ECE171A: Linear Control System Theory
Lecture 3: System Modeling

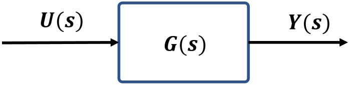
Instructor:
Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistant:
Chenfeng Wu: chw357@ucsd.edu

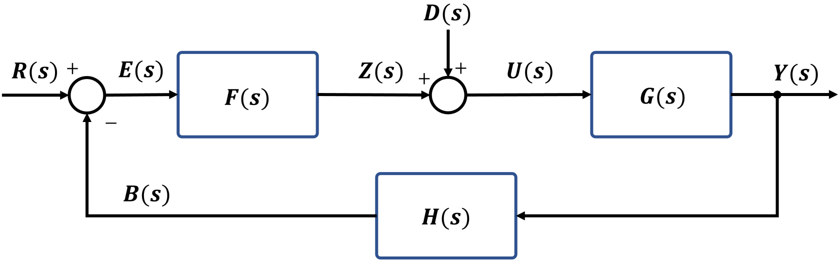
UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

Block Diagram

- **Block diagram**: a graphical representation of a control system
- **Block**: represents the input-output relationship of a system element using its transfer function

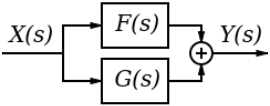


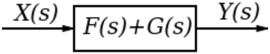
- To represent a multi-element system, the blocks are interconnected
- **Summing point**: adds/subtracts two or more input signals



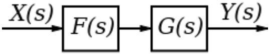
Block Diagram Transformations

- A block diagram can be simplified using equivalent transformations
- **Parallel connection**: if two or more elements are connected in parallel, the total transfer function is the sum of the individual transfer functions:





- **Series connection**: if two or more elements are connected in series, the total transfer function is the product of the individual transfer functions:



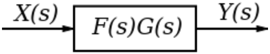
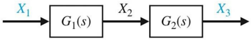
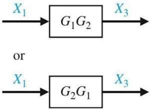
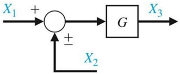
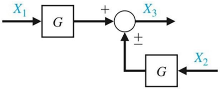
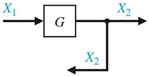
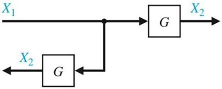
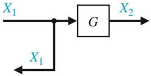
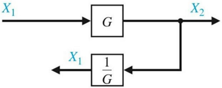
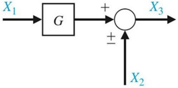
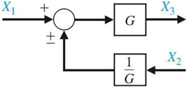
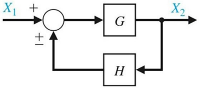
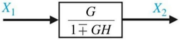
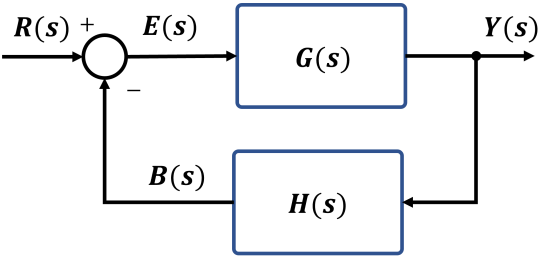


Table 2.5 Block Diagram Transformations

<table>
<tr><th>Transformation</th><th>Original Diagram</th><th>Equivalent Diagram</th></tr>
<tr><td>1. Combining blocks in cascade</td><td></td><td></td></tr>
<tr><td>2. Moving a summing point behind a block</td><td></td><td></td></tr>
<tr><td>3. Moving a pickoff point ahead of a block</td><td></td><td></td></tr>
<tr><td>4. Moving a pickoff point behind a block</td><td></td><td></td></tr>
<tr><td>5. Moving a summing point ahead of a block</td><td></td><td></td></tr>
<tr><td>6. Eliminating a feedback loop</td><td></td><td></td></tr>
</table>

