial conditions.

    By default, the independent variable is 't'. The independent variable
    may be changed from 't' to some other symbolic variable by including
    that variable as the last input argument.
```

- Use function *diff* to denote differentiation
  - $\dot{y}(t)$: in Matlab, use *diff(y,t)*
  - $\ddot{y}(t)$*:* in Maltab, use *diff(y,t,2)*

Example

- Solve following differential equation with initial values

$$3\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 1$$

$$y(0) = 0, \dot{y}(0) = 0$$

- Check the code file provided in LMO

Establish Transfer Function

- Transfer function could be expressed in two ways:
- Polynomial:

$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1}s + b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}$$

a_i, b_i are coefficients for denominator and numerator.

- Pole-zero

$$G(s) = \frac{k(s - z_1)(s - z_2) \ldots (s - z_m)}{(s - p_1)(s - p_2) \ldots (s - p_n)}$$

p_i, z_i are the poles and zeros of the transfer function

k is the coefficient in pole-zero transfer function

Establish Transfer Function

- Function tf:

```
>> help tf
 tf  Construct transfer function or convert to transfer function.

   Construction:
     SYS = tf(NUM,DEN) creates a continuous-time transfer function SYS with
     numerator NUM and denominator DEN. SYS is an object of type tf when
     NUM,DEN are numeric arrays, of type GENSS when NUM,DEN depend on tunable
     parameters (see REALP and GENMAT), and of type USS when NUM,DEN are
     uncertain (requires Robust Control Toolbox).
```

- Function zpk

```
>> help zpk
 zpk  Constructs zero-pole-gain model or converts to zero-pole-gain format.

   Construction:
     SYS = zpk(Z,P,K) creates a continuous-time zero-pole-gain (zpk) model
     SYS with zeros Z, poles P, and gains K. SYS is an object of class @zpk.

     SYS = zpk(Z,P,K,Ts) creates a discrete-time zpk model with sampling
     time Ts (set Ts=-1 if the sample time is undetermined).
```

Control System Toolbox required

Example

- Establish following polynomial transfer function in Matlab, then transfer it to zero-pole type. Compute the poles and zeros.

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

- Establish following zpk transfer function in Matlab, then transfer it to polynomial type.

$$F(s) = \frac{10(s+2)}{s(s+1)(s+3)}$$

Example

- The poles of the system is $p_1 = -5$, $p_2 = -2 + 2\text{j}$, $p_3 = -2 - 2j$, no zeros, gain is 100, establish the transfer function in both zpk and polynomial.
- Check the code file provided in LMO

Map of Poles and Zeros

- Function pzmap:

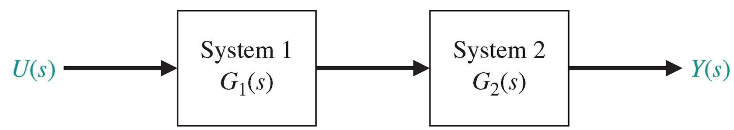
```
pzmap  Pole-zero map of dynamic systems.

   pzmap(SYS) computes the poles and (transmission) zeros of the
   dynamic system SYS and plots them in the complex plane. The poles
   are plotted as x's and the zeros are plotted as o's.

   pzmap(SYS1,SYS2,...) shows the poles and zeros of several systems
   SYS1,SYS2,... on a single plot. You can specify distinctive colors
   for each model, for example:
      pzmap(sys1,'r',sys2,'y',sys3,'g')
```

- Try with the previous examples.

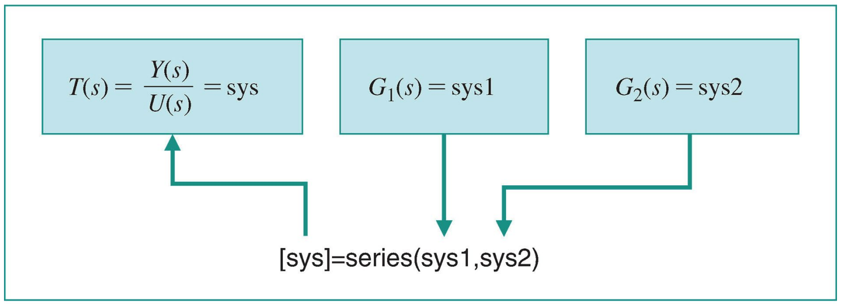
Simplify Block Diagram

- Series connection:

```
>> help series
 series  Series connection of two input/output models.

