State-Space & Canonical Forms

ECE 461/661 Controls Systems

Jake Glower - Lecture #13

Please visit Bison Academy for corresponding lecture notes, homework sets, and solutions

State-Space & Canonical Forms

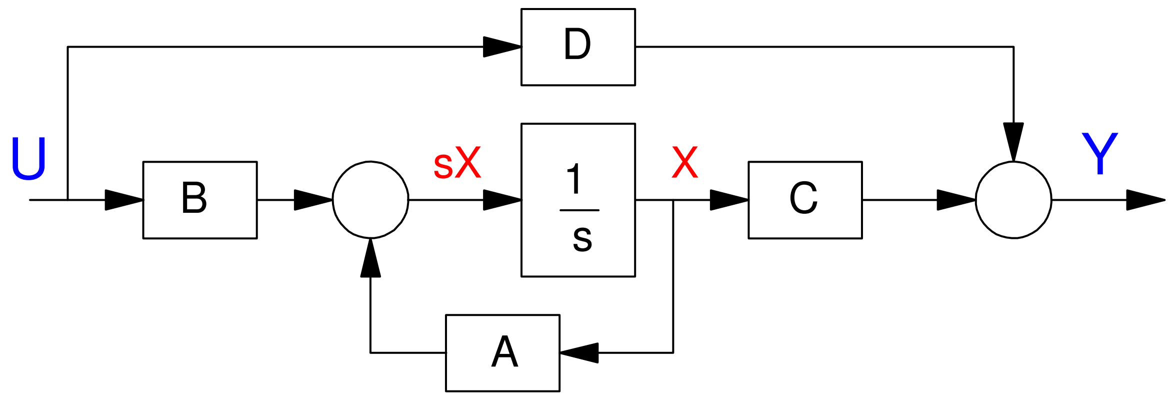
State-Space:

- Matrix-based way to describe a system
- How Matlab actually stores a system

Standard Form:

$$sX = AX + BU$$

$$Y = CX + DU$$



Transfer Function

$$Y = \left(C(sI - A)^{-1} B + D \right) U$$

MATLAB Commands

• G = ss(A, B, C, D);	input a system in state-space form
• G = tf(num, den)	input a system in transfer function form
• G = zpk(z, p, k)	input a system in zeros, poles, gain form
• ss(G)	determine A, B, C, D for system G (answer is not unique)
• tf(G)	determine the transfer function of system G
• zpk(G)	determine the zeros, poles, and gain of system G

Matrix Algebra

An nxm matrix has n rows and m columns.

$$A_{2x3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

Scalar Multiplication:

$$bA = \begin{bmatrix} ba_{11} & ba_{12} & ba_{13} \\ ba_{21} & ba_{22} & ba_{23} \end{bmatrix}$$

Matrix Addition:

- Matrices with the same dimensions can be added:

$$A + B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + a_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \end{bmatrix}$$

Matrix Multiplication:

- Inner dimensions must match
- $A_{xy} \cdot B_{yz} = C_{xz}$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix}$$

$$= \begin{bmatrix} \sum a_{1x}b_{x1} & \sum a_{1x}b_{x2} \\ \sum a_{2x}b_{x1} & \sum a_{2x}b_{x2} \end{bmatrix}$$

Matrix Inverse

$$A \cdot A^{-1} = I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For a 2x2 matrix,

$$A^{-1} = \begin{bmatrix} \frac{a_{22}}{\Delta} & \frac{-a_{12}}{\Delta} \\ \frac{-a_{21}}{\Delta} & \frac{a_{11}}{\Delta} \end{bmatrix} \qquad \Delta = a_{11} a_{22} - a_{12} a_{22}$$

Placing a System in State-Space Form:

i) Write N equations for the N voltage nodes

ii) Solve for the highest derivative for each equation

iii) Rewrite in matrix form.

$$\frac{dx_1}{dt} = x_2 - x_1 + 0.3u$$

$$\frac{dx_2}{dt} = x_1 - 1.3x_2$$

$$y = x_2$$

$$s\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1.3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]U$$