Feedback Control System without Disturbances

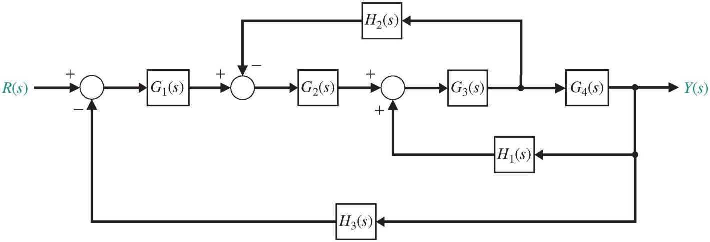


- Forward Path Transfer Function (FPTF): $\frac{Y(s)}{E(s)} = G(s)$
- Error: $E(s) = R(s) - B(s) = R(s) - H(s)Y(s)$
- Closed-Loop Transfer Function:

$$\frac{Y(s)}{R(s)} = \frac{\text{FPTF}}{1 \pm (\text{FPTF})(\text{Feedback TF})} = \frac{G(s)}{1 \pm G(s)H(s)}$$

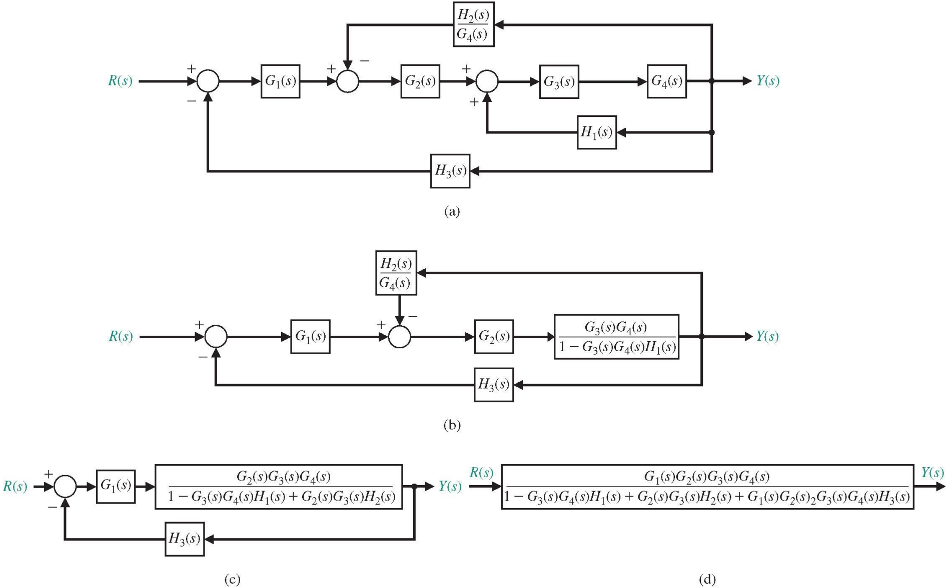
Block Diagram Reduction Example

- Consider a multi-loop feedback control system:



- Apply equivalent transformations to eliminate the feedback loops and obtain the system transfer function $\frac{Y(s)}{R(s)}$

