Output Feedback

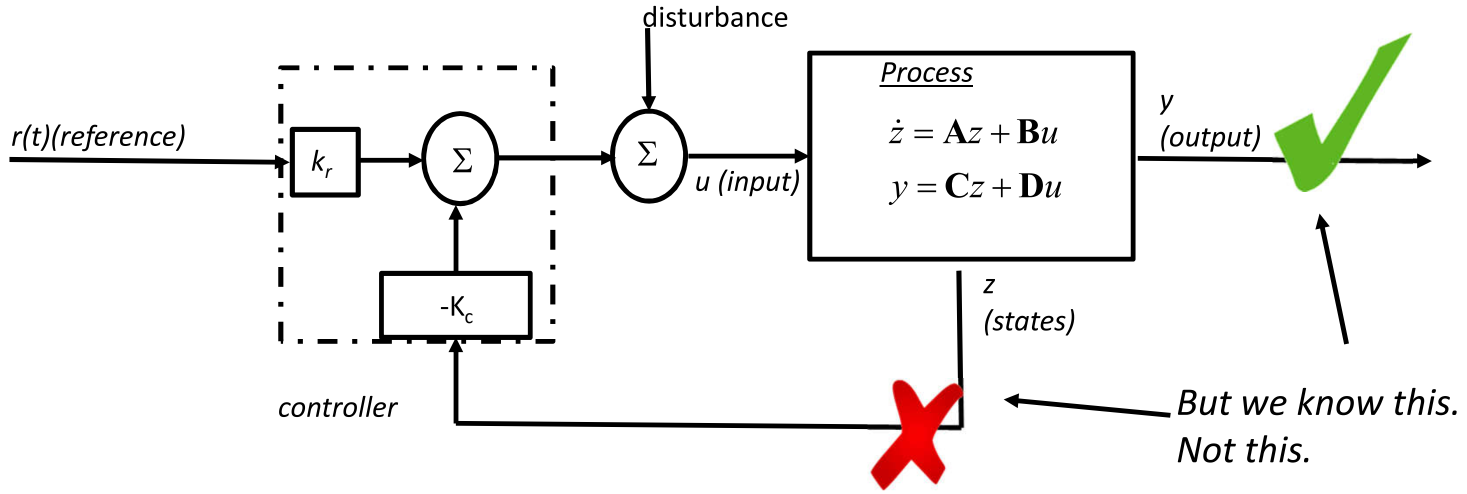
Mitch Pryor

Objective: Use output not states

Given a system with the following dynamic model and output:

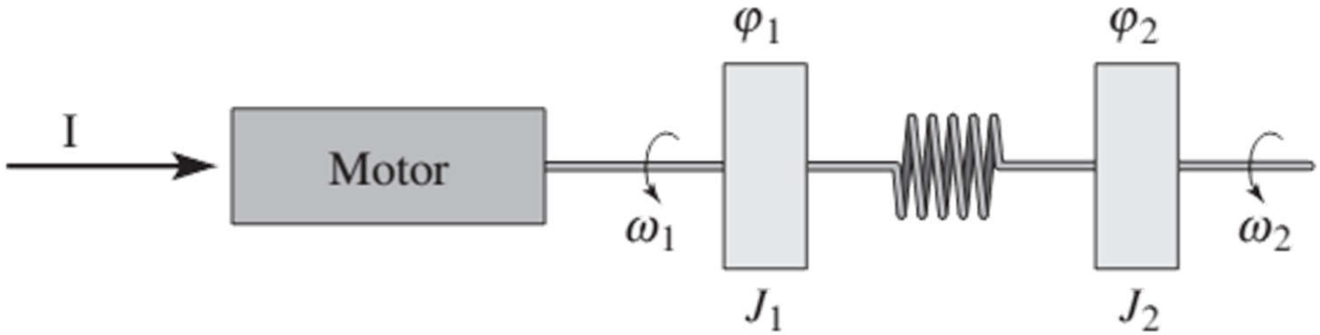
$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$\mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}u$$

Design a linear controller with a single input which is stable at an equilibrium point we define as z_e=0.



But first, back to our example.

Consider a motor driving a system consisting of two inertial disks connected by a compliant shaft:

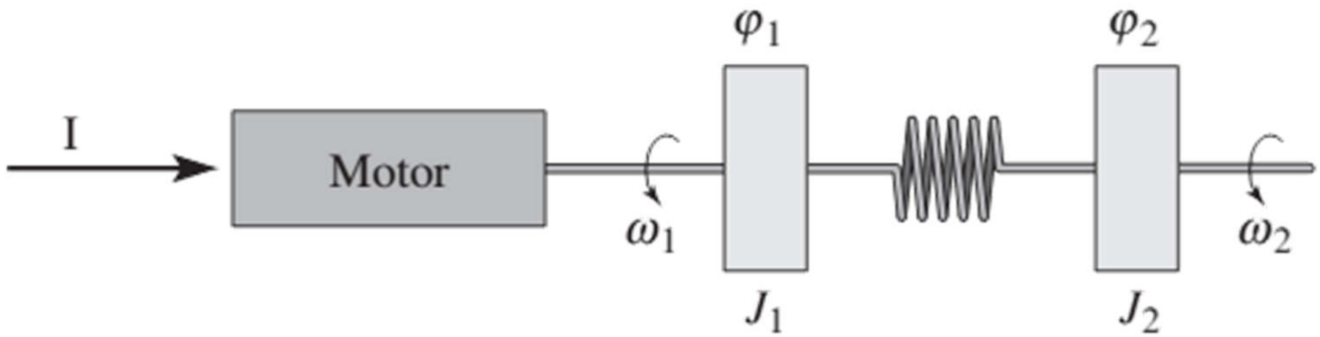


Where the inertial of each disk is given as J_1 and J_2. The spring constant and friction in the shaft are c and k. And the motor constant is k_I.

