Canonical Forms and Similarity Transforms

NDSU ECE 463/663

Lecture #5

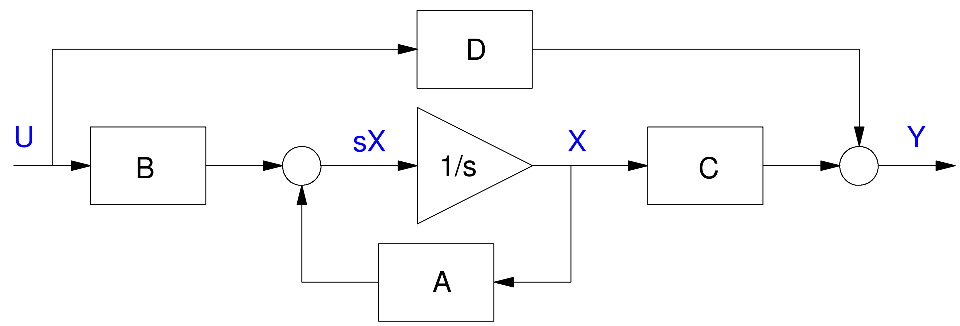
Inst: Jake Glower

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

Canonical Forms

Problem: Represent the transfer function

$$Y = \left(\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$$



in state-space form

$$sX = AX + BU$$

$$Y = CX + DU$$

{A, B, C, D} has 25 degrees of freedom

The transfer function privides 8 constraints

- There are an infinte number of solutions
- Some of these have names (canonical forms)
- Many do not

Controller Canonical Form

Define a dummy variable, X

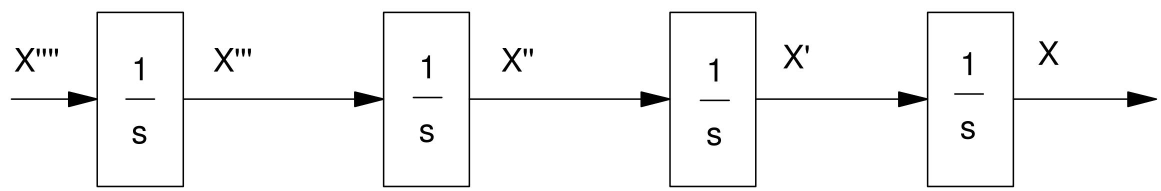
$$X = \left(\frac{1}{s^4+b_3s^3+b_2s^2+b_1s+b_0}\right)U$$

$$Y = (a_3s^3 + a_2s^2 + a_1s + a_0)X$$

Solve for the highest derivative of X

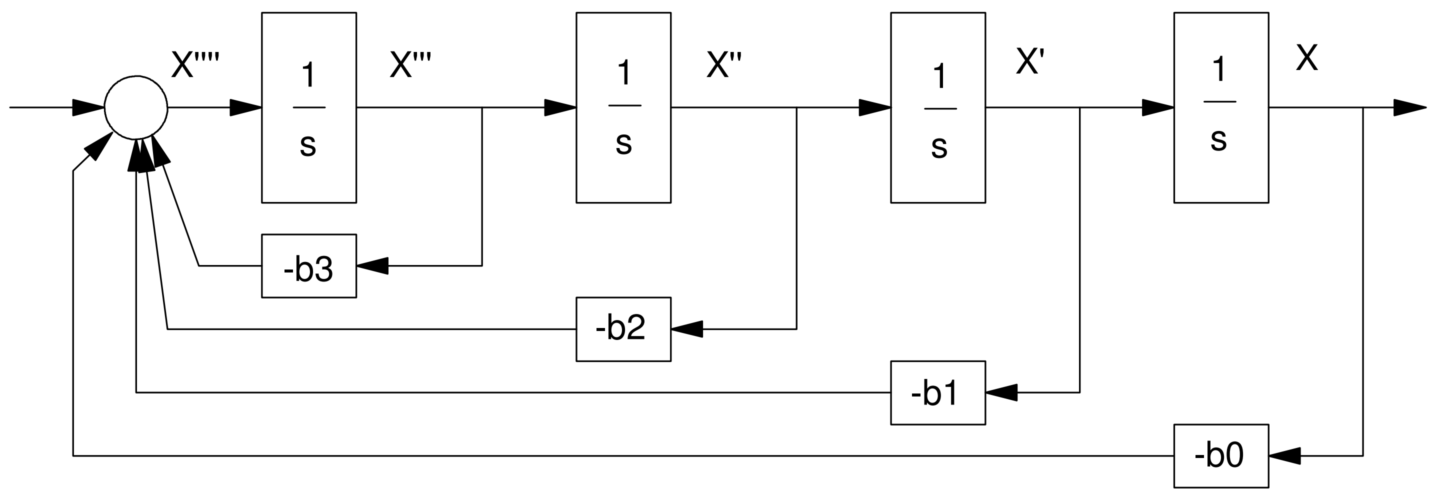
$$s^4X = U - b_3s^3X - b_2s^2X - b_1sX - b_0X$$