To find the transfer function in MATLAB,

```
A = [-1, 1; 1, -1.3]
B = [0.3; 0]
C = [0, 1]
D = 0;

G = ss(A, B, C, D);
tf(G)

       0.3
-----------------
s^2 + 2.3 s + 0.3
```

Example: Find the transfer function from U to X4.

$$sx_1 = u - 2x_1 + x_2$$

$$sx_2 = x_1 - 2x_2 + x_3$$

$$sx_3 = x_2 - 2x_3 + x_2$$

$$sx_4 = x_3 - x_4$$

State-Space

$$s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]U$$

Input this into MATLAB

```
A = [-2,1,0,0;1,-2,1,0;0,1,-2,1;0,0,1,-1];
B = [1;0;0;0];
C = [0,0,0,1];
D = 0;

G = ss(A,B,C,D);
tf(G)

               1
-------------------------------
s^4 + 7 s^3 + 15 s^2 + 10 s + 1


zpk(G)

                 1
------------------------------------
(s+1) (s+2.347) (s+3.532) (s+0.1206)
```

Changing the output:

- Average temperature of the bar
- Poles stay the same
- Zeros change

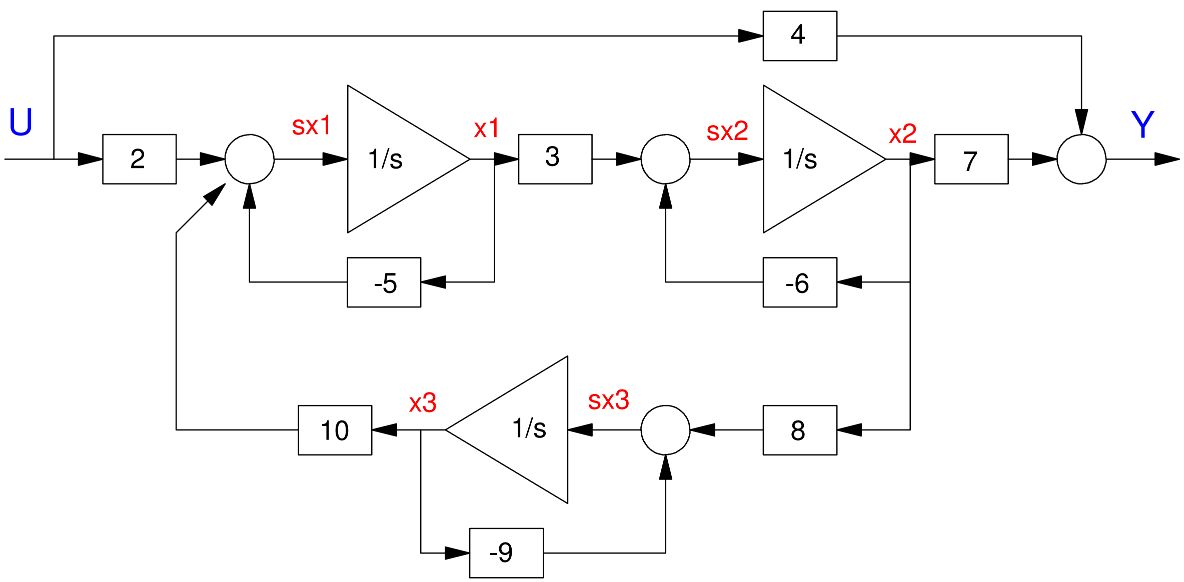
$$y = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]U$$

```
C = [0.25,0.25,0.25,0.25];
D = 0;

G = ss(A,B,C,D);
zpk(G)

  0.25 (s+3.414) (s+2) (s+0.5858)
------------------------------------
(s+1) (s+2.347) (s+3.532) (s+0.1206)
```

Handout: Express the dynamics in state-space form:

$$s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \_\_ \\ \_\_ \\ \_\_ \end{bmatrix} U$$

$$Y = \begin{bmatrix} \_\_ & \_\_ & \_\_ \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [\_\_]U$$