$$J_1 = 10/9, \quad J_2 = 10, \quad c = 0.1, \quad k = 1, \quad k_I = 1,$$

- *Verify the eigenvalues of the open loop system are 0, 0, and -0.05±i*
- *Design a state feedback controller that produce the system eigenvalues -2, -1, and -1±i.*
- *Simulate the system for a commanded step change in position of the second (outer) inertial disk.*

The equations of motion



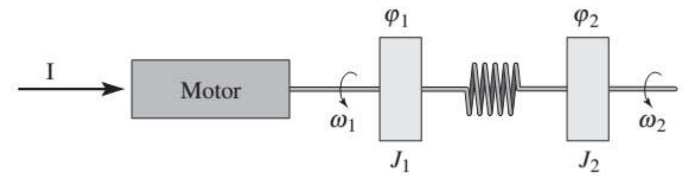
Motor constant

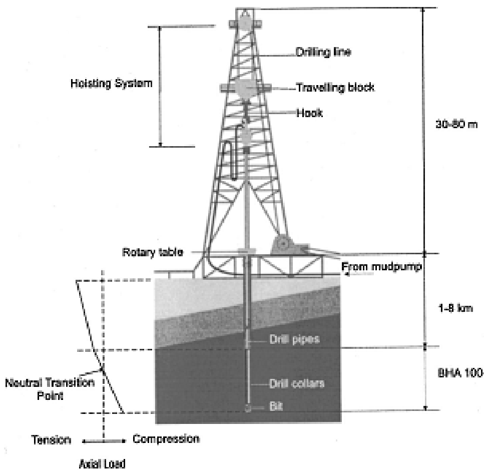
$$J_1 \frac{d^2\varphi_1}{dt^2} + c\left(\frac{d\varphi_1}{dt} - \frac{d\varphi_2}{dt}\right) + k(\varphi_1 - \varphi_2) = k_I I, \quad \leftarrow \textit{Input Current}$$

$$J_2 \frac{d^2\varphi_2}{dt^2} + c\left(\frac{d\varphi_2}{dt} - \frac{d\varphi_1}{dt}\right) + k(\varphi_2 - \varphi_1) = T_d. \quad \leftarrow \textit{Disturbance torque}$$

Derive a state space model for the system by introducing the (normalized) state variables $x_1 = \varphi_1$, $x_2 = \varphi_2$, $x_3 = \omega_1/\omega_0$, and $x_4 = \omega_2/\omega_0$, where $\omega_0 = \sqrt{k(J_1 + J_2)/(J_1 J_2)}$ is the undamped natural frequency of the system when the control signal is zero.

Which is totally a real thing





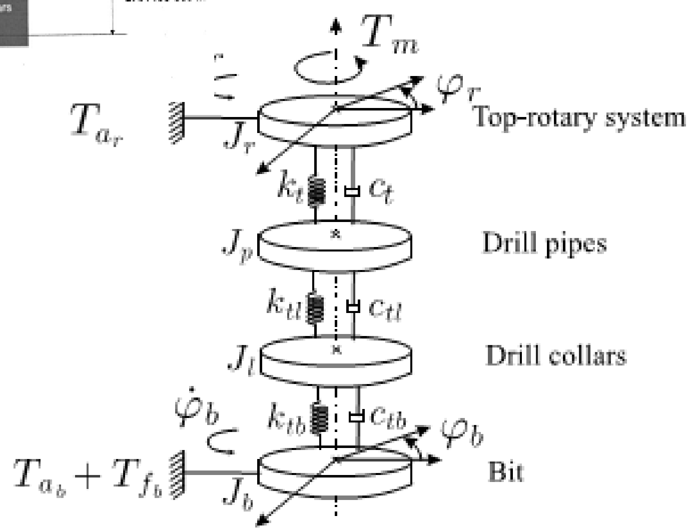


Fig. 3: Robonaut 2 lifting 20 lbs at full extension

Fig. 2: Custom torsion springs from the R2 series elastic actuators

We verified the open loop eigenvalues.

```
%main function
clear all;

%system parameters
J1 = 10/9; J2 = 10;
c = 0.1; k = 1; ki = 1;

%state space model
A = [ 0 1 0 0;
    -k/J1 -c/J1 k/J1 c/J1;
    0 0 0 1;
    k/J2 c/J2 -k/J2 -c/J2 ];

%state space model normalized
w0 = sqrt( k*(J1+J2)/(J1*J2) );

An = [ 0 0 1 0;
    0 0 0 1;
    -w0*k/J1 w0*k/J1 -w0*c/J1 w0*c/J1;
    w0*k/J2 -w0*k/J2 w0*c/J2  -w0*c/J2 ];

%verify open loop eigenvalues
eig(A)
eig(An)
cond(A)
cond(An)
```