Given the 4th derivative of X, integrate four times to get X



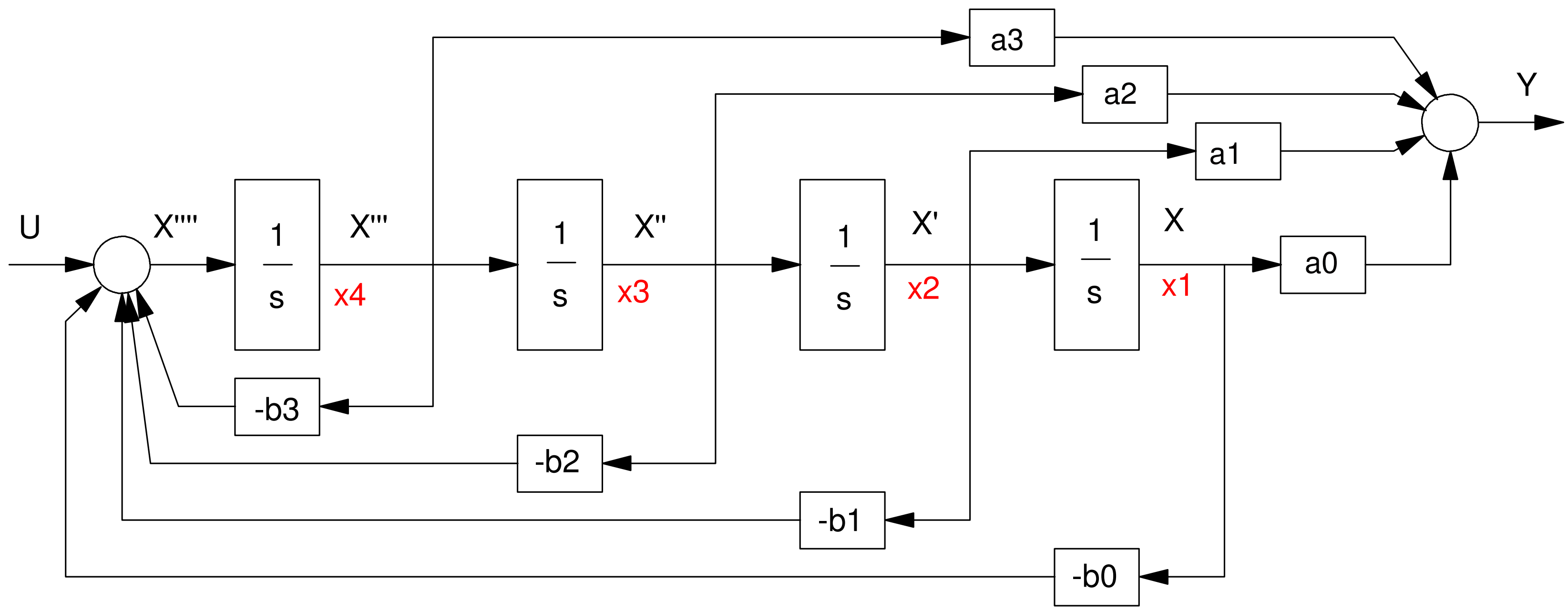
Generate X''' according to the dynamics

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



Now that you have X and its derivatives, create Y

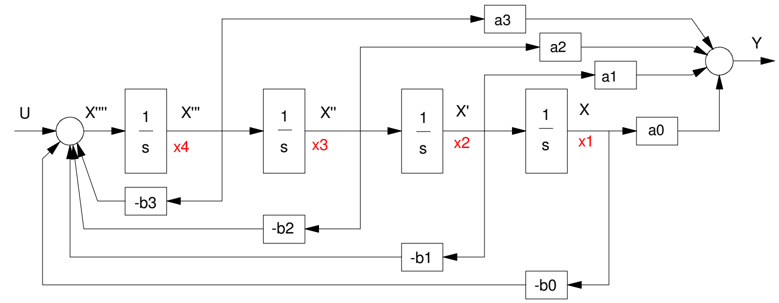
$$Y = (a_3 s^3 + a_2 s^2 + a_1 s + a_0)X$$