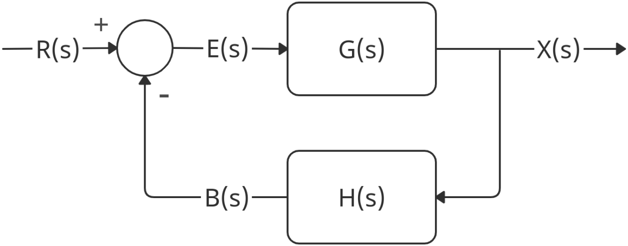
Block Diagram Reduction Example



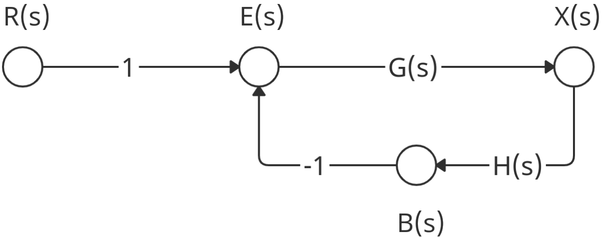
Signal Flow Graph

- **Signal Flow Graph** (SFG): a graphical representation of a control system, consisting of nodes connected by directed branches
- **Node**: a junction point representing a signal variable as the sum of all signals entering the node
- **Branch**: a directed line connecting two nodes with associated transfer function
- **Path**: continuous succession of branches traversed in the same direction
- **Forward Path**: starts at an input node, ends at an output node, and no node is traversed more than once
- **Path Gain**: the product of all branch gains along the path
- **Loop**: a closed path that starts and ends at the same node and no node is traversed more than once
- **Non-touching Loops**: loops that do not contain common nodes

Feedback Control System



(a) Block Diagram



(b) Signal Flow Graph

Mason's Gain Formula

- A method for reducing an SFG to a single transfer function
- The transfer function $T^{ij}(s)$ from **input** $X_i(s)$ to **any** variable $X_j(s)$ is:

$$T^{ij}(s) = \frac{X_j(s)}{X_i(s)} = \frac{\sum_k P_k^{ij}(s)\Delta_k^{ij}(s)}{\Delta(s)}$$

  where:
  - $\Delta(s)$: graph determinant
  - $P_k^{ij}(s)$: gain of the k-th forward path between $X_i(s)$ and $X_j(s)$
  - $\Delta_k^{ij}(s)$: graph determinant with the loops touching the k-th forward path between $X_i(s)$ and $X_j(s)$ removed
- The transfer function $T^{nj}(s)$ from **non-input** $X_n(s)$ to variable $X_j(s)$ is:

$$T^{nj}(s) = \frac{X_j(s)}{X_n(s)} = \frac{X_j(s)/X_i(s)}{X_n(s)/X_i(s)} = \frac{T^{ij}(s)}{T^{in}(s)} = \frac{\sum_k P_k^{ij}(s)\Delta_k^{ij}(s)}{\sum_k P_k^{in}(s)\Delta_k^{in}(s)}$$

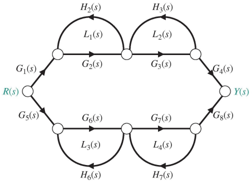
Mason's Gain Formula

- $L_n(s)$: gain of the n-th loop
- $\Delta(s)$: graph determinant

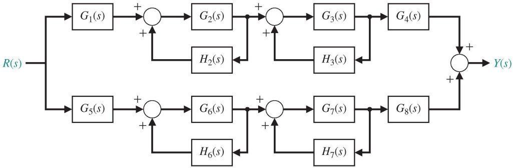
$$\begin{aligned}\Delta(s) = 1 &- \sum(\text{individual loop gains}) \\ &+ \sum\prod(\text{gains of all 2 non-touching loop combinations}) \\ &- \sum\prod(\text{gains of all 3 non-touching loop combinations}) \\ &+ \cdots \\ = 1 &- \sum_{n} L_n(s) + \sum_{\substack{n,m \\ \text{nontouching}}} L_n(s)L_m(s) - \sum_{\substack{n,m,p \\ \text{nontouching}}} L_n(s)L_m(s)L_p(s) + \cdots\end{aligned}$$

- $\Delta_k^{ij}(s)$: graph determinant with the loops touching the k-th forward path between $X_i(s)$ and $X_j(s)$ removed

Mason's Gain Formula Example 1



(a)



(b)

- Determine the transfer function $\frac{Y(s)}{R(s)}$ using Mason's gain formula
- Forward paths from $R(s)$ to $Y(s)$:

$$P_1(s) = G_1(s)G_2(s)G_3(s)G_4(s)$$
$$P_2(s) = G_5(s)G_6(s)G_7(s)G_8(s)$$

- Loop gains:

$$L_1(s) = G_2(s)H_2(s), \qquad L_2(s) = H_3(s)G_3(s),$$
$$L_3(s) = G_6(s)H_6(s), \qquad L_4(s) = G_7(s)H_7(s)$$