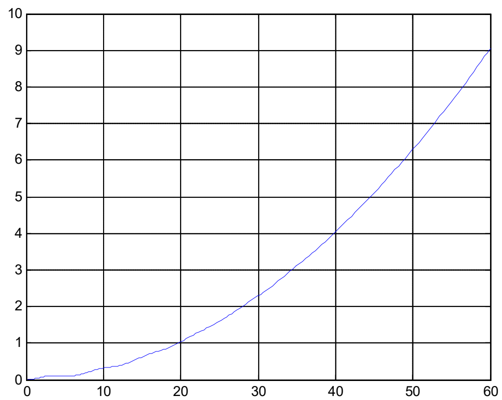
```
>> Prob6_11

ans =

  -0.0500 + 0.9987i
  -0.0500 - 0.9987i
   0.0000 + 0.0000i
   0.0000 - 0.0000i


ans =

  -0.0500 + 0.9987i
  -0.0500 - 0.9987i
   0.0000 + 0.0000i
   0.0000 - 0.0000i

ans =   2.8921e+17

ans =   2.1357e+18
```

No…. (and we got the right answer)

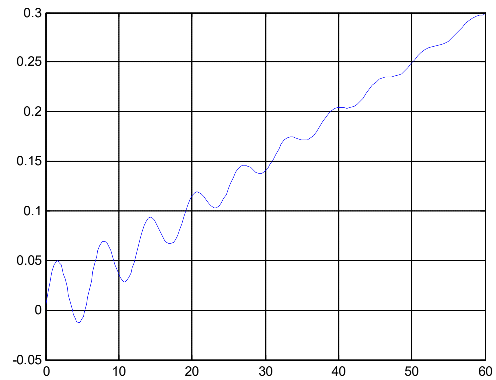
But, whoa….

Open loop with step input (I=0.05)

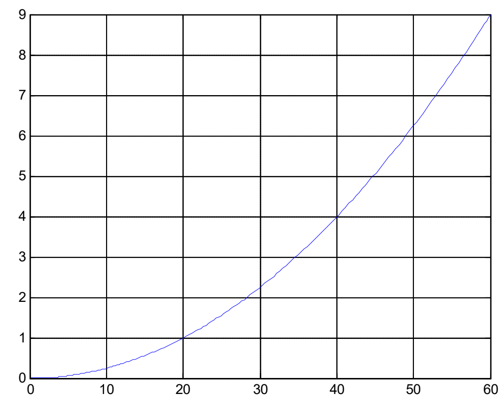
z_1



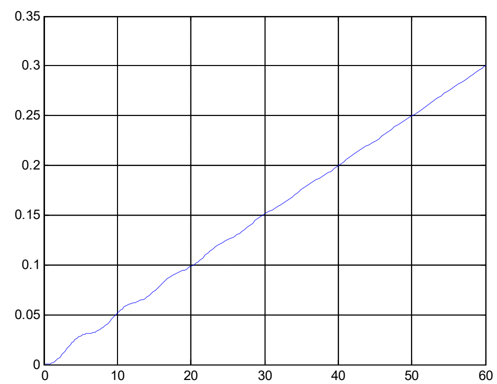
z_2



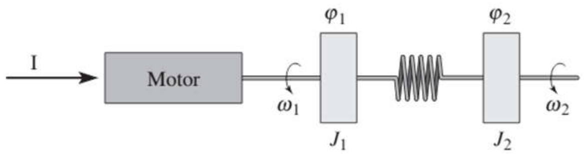
z_3



z_4



```
[t, z ] = ode45('motor', [0 60], [ 0 0 0 0 ]);
plot( t, z(:,1) ); plot( t, z(:,2) );
plot( t, z(:,3) ); plot( t, z(:,4) );
```



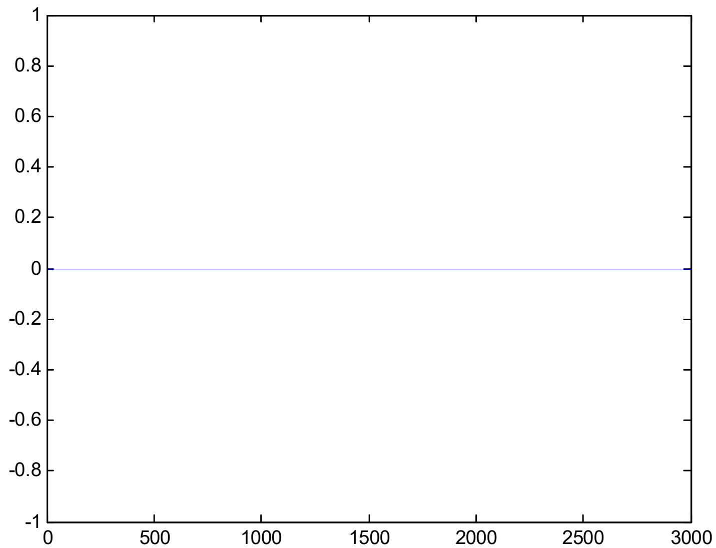
$$J_1 \frac{d^2\varphi_1}{dt^2} = -c(\omega_1 - \omega_2) - k(\varphi_1 - \varphi_2) + k_1 I$$

$$J_2 \frac{d^2\varphi_2}{dt^2} = -c(\omega_2 - \omega_1) - k(\varphi_2 - \varphi_1) + T_d$$