Controller Canonical Form for $Y = \left(\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}\right) U$

Express in matrix form

$$s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -b_0 & -b_1 & -b_2 & -b_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}U$$



$$Y = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} X + [0]U$$

Controller canonical form has some nice properties:

- The transfer function can be found by inspection: the numerator and denominator polynomials appear in the A and C matrices
- You can control (set to any value) all of the states with input, U.

Controller canonical form also has some of the *worst* numerical properties.

Observer Canonical Form

Given a system

$$sX = AX + BU$$

$$Y = CX + DU$$

the transfer function from U to Y is

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

For a single-input single-output (SISO) system this is also

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

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

Another perfectly valid representation for a system is to let

$$A^T \to A$$

$$B^T \to C$$

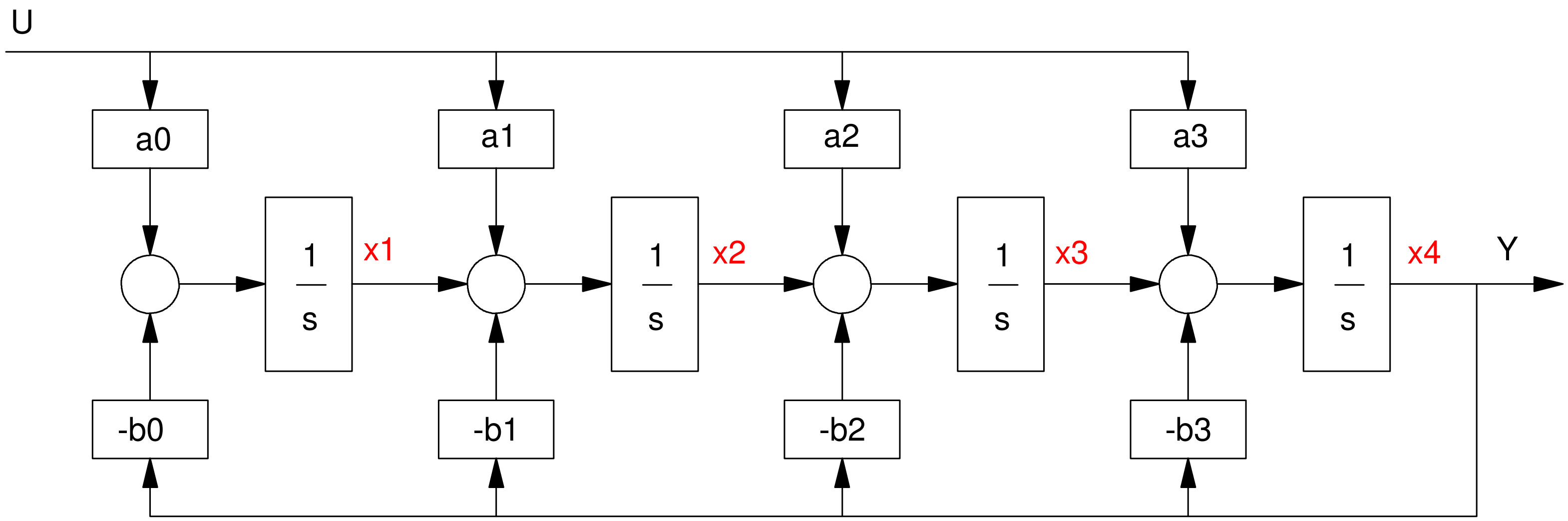
$$C^T \to B$$

For example, the 4th-order system from before becomes

$$s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -b_0 \\ 1 & 0 & 0 & -b_1 \\ 0 & 1 & 0 & -b_2 \\ 0 & 0 & 1 & -b_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} U$$

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

This is called *observer canonical form:* from the output (Y) you can determine all of the states through differentiation. The block-diagram representation for this system is:



Observer Canonical form for $Y = \left(\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$

Cascade Form

If you have real poles, you can write the transfer function as

$$Y = \left(\frac{a_4 + a_3(s+p_4) + a_2(s+p_3)(s+p_4) + a_1(s+p_2)(s+p_3)(s+p_4)}{(s+p_1)(s+p_2)(s+p_3)(s+p_4)}\right) U$$