Canonical Forms

- 4th-Order system has 9 constraints

$$Y = \left(\frac{c_4 s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$$

- State-Space has 25 degrees of freedom

$$sX_{4x1} = A_{4x4} X_{4x1} + B_{4x1} U$$

$$Y = C_{1x4} X_{4x1} + D_{1x1} U$$

There are an infinite number of ways to express a system in state-space

- Some have names (canonical forms)
- Many do not

Controller Canonical Form:

Change the problem to

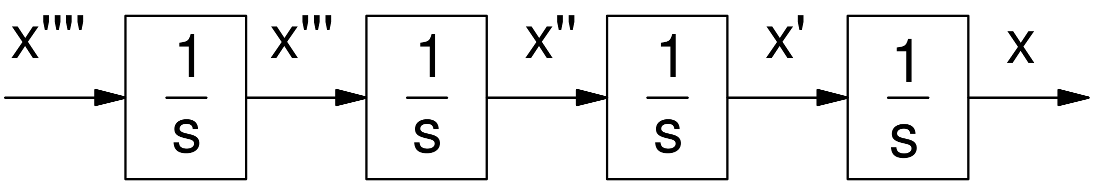
$$X = \left(\frac{1}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}\right) U$$

$$Y = (c_3 s^3 + c_2 s^2 + c_1 s + c_0) X$$

Solve for the highest derivative of X:

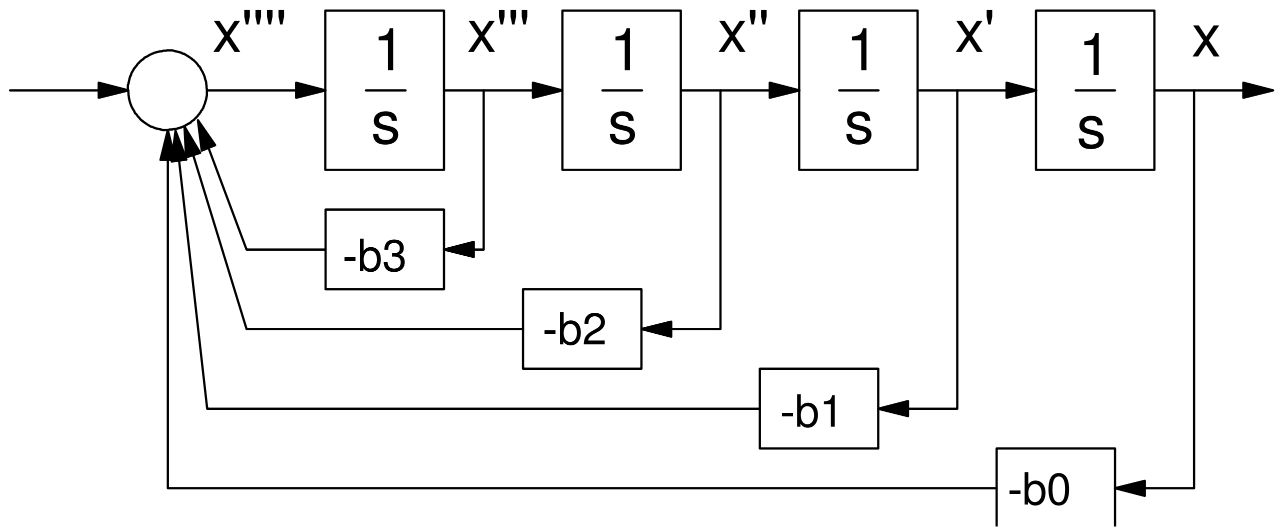
$$s^4 X = U - b_3 s^3 X + b_2 s^2 X + b_1 s X + b_0 X$$

Integrate s^4X four times to get X:



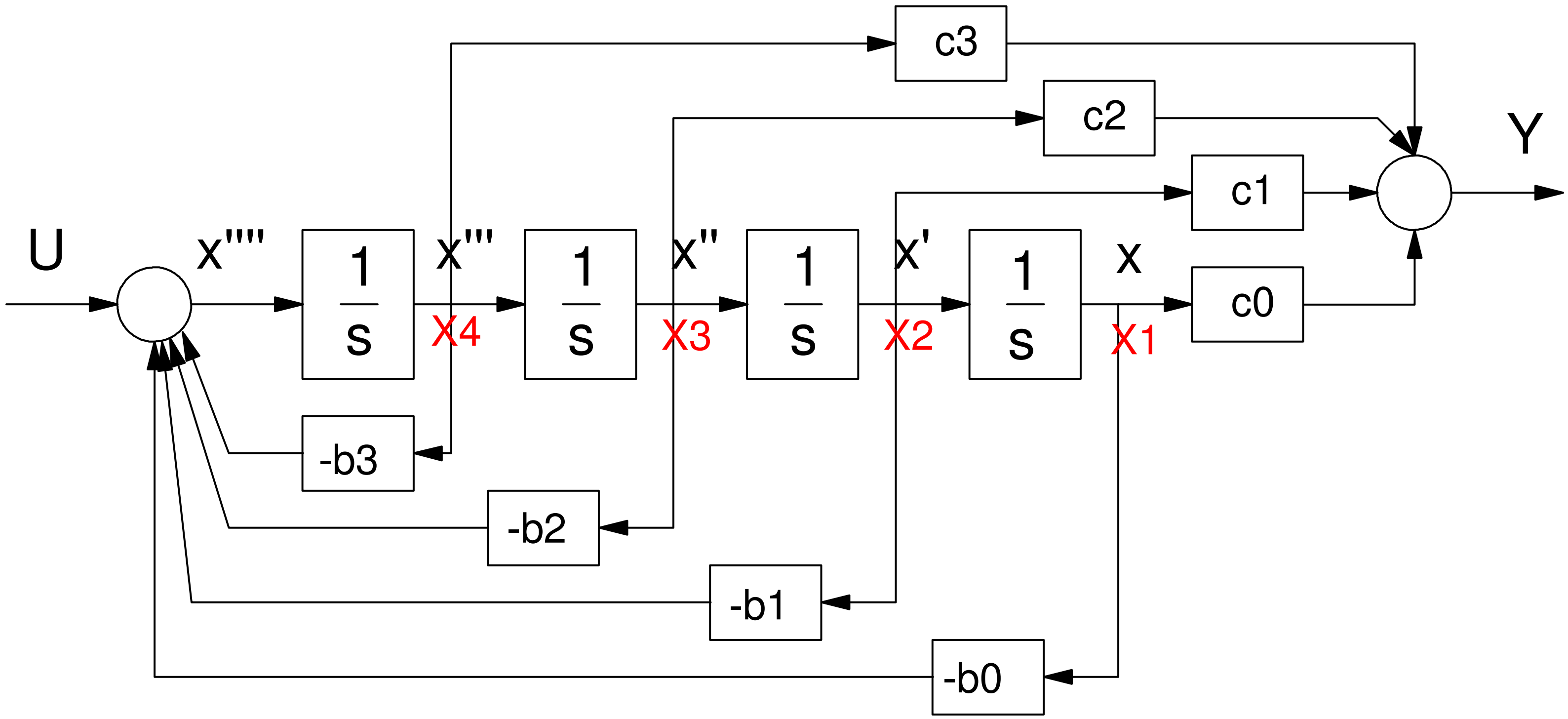
Create s4X from U and its derivatives

$$s^4X = U - b_3s^3X + b_2s^2X + b_1sX + b_0X$$



Create Y from the derivatives of X:

$$Y = (c_3 s^3 + c_2 s^2 + c_1 s + c_0)X$$



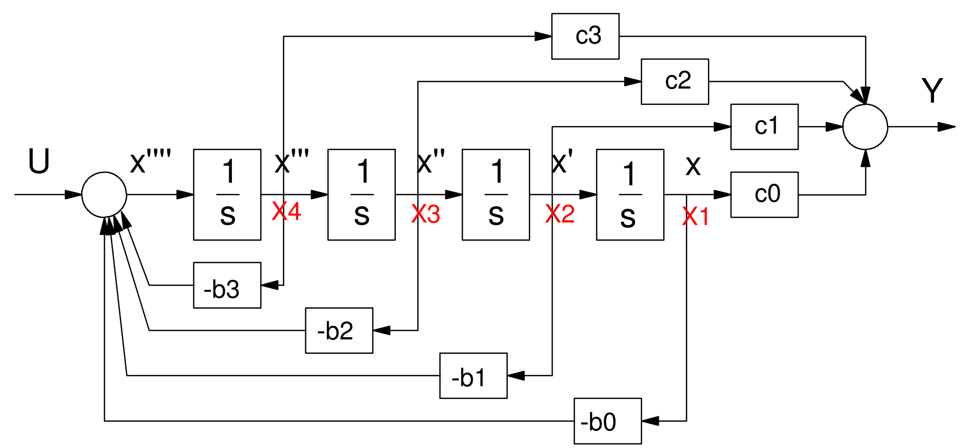
Controller Canonical Form

Controller Canonical Form in State Space

$$Y = \left(\frac{c_3 s^3 + c_2 s^2 + c_1 s + c_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$$

$$s \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -b_0 & -b_1 & -b_2 & -b_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} U$$

$$Y = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0] U$$

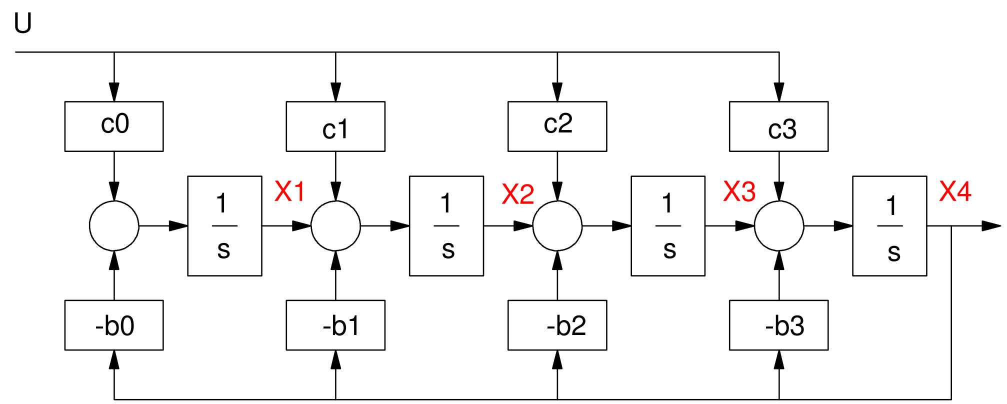


Observer Canonical Form:

$$\left(C(sI-A)^{-1}B\right)^T = B^T(sI-A^T)^{-1}C^T$$

$$s\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -b_0 \\ 1 & 0 & 0 & -b_1 \\ 0 & 1 & 0 & -b_2 \\ 0 & 0 & 1 & -b_3 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}U$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$



Cascade Form:

Factor the denominator

$$Y = \left(\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{(s+p_1)(s+p_2)(s+p_3)(s+p_4)}\right) U$$

Create 4 dummy states

$$X_1 = \left(\frac{1}{s+p_1}\right) U \qquad X_2 = \left(\frac{1}{s+p_2}\right) X_1$$

$$X_3 = \left(\frac{1}{s+p_3}\right) X_2 \qquad X_4 = \left(\frac{1}{s+p_4}\right) X_3$$