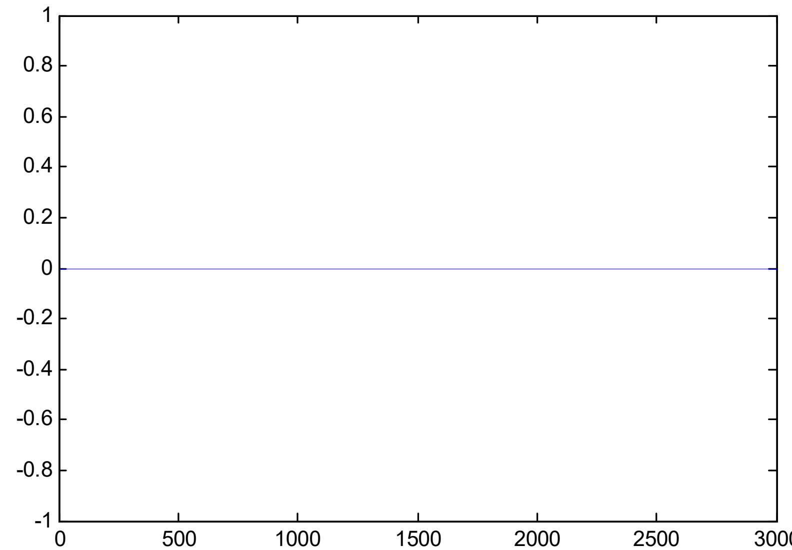
$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \varphi_1 \\ \omega_1 \\ \varphi_2 \\ \omega_2 \end{bmatrix}$$

Do these results make sense?

Open loop with a step input (I=0.0)



z_1



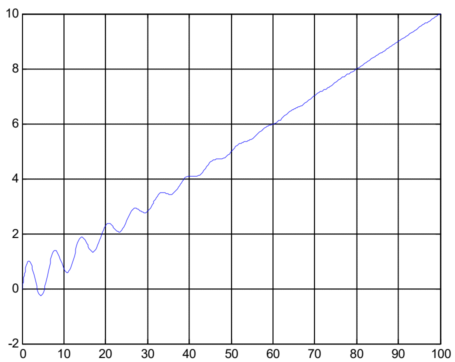
z_3

```
[t, z ] = ode45('motor', [0 300], [ 0 0 0 0 ]);
plot( t, z(:,1) );
plot( t, z(:,3) );
```

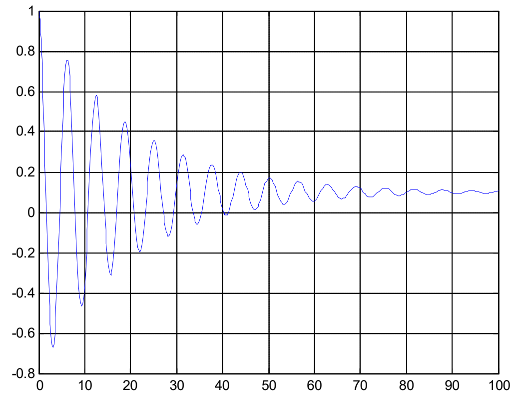
Why did I do this?

Open loop with a step input (I=0.0)

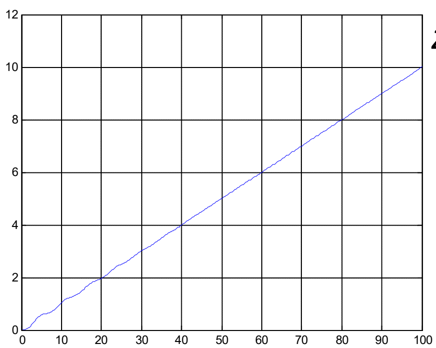
z_1



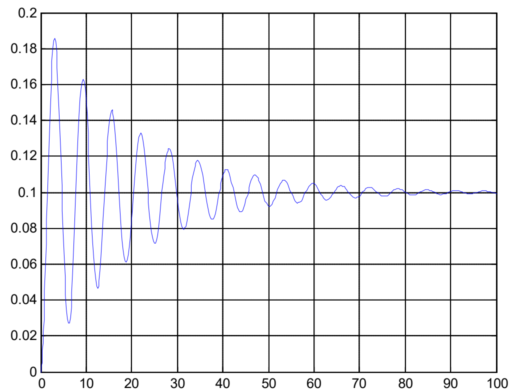
z_2



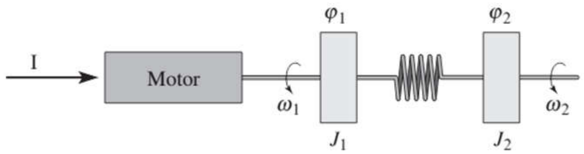
z_3



z_4




```
[t, z ] = ode45('motor', [0 100], [ 0 0 1 0 ]);
plot( t, z(:,1) );
plot( t, z(:,3) );
```