For this system, you could write it as four cascaded 1st-order systems

$$x_1 = \left(\frac{1}{s+p_1}\right) U$$

$$x_2 = \left(\frac{1}{s+p_2}\right) X_1$$

$$x_3 = \left(\frac{1}{s+p_3}\right) X_2$$

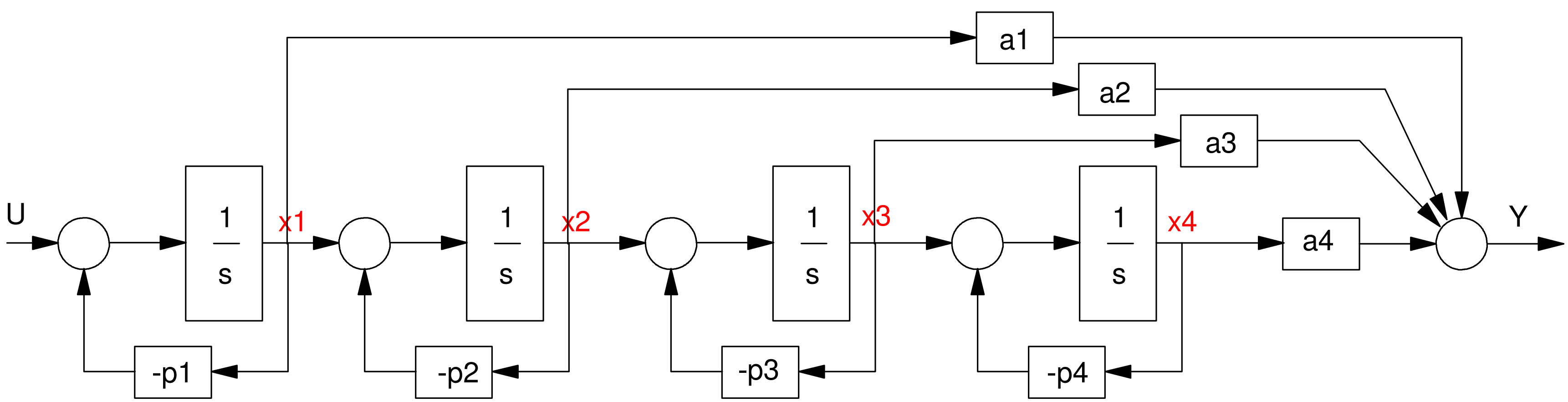
$$x_4 = \left(\frac{1}{s+p_4}\right) X_3$$

$$Y = a_4 x_4 + a_3 x_3 + a_2 x_2 + a_1 x_1$$

The state-space model is

$$s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 & 0 \\ 1 & -p_2 & 0 & 0 \\ 0 & 1 & -p_3 & 0 \\ 0 & 0 & 1 & -p_4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} X$$



Jordan (Diagonal) Canonical Form

Use partial fraction expansion to express

$$Y = \left(\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$$

as

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

Treat this as four coupled systems

$$x_1 = \left(\frac{c_1}{s+p_1} \right) U \qquad x_2 = \left(\frac{c_2}{s+p_2} \right) U$$

$$x_3 = \left(\frac{c_3}{s+p_3} \right) U \qquad x_4 = \left(\frac{c_4}{s+p_4} \right) U$$