                                  +------+
                           v2 --->|      |
                   +------+       |  M2  |-----> y2
                   |      |------>|      |
          u1 ----->|      |y1   u2 +------+
                   |  M1  |
                   |      |---> z1
                   +------+

                                                ut/ouput models
                                                1 and INPUTS2
                                                are connected
```



(a)



(b)

Example

- Two blocks connected in series, $G_1(s) = \frac{s+1}{s+2}$, $G_2(s) = \frac{10}{s}$, get the equivalent transfer function.

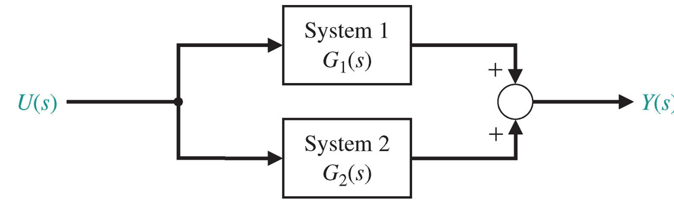
Simplify Block Diagram

- Parallel connection:

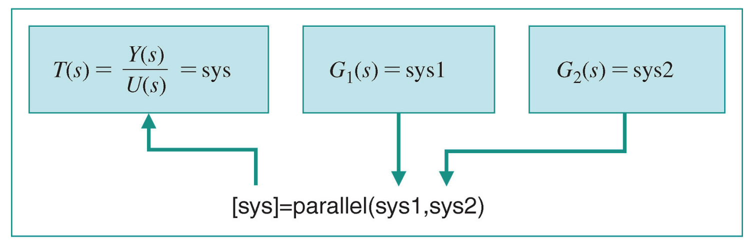
parallel Parallel connection of two input/output models.

```
                     +------+
       v1 ---------->|      |----------> z1
                     |  M1  |
              u1 +-->|      |---+ y1
                 |   +------+   |
        u ------>+              O------> y
                 |   +------+   |
              u2 +-->|      |---+ y2
                     |  M2  |
       v2 ---------->|      |----------> z2
                     +------+
```

M = **parallel**(M1,M2,IN1,IN2,OUT1,OUT2) connects the input/output models M1 and M2 in parallel. The inputs specified by IN1 and IN2 are connected and the outputs specified by OUT1 and OUT2 are summed. The resulting model M maps [v1;u;v2] to [z1;y;z2]. The vectors IN1 and IN2 contain indices into the input vectors of M1 and M2, respectively, and define the input channels u1 and u2 in the diagram. Similarly, the vectors OUT1 and OUT2 contain indexes into the outputs of M1 and M2.



(a)



(b)

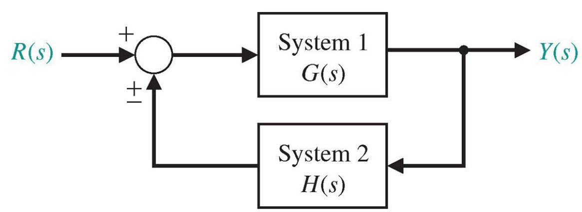
Example

- Two blocks connected in parallel, $G_1(s) = s + 2$, $G_2(s) = \frac{5}{s}$, get the equivalent transfer function.

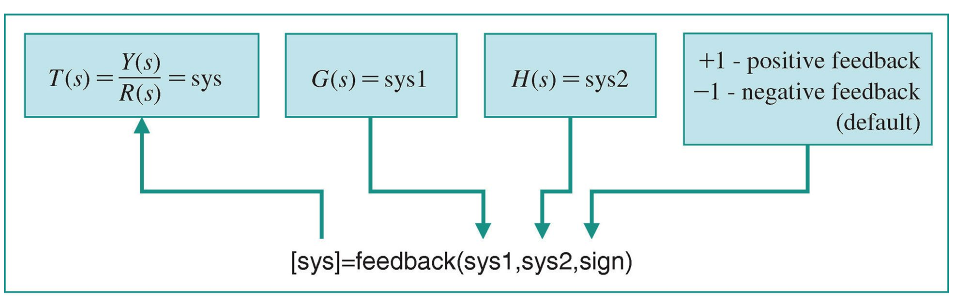
Simplify Block Diagram

- Feedback loop:

```
>> help feedback
 feedback  Feedback connection of two input/output systems.

    M = feedback(M1,M2) computes a closed-loop model M for the feedback loop:

          u --->O---->[ M1 ]----+---> y
                |               |           y = M * u
                +-----[ M2 ]<---+

    Negative feedback is assumed and the model M maps u to y. To apply
    positive feedback, use the syntax M = feedback(M1,M2,+1).
```



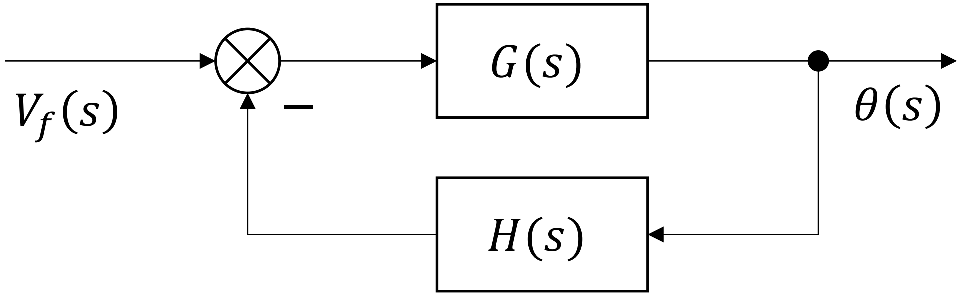
(a)



(b)

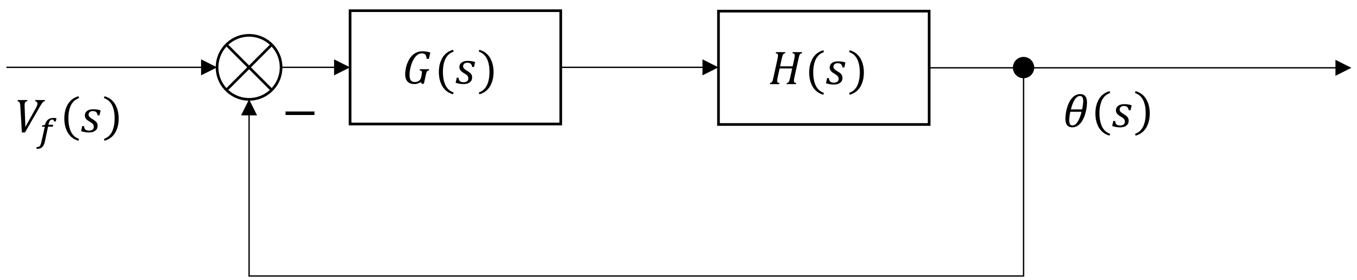
Example 1

- Compute the transfer function in following block diagram, where $G(s) = \frac{1}{500s^2}$, $H(s) = \frac{s+1}{s+2}$.



Example 2

- Compute the transfer function in following block diagram, where $G(s) = \frac{1}{500s^2}$, $H(s) = \frac{s+1}{s+2}$.

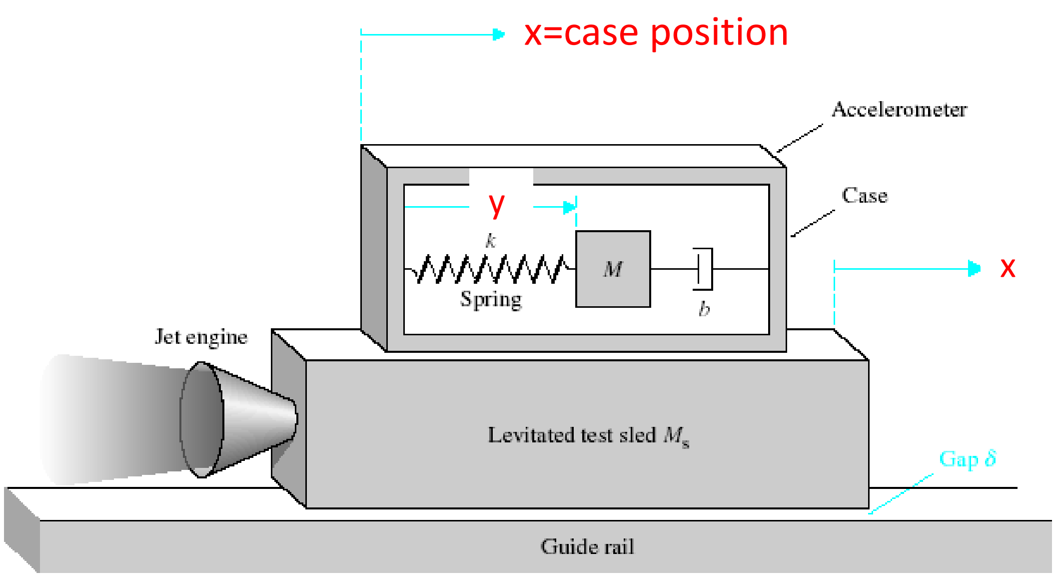


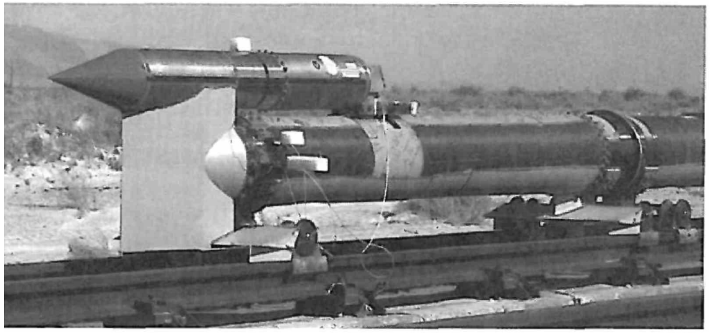
Design Example

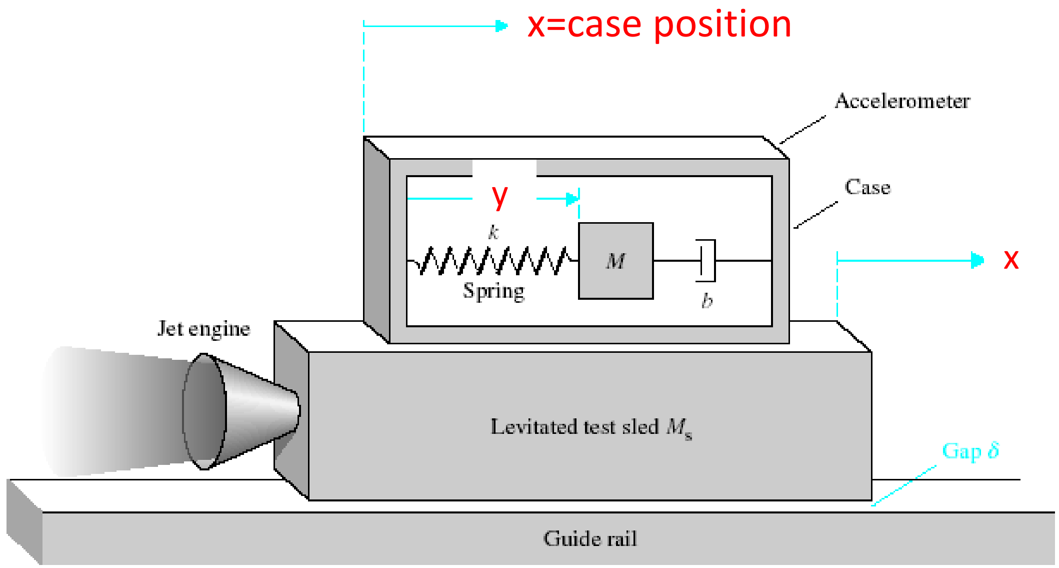
Explore by yourself

Real-world Example: Mechanical Accelerometer