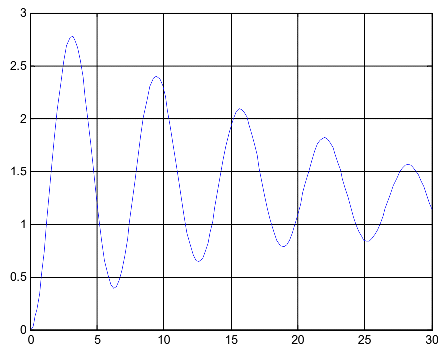
$$J_1 \frac{d^2\varphi_1}{dt^2} = -c(\omega_1 - \omega_2) - k\left(\varphi_1 - \varphi_2\right) + k_1 I$$

$$J_2 \frac{d^2\varphi_2}{dt^2} = -c(\omega_2 - \omega_1) - k\left(\varphi_2 - \varphi_1\right) + T_d$$

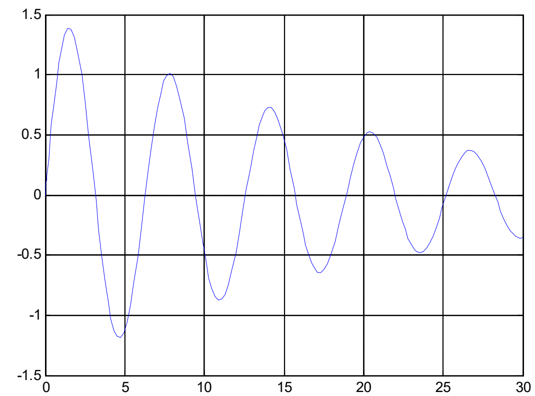
Why did I do this?

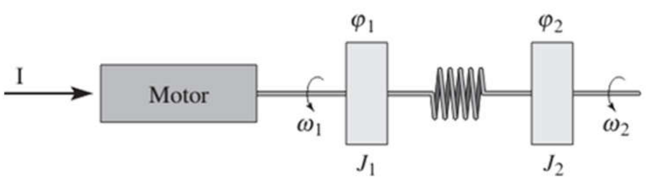
Open loop with (I=1.5, Td=-0.1675)

z_1

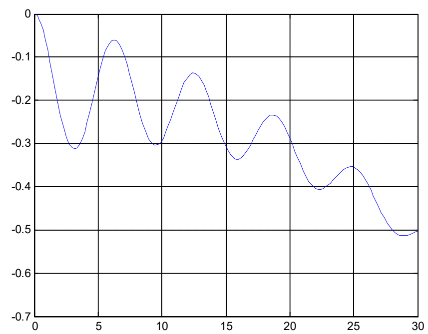


z_2

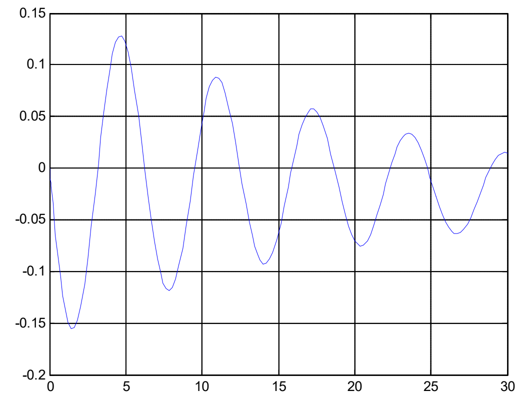




z_3



z_4



$$J_1 \frac{d^2\varphi_1}{dt^2} = -c(\omega_1 - \omega_2) - k\left(\varphi_1 - \varphi_2\right) + k_1 I$$

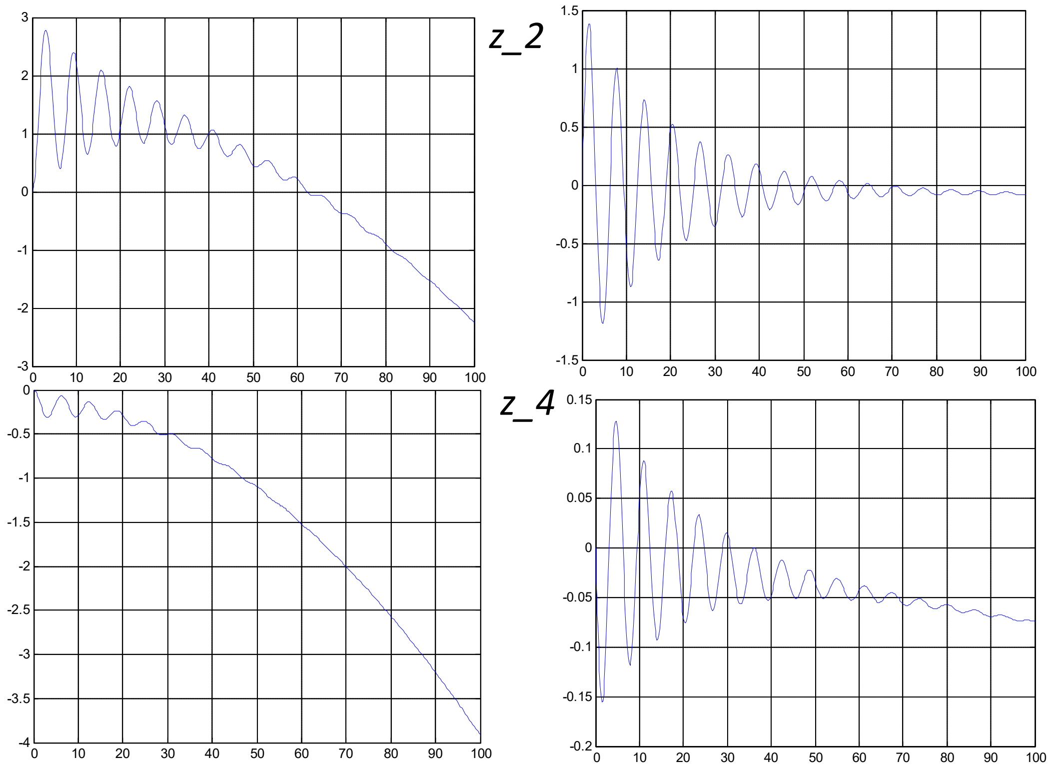
$$J_2 \frac{d^2\varphi_2}{dt^2} = -c(\omega_2 - \omega_1) - k\left(\varphi_2 - \varphi_1\right) + T_d$$