Mason's Gain Formula Example 1

- Determinant:

$$\Delta(s) = 1 - (L_1(s) + L_2(s) + L_3(s) + L_4(s)) \\ + (L_1(s)L_3(s) + L_1(s)L_4(s) + L_2(s)L_3(s) + L_2(s)L_4(s))$$

- Cofactor of path 1:

$$\Delta_1(s) = 1 - (L_3(s) + L_4(s))$$

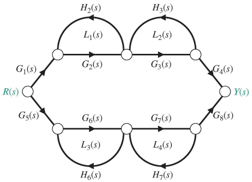
- Cofactor of path 2:

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

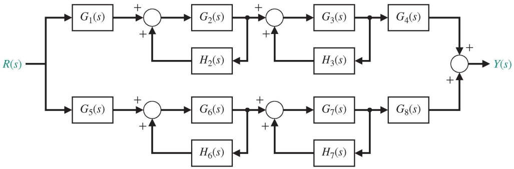
- Transfer function:

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

Mason's Gain Formula Example 1



(a)

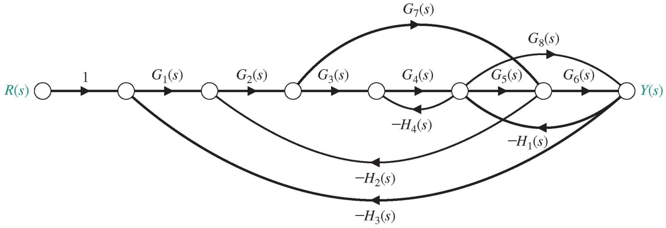


(b)

- The transfer function can also be obtained using block diagram transformations:

$$
\begin{aligned}
T(s) &= G_1(s)\left(\frac{G_2(s)}{1-G_2(s)H_2(s)}\right)\left(\frac{G_3(s)}{1-G_3(s)H_3(s)}\right)G_4(s) \\
&\quad + G_5(s)\left(\frac{G_6(s)}{1-G_6(s)H_6(s)}\right)\left(\frac{G_7(s)}{1-G_7(s)H_7(s)}\right)G_8(s) \\
&= G_1(s)G_2(s)G_3(s)G_4(s)\frac{\Delta_1(s)}{\Delta(s)} + G_5(s)G_6(s)G_7(s)G_8(s)\frac{\Delta_2(s)}{\Delta(s)}
\end{aligned}
$$

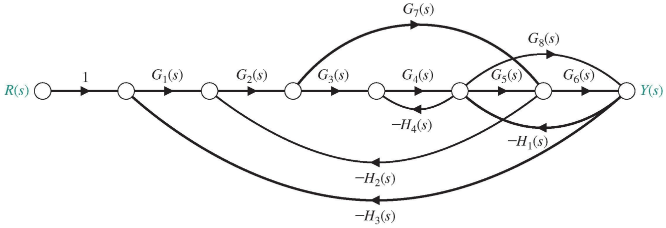
Mason's Gain Formula Example 2



- Determine the transfer function $\frac{Y(s)}{R(s)}$ using Mason's gain formula
- Forward paths from $R(s)$ to $Y(s)$:

$$P_1(s) = G_1(s)G_2(s)G_3(s)G_4(s)G_5(s)G_6(s)$$
$$P_2(s) = G_1(s)G_2(s)G_7(s)G_6(s)$$
$$P_3(s) = G_1(s)G_2(s)G_3(s)G_4(s)G_8(s)$$