The output is then

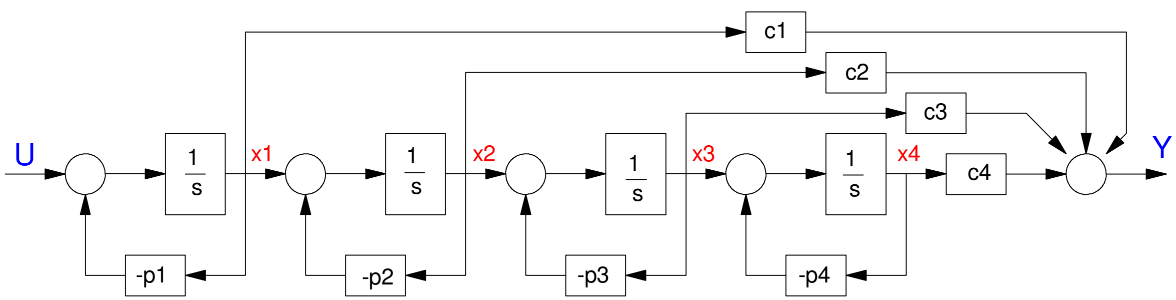
$$y = c_1 X_1 + c_2 X_2 + c_3 X_3 + c_4 X_4$$

Cascade Form:

$$Y = \left(\frac{c_4 + c_3(s+p_4) + c_2(s+p_4)(s+p_3) + c_1(s+p_4)(s+p_3)(s+p_2)}{(s+p_1)(s+p_2)(s+p_3)(s+p_4)} \right) U$$

$$s\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 & 0 \\ 1 & -p_2 & 0 & 0 \\ 0 & 1 & -p_3 & 0 \\ 0 & 0 & 1 & -p_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} c_1 & c_2 & c_3 & c_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$



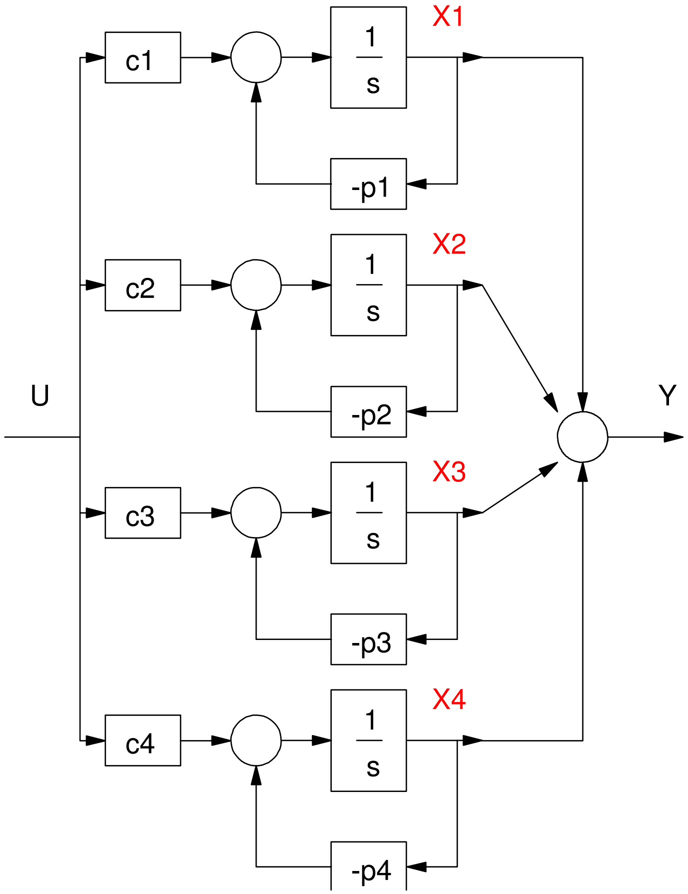
Jordan Form:

Use partial fraction expansion

$$Y = \left(\left(\frac{c_1}{s+p_1}\right) + \left(\frac{c_2}{s+p_2}\right) + \left(\frac{c_3}{s+p_3}\right) + \left(\frac{c_4}{s+p_4}\right)\right) U$$

$$s\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 & 0 \\ 0 & -p_2 & 0 & 0 \\ 0 & 0 & -p_3 & 0 \\ 0 & 0 & 0 & -p_4 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$



Summary

State-space is a way to express the dynamics using matrices

- It's how Matlab stores dynamic systems

There are an infinite number of solutions

- Pick your favorite canonical form

If the numerator has 's' terms, you don't have to take derivatives

- Numerator terms show up in the B and/or C matrices

Matlab is useful for going from state-space to transfer function form