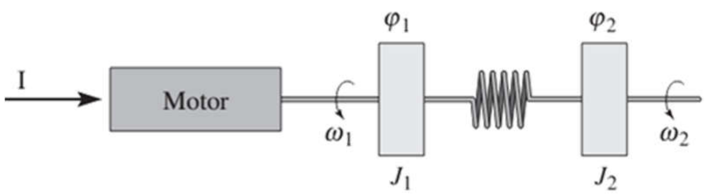
Why did I do this?

```
[t, z ] = ode45('motor', [0 30], [ 0 0 0 0 ]);
plot( t, z(:,1) );
plot( t, z(:,3) );
```

Open loop with (I=1.5, Td=-0.1675)

z_1 z_2 z_3 z_4





$$J_1 \frac{d^2\varphi_1}{dt^2} = -c(\omega_1 - \omega_2) - k(\varphi_1 - \varphi_2) + k_1 I$$

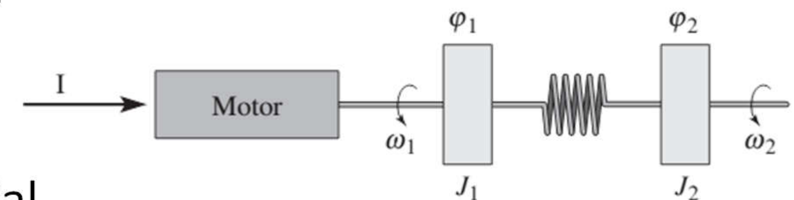
$$J_2 \frac{d^2\varphi_2}{dt^2} = -c(\omega_2 - \omega_1) - k(\varphi_2 - \varphi_1) + T_d$$

Why did I do this?

```
[t, z ] = ode45('motor', [0 100], [ 0 0 0 0 ]);
plot( t, z(:,1) );
plot( t, z(:,3) );
```

Open loop summary…

- I experimented with the open loop system until:
  - I was confident in my programming
  - I understood the physical behavior of the system
- What I learned…
  - It is hard to remove energy unless there is a difference in the velocity between the two inertial objects
  - It is hard to have difference unless
    - There is a disturbance torque
    - They start at different locations.



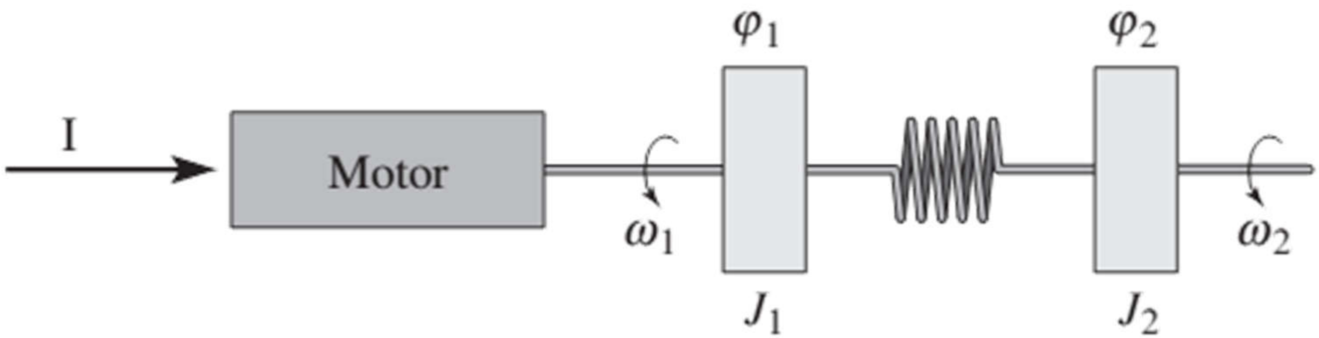
$$J_1 \frac{d^2\varphi_1}{dt^2} + c\left(\frac{d\varphi_1}{dt} - \frac{d\varphi_2}{dt}\right) + k(\varphi_1 - \varphi_2) = k_I I,$$

$$J_2 \frac{d^2\varphi_2}{dt^2} + c\left(\frac{d\varphi_2}{dt} - \frac{d\varphi_1}{dt}\right) + k(\varphi_2 - \varphi_1) = T_d.$$

State/Output feedback

- So…let's design a controller….