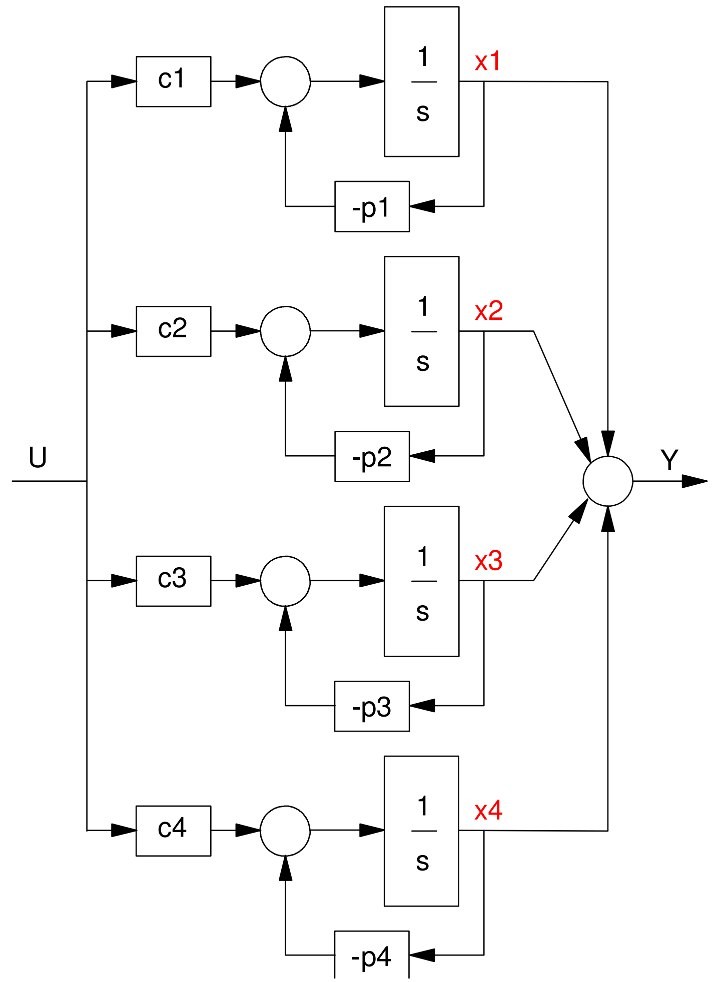
with

$$Y = x_1 + x_2 + x_3 + x_4$$

In state-space

$$s\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 & 0 \\ 0 & -p_2 & 0 & 0 \\ 0 & 0 & -p_3 & 0 \\ 0 & 0 & 0 & -p_4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} X$$



Note:

- Cascade and Jordan form have the best numerical properties
- They're also the hardest to get to

Recap:

In state-space, a dynamic system is written as

$$sX = AX + BU$$

$$Y = CX + DU$$

with the transfer function from U to Y being

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

{A, B, C, D} can be expressed several ways:

- Controller canonical form
- Observer canonical form
- Cascade form
- Jordan form

What is the relationship between each of these forms?

Similarity Transforms:

Let Z be a change of variable defined as

$$X = TZ$$

or

$$Z = T^{-1}X$$

where T is an NxN non-singular matrix called the *similarity transform*.

Example:

$$X = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \qquad Z = \begin{bmatrix} V_1 + V_2 \\ V_2 + V_3 \\ V_1 + V_2 + V_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} X$$

Substitute

$$sTZ = ATZ + BU$$

$$Y = CTZ + DU$$

or

$$sZ = T^{-1}ATZ + T^{-1}BU = A_z Z + B_z U$$

$$Y = CTZ + DU = C_z Z + D_z U$$

{A, B, C, D} is related to {Az, Bz, Cz, Dz} as

$$A_z = T^{-1}AT$$

$$B_z = T^{-1}B$$

$$C_z = CT$$

$$D_z = D$$

Different canonical forms are related through a change of variable

- *i.e. through a similarity transform, T*

Case 1: Converting to and from Jordan Form

This is the easiest transform. Almost by definition, the transformation matrix is the Eigenvector matrix

For example, convert the following system to Jordan form:

$$sX = \begin{bmatrix} -2.1 & 1 & 0 & 0 \\ 1 & -2.1 & 1 & 0 \\ 0 & 1 & -2.1 & 1 \\ 0 & 0 & 1 & -1.1 \end{bmatrix} X + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

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

In Matlab:

```
>> A = [-2.1,1,0,0;1,-2.1,1,0;0,1,-2.1,1;0,0,1,-1.1]

   -2.1000    1.0000         0         0
    1.0000   -2.1000    1.0000         0
         0    1.0000   -2.1000    1.0000
         0         0    1.0000   -1.1000

>> B = [1;0;0;0]

     1
     0
     0
     0

>> C = [0,0,0,1]

     0     0     0     1

>> D = 0;
>> [M,N] = eig(A)

M = eigenvectors

   -0.4285   -0.6565    0.5774    0.2280
    0.6565    0.2280    0.5774    0.4285
   -0.5774    0.5774   -0.0000    0.5774
    0.2280   -0.4285   -0.5774    0.6565
```

The similarity transform, T, is simply the eigenvector matrix:

```
>> T = M;
```