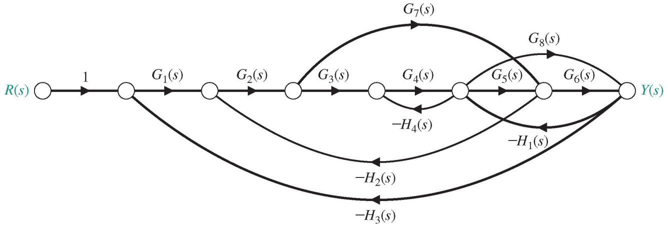
Mason's Gain Formula Example 2



- Loop gains:

$$L_1(s) = -G_2(s)G_3(s)G_4(s)G_5(s)H_2(s), \qquad L_2(s) = -G_5(s)G_6(s)H_1(s),$$
$$L_3(s) = -G_8(s)H_1(s), \qquad L_4(s) = -G_7(s)H_2(s)G_2(s)$$
$$L_5(s) = -G_4(s)H_4(s), \qquad L_6(s) = -G_1(s)G_2(s)G_3(s)G_4(s)G_5(s)G_6(s)H_3(s)$$
$$L_7(s) = -G_1(s)G_2(s)G_7(s)G_6(s)H_3(s), \qquad L_8(s) = -G_1(s)G_2(s)G_3(s)G_4(s)G_8(s)H_3(s)$$

Mason's Gain Formula Example 2



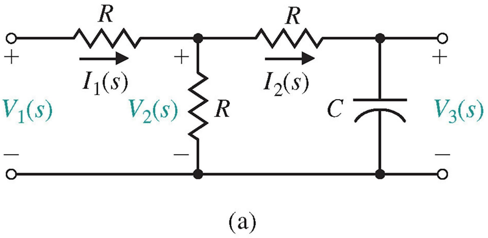
- Cofactors: $\Delta_1(s) = \Delta_3(s) = 1$ and $\Delta_2(s) = 1 - L_5(s)$
- Determinant: L_5 does not touch L_4 or L_7 and L_3 does not touch L_4:

$$\Delta(s) = 1 - (L_1(s) + L_2(s) + L_3(s) + L_4(s) + L_5(s) + L_6(s) + L_7(s) + L_8(s)) \\ + (L_5(s)L_4(s) + L_5(s)L_7(s) + L_3(s)L_4(s))$$

- Transfer function:

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

Mason's Gain Formula Example 3



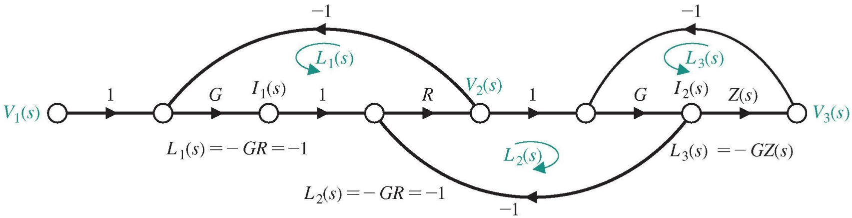
(a)

- Consider a ladder circuit with one energy storage element
- Determine the transfer function from $V_1(s)$ to $V_3(s)$
- The current and voltage equations are:

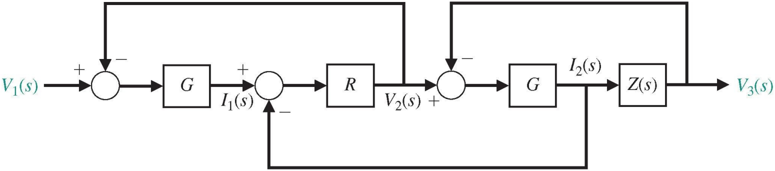
$$I_1(s) = \frac{1}{R}(V_1(s) - V_2(s)) \qquad I_2(s) = \frac{1}{R}(V_2(s) - V_3(s))$$

$$V_2(s) = R(I_1(s) - I_2(s)) \qquad V_3(s) = \frac{1}{Cs}I_2(s)$$

Mason's Gain Formula Example 3



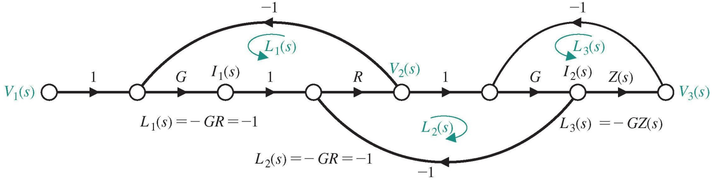
(b)