Consider a motor driving a system consisting of two inertial disks connected by a compliant shaft:



Where the inertial of each disk is given as J_1 and $J_{2.}$ The spring constant and friction in the shaft are c and k. And the motor constant is k_I.

$$J_1 = 10/9, \qquad J_2 = 10, \qquad c = 0.1, \qquad k = 1, \qquad k_I = 1,$$

- *Verify the eigenvalues of the open loop system are 0, 0, and -0.05±i*
- *Design a state feedback controller that produce the system eigenvalues -2, -1, and -1±i.*
- *Simulate the system for a commanded step change in position of the second (outer) inertial disk.*

Are the states reachable?

$$\frac{dz}{dt} = \mathbf{Az} + \mathbf{Bu} + d$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k/J_1 & -c/J_1 & k/J_1 & c/J_1 \\ 0 & 0 & 0 & 1 \\ k/J_2 & c/J_2 & -k/J_2 & -c/J_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ k_I/J_1 \\ 0 \\ 0 \end{bmatrix} I + \cancel{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/J_2 \end{bmatrix} T_d}$$

What is the input we control?

$$\mathbf{w}_r = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix}$$

```
Wr = B;
for( i=2:length(A))
    Wr(:,i) = A*Wr(:,i-1);
end
Wr
rank(Wr)
det(Wr)
cond(Wr)
```

```
>> Prob6_11

Wr =

         0    0.9000   -0.0810   -0.8019
    0.9000   -0.0810   -0.8019    0.1612
         0         0    0.0090    0.0891
         0    0.0090    0.0891   -0.0179


ans =

     4


ans =

    0.0066


ans =

   19.0886
```

So…… ?

Are the states reachable?

$$\frac{dz}{dt} = \mathbf{Az} + \mathbf{Bu}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_o k/J_1 & \omega_o k/J_1 & -\omega_o c/J_1 & \omega_o c/J_1 \\ \omega_o k/J_2 & -\omega_o k/J_2 & \omega_o c/J_2 & -\omega_o c/J_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_I/J_1 & 0 \\ 0 & 1/J_2 \end{bmatrix} \begin{bmatrix} I \\ T_d \end{bmatrix}$$

What if I use two columns of B (i.e. two inputs)?

$$\mathbf{w}_r = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix}$$

```
>> Prob6_11

Wr =
s
         0         0    0.9000         0   -0.0810    0.0810   -0.8019    0.8019
    0.9000         0   -0.0810    0.0810   -0.8019    0.8019    0.1612   -0.1612
         0         0         0    0.9000    0.0090   -0.0090    0.0891   -0.0891
         0    0.9000    0.0090   -0.0090    0.0891   -0.0891   -0.0179    0.0179

rank =

     4
```

Look at rank, not determinant…

But T_d is not really an input!

So let's find a controller…

Option: Rely on MATLAB

```
%state space model (1 input)
A = [ 0 1 0 0;
    -k/J1 -c/J1 k/J1 c/J1;
    0 0 0 1;
    k/J2 c/J2 -k/J2 -c/J2 ];

B = [ 0;
    kI/J1;
    0;
    0; ];

C = [1 0 0 0];

D = 0;

%p = [-1 -2 -1+1i -1-1i ];
p = [-1 -2 -3 -4 ];

sys = ss( A, B, C, D );
Kc = place( A, B, p )
eigs(A-B*Kc)
kr = -1/(C*inv(A-B*Kc)*B)
```

$$u = -K_c z + k_r r$$

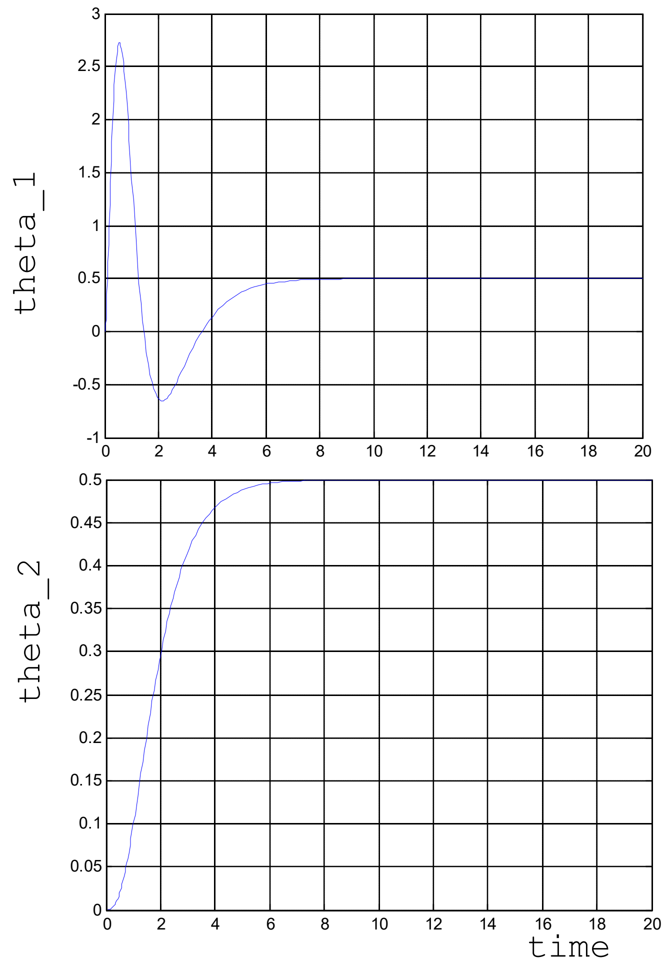
$$\begin{aligned} \dot{z} &= \mathbf{A}z + \mathbf{B}u \\ &= \mathbf{A}z + \mathbf{B}\left(-\mathbf{K_c}z + k_r r\right) \\ &= \left(\mathbf{A} - \mathbf{B}\mathbf{K_c}\right)z + \mathbf{B}k_r r \end{aligned}$$