- A mechanical accelerometer is used to measure the acceleration of a rocket test sled. The test sled maneuvers above a guide rail a small distance δ.
- The accelerometer provides a measurement of the acceleration $a(t)$ of the sled, since the position *y* of the mass *M,* with respect to the accelerometer case, is related to the acceleration of the case (and the sled).
- The goal is to design an accelerometer with an appropriate dynamic responsiveness. We wish to design an accelerometer with an acceptable time for the desired measurement characteristic.







The sum of the forces acting on the mass is

$$-b\frac{dy}{dt} - ky = M\frac{d^2}{dt^2}(y + x)$$

or

$$M\frac{d^2y}{dt^2} + b\frac{dy}{dt} + ky = -M\frac{d^2x}{dt^2}.$$

Since

$$M_s\frac{d^2x}{dt^2} = F(t),$$

is the engine force, we have

$$M\ddot{y} + b\dot{y} + ky = -\frac{M}{M_s}F(t),$$

or

$$\ddot{y} + \frac{b}{M}\dot{y} + \frac{k}{M}y = -\frac{F(t)}{M_s}.$$

$$\ddot{y} + \frac{b}{M}\dot{y} + \frac{k}{M}y = -\frac{F(t)}{M_s}.$$

We select the coefficients where $b/M = 3$, $k/M = 2$, $F(t)/M_s = Q(t)$, and we consider the initial conditions $y(0) = -1$ and $\dot{y}(0) = 2$. We then obtain the Laplace transform equation, when the force, and thus $Q(t)$, is a step function, as follows:

$$(s^2Y(s) - sy(0) - \dot{y}(0)) + 3(sY(s) - y(0)) + 2Y(s) = -Q(s).$$

Since $Q(s) = P/s$, where P is the magnitude of the step function, we obtain

$$(s^2Y(s) + s - 2) + 3(sY(s) + 1) + 2Y(s) = -\frac{P}{s},$$

or

$$(s^2 + 3s + 2)Y(s) = \frac{-(s^2 + s + P)}{s}.$$

Thus the output transform is

$$Y(s) = \frac{-(s^2 + s + P)}{s(s^2 + 3s + 2)} = \frac{-(s^2 + s + P)}{s(s + 1)(s + 2)}.$$

Expanding in partial fraction form yields

$$Y(s) = \frac{k_1}{s} + \frac{k_2}{s + 1} + \frac{k_3}{s + 2}.$$

We then have

$$k_1 = \left.\frac{-(s^2 + s + P)}{(s+1)(s+2)}\right|_{s=0} = -\frac{P}{2}.$$

Similarly, $k_2 = +P$ and $k_3 = \dfrac{-P-2}{2}$. Thus,

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

Therefore, the output measurement is

$$y(t) = \frac{1}{2}[-P + 2Pe^{-t} - (P+2)e^{-2t}], \quad t \geq 0.$$

Thank You !