The system in state-variable Z becomes:

```
>> Az = inv(T)*A*T

   -3.6320         0         0         0
         0   -2.4470         0         0
         0         0   -1.1000         0
         0         0         0   -0.2210

>> Bz = inv(T)*B

   -0.4285
   -0.6565
    0.5774
    0.2280

>> Cz = C*T

    0.2280   -0.4285   -0.5774    0.6565

>> Dz = D

     0
```

This is Jordan form

$$sZ = \begin{bmatrix} -3.632 & & & \\ & -2.4470 & & \\ & & -1.1 & \\ & & & -0.22 \end{bmatrix} Z + \begin{bmatrix} -0.4285 \\ -0.6565 \\ 0.5774 \\ 0.2280 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0.2280 & -0.4285 & -0.5774 & 0.6565 \end{bmatrix} Z + [0] U$$

Note that the transfer function doesn't change:

```
>> Gx = ss(A,B,C,D);

>> zpk(Gx)

                  1
--------------------------------------
(s+3.632) (s+2.447) (s+1.1) (s+0.2206)


>> Gz = ss(Az,Bz,Cz,Dz);

>> zpk(Gz)

                  1
--------------------------------------
(s+3.632) (s+2.447) (s+1.1) (s+0.2206)
```

Case 2: Converting to Output and its Derivatives

Let Z be the output and its derivatives

$$Z = \begin{bmatrix} y \\ y' \\ y'' \\ y''' \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} X = T^{-1} X$$

then

```
>> T = inv([C; C*A; C*A*A;  C*A*A*A])

    1.6510    7.0300    5.3000    1.0000
    1.3100    3.2000    1.0000         0
    1.1000    1.0000         0         0
    1.0000         0         0         0
```

{Az, Bz, Cz, Dz} become:

```
>> Az = inv(T)*A*T

         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
   -2.1570  -13.2140  -17.1600   -7.4000

>> Bz = inv(T)*B

         0
    0.0000
         0
    1.0000

>> Cz = C*T

     1     0     0     0

>> Dz = D

     0
```

Note again that the eigenvalues don't change with a similarity transform

```
>> eig(A)'

   -3.6321   -2.4473   -1.1000   -0.2206

>> eig(Az)'

   -3.6321   -2.4473   -1.1000   -0.2206
```

nor does the transfer function

```
>> Gx = ss(A,B,C,D);
>> zpk(Gx)

                  1
--------------------------------------
(s+3.632) (s+2.447) (s+1.1) (s+0.2206)


>> Gz = ss(Az,Bz,Cz,Dz);
>> zpk(Gz)

                  1
--------------------------------------
(s+3.632) (s+2.447) (s+1.1) (s+0.2206)
```

Case 3: Converting to a difference in states:

Let the states be

$$Z = \begin{bmatrix} x_1 - x_2 \\ x_2 - x_3 \\ x_3 - x_4 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} X = T^{-1} X$$

In Matlab

```
>> Ti = [1,-1,0,0;0,1,-1,0;0,0,1,-1;0,0,0,1]

     1    -1     0     0
     0     1    -1     0
     0     0     1    -1
     0     0     0     1

>> T = inv(Ti)

     1     1     1     1
     0     1     1     1
     0     0     1     1
     0     0     0     1
```

```
>> Az = inv(T)*A*T

   -3.1         0    -1.0    -1.0
    1.0      -2.1     1.0       0
      0       1.0    -2.1       0
      0         0     1.0    -0.1

>> Bz = inv(T)*B

     1
     0
     0
     0

>> Cz = C*T

     0     0     0     1

>> Dz = D

     0
```

Again, the eigenvalues don't change with a similarity transform

```
>> eig(A)'

   -3.6321   -2.4473   -1.1000   -0.2206

>> eig(Az)'

   -3.6321   -2.4473   -1.1000   -0.2206
```

and the transfer function doesn't change

```
>> Gx = ss(A,B,C,D);
>> zpk(Gx)

                  1
--------------------------------------
(s+3.632) (s+2.447) (s+1.1) (s+0.2206)


>> Gz = ss(Az,Bz,Cz,Dz);
>> zpk(Gz)

                  1
--------------------------------------
(s+3.632) (s+2.447) (s+1.1) (s+0.2206)
```

Conclusion

There are an infinite many ways to represent a system in state-space. All related by a similarity transform.

Each transformed system has the same eigenvalues: how you represent the system doesn't affect how the energy in the system moves about.

It may be difficult to determine what the similarity transform is that relates two similar systems.