(c)

- Admittance: $G = \frac{1}{R}$
- Impedence: $Z(s) = \frac{1}{Cs}$

Mason's Gain Formula Example 3



(b)

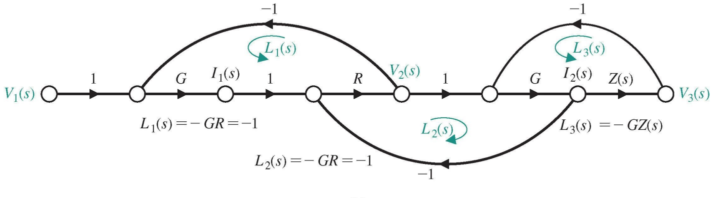
- Forward path: $P_1(s) = GRGZ(s) = GZ(s) = \frac{1}{RCs}$
- Loops: $L_1(s) = -GR = -1$, $L_2(s) = -GR = -1$, $L_3(s) = -GZ(s)$
- Cofactor: all loops touch the forward path: $\Delta_1(s) = 1$
- Determinant: loops $L_1(s)$ and $L_3(s)$ are non-touching:

$$\Delta(s) = 1 - (L_1(s) + L_2(s) + L_3(s)) + L_1(s)L_3(s) = 3 + 2GZ(s)$$

- Transfer function:

$$T(s) = \frac{V_3(s)}{V_1(s)} = \frac{P_1(s)}{\Delta(s)} = \frac{GZ(s)}{3 + 2GZ(s)} = \frac{1/(3RC)}{s + 2/(3RC)}$$

Mason's Gain Formula Example 3



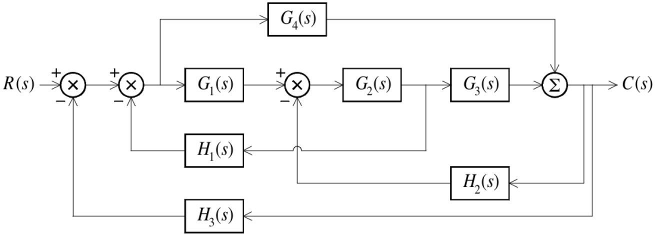
(b)

- Determine the transfer function from $I_1(s)$ to $I_2(s)$
- Instead of re-drawing the signal flow graph, we can use:

$$\frac{I_2(s)}{I_1(s)} = \frac{I_2(s)/V_1(s)}{I_1(s)/V_1(s)} = \frac{G}{G(2+GZ(s))} = \frac{1}{2+GZ(s)} = \frac{s}{2s+1/(RC)}$$

- One forward path from $V_1(s)$ to $I_2(s)$ with gain $GRG = G$ and cofactor 1
- One forward path from $V_1(s)$ to $I_1(s)$ with gain G and cofactor $1-(L_2(s)+L_3(s)) = 2+GZ(s)$

Mason's Gain Formula Example 4



- Determine the transfer function from $R(s)$ to $C(s)$
- Forward paths:

$$P_1(s) = G_1(s)G_2(s)G_3(s) \qquad P_2(s) = G_4(s)$$

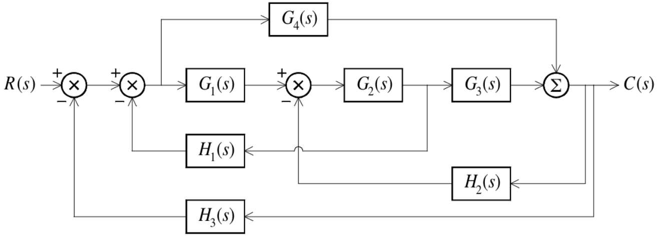
- Loops:

$$L_1(s) = -G_1(s)G_2(s)H_1(s) \qquad L_2(s) = -G_2(s)G_3(s)H_2(s)$$

$$L_3(s) = -G_1(s)G_2(s)G_3(s)H_3(s) \qquad L_4(s) = -G_4(s)H_3(s)$$

$$L_5(s) = G_2(s)H_1(s)G_4(s)H_2(s)$$

Mason's Gain Formula Example 4



- Cofactors: both forward paths touch all loops: $\Delta_1(s) = \Delta_2(s) = 1$
- Determinant: all loop pairs are touching:

$$\Delta(s) = 1 - (L_1(s) + L_2(s) + L_3(s) + L_4(s) + L_5(s))$$

- Transfer function:

$$T(s) = \frac{C(s)}{R(s)} = \frac{P_1(s) + P_2(s)}{\Delta(s)} = \frac{G_1(s)G_2(s)G_3(s) + G_4(s)}{\Delta(s)}$$

MATLAB Polynomial Functions

- Consider:

$$p(s) = (s - 11.6219)(s + 0.3110 + 2.6704j)(s + 0.3110 - 2.6704j)$$

- poly: convert roots to polynomial coefficients:

```
r = [11.6219, -0.3110-2.6704i, -0.3110+2.6704i]
a = poly(r) = [1.0, -11.0, 0.0, -84.0]
```

- polyval: evaluate a polynomial, e.g., $p(1-2j)$:

```
polyval(a, 1-2i) = -62 + 46i
```

- roots: find polynomial roots:

```
roots(a) = [11.6219, -0.3110-2.6704i, -0.3110+2.6704i]
```

- conv: expand the product of two polynomials, e.g., $(3s^2+2s+1)(s+4)$:

```
conv([3, 2, 1], [1, 4]) = [3, 14, 9, 4]
```

MATLAB Control System Functions

- SYS = tf(NUM,DEN): creates a continuous-time transfer function SYS with numerator NUM and denominator DEN:

```
dcmotor = tf(200,[1 1]);
```

- SYS = series(SYS1,SYS2): series connection of SYS1 and SYS2:

```
fwdsys = series(tf(200,[1 1]), tf(1,[1 8]));
```

- SYS = parallel(SYS1,SYS2): parallel connection of SYS1 and SYS2

```
fwdsys = parallel(tf(200,[1 1]), tf(1,[1 8]));
```

- SYS = feedback(SYS1, SYS2, sign): feedback connection of SYS1 and SYS2:

```
fbksys = feedback(series(tf(200,[1 1]), tf(1,[1 8])),tf(1,[0.25 1]))
```

MATLAB Control System Functions

- SYS = zpk(Z,P,K) creates a continuous-time zero-pole-gain (zpk) model SYS with zeros Z, poles P, and gains K:

```
dcmotor = zpk([],[-1],200);
fbksys = zpk([-4],[-8.8426, -2.0787 + 1.7078i, -2.0787 -1.7078i],8);
```

- P = pole(SYS) returns the poles P of SYS:

```
sp = pole(fbksys) = [-8.8426, -2.0787 + 1.7078i, -2.0787 -1.7078i]
```

- [Z,G] = zero(SYS) computes the zeros Z and gain G of SYS:

```
[sz,k] = zero(fbksys) = [-4, 8]
```

- pzmap(SYS): computes and plots the poles and zeros of SYS

```
pzmap(fbksys)
```

MATLAB Control System Functions

- Y = step(SYS,T): computes the step response Y of SYS at times T

```
t = 0:0.01:5;
step(fbksys,t);
```

- Y = impulse(SYS, T): computes the impulse response Y of SYS at times T

```
t = 0:0.01:5;
impulse(fbksys,t);
```

- Y = lsim(SYS,U,T): computes the output response Y of SYS with input U at times T

```
[u,t] = gensig('square',4,10,0.1);
lsim(fbksys,u,t);
```