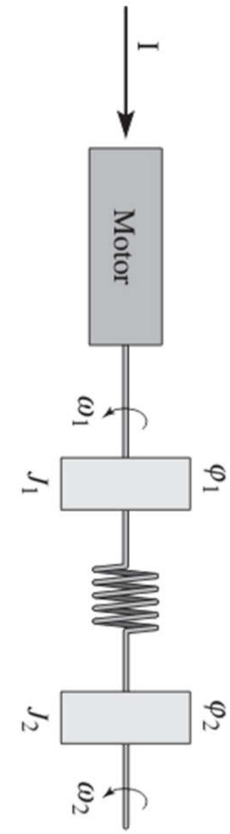
```
Kc =

       32.489 11 234.18 517.89

ans =
           -4
           -3
           -2
           -1

kr = 266.67
```

Are these the right eigenvalues?
Why or why not?

System Response (to dummy eigenvalues)

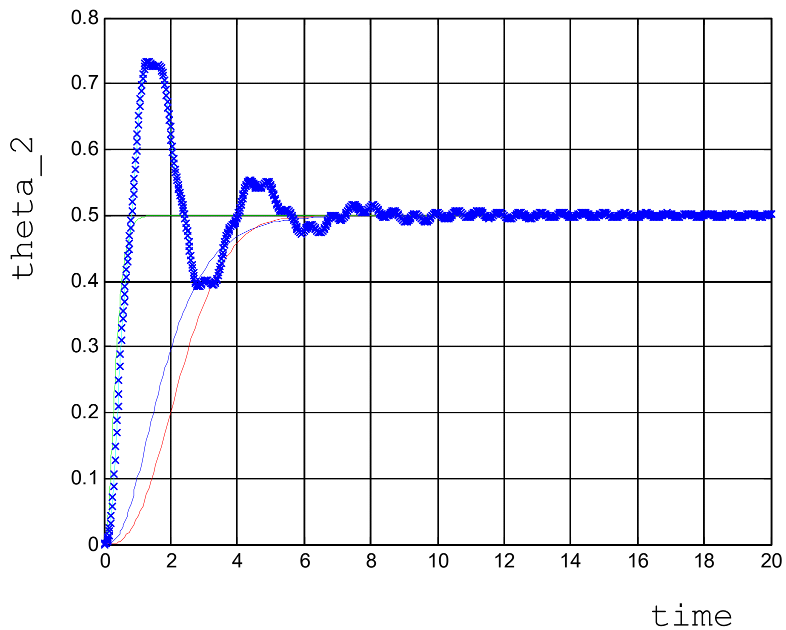
```
% EE362K
% Super fun example time!!!

clear all;
J1 = 10/9;J2 = 10;
c = 0.1; k = 1;
ki = 1;

A = [ 0 1 0 0;
    -k/J1 -c/J1 k/J1 c/J1
    0 0 0 1;
    k/J2 c/J2 -k/J2 -c/J2 ];

B = [ 0 ki/J1 0 0 ]'
C = [ 1 0 0 0 ];
ref = 0.5;

%check to see if a controller exists...
p = [ -4 -3 -2 -1 ];
Kp = place( A, B, p )
kr = -1/(C*inv(A-B*Kp)*B)

%simulate the controller
[t z] = ode45(@motor, [0 20], [0 0 0 0]);
plot( t, z(:,1)); grid on;
```





System Response

```
% EE362K
% Summer 2014
% hwk 5 prob 7

clear all;
J1 = 10/9;J2 = 10;
c = 0.1; k = 1;
ki = 1;

A = [ 0 1 0 0;
    -k/J1 -c/J1 k/J1 c/J1
    0 0 0 1;
    k/J2 c/J2 -k/J2 -c/J2 ];

B = [ 0 ki/J1 0 0 ]'
C = [ 0 0 1 0 ];
ref = 0.5;

%check to see if a controller exists...
p = [ ? ? ? ? ];
Kp = place( A, B, p )
kr = -1/(C*inv(A-B*Kp)*B)

%simulate the controller
[t z] = ode45(@motor, [0 20], [0 0 0 0]);
plot( t, z(:,3)); grid on;
```




```
p = [ -10 -9 -8 -7 ]; %green
p = [ -4 -3 -2 -1 ]; %blue
p = [ -2 -1 -1+1i -1-1i ]; %red
p = [ -.5+2i -.5-2i -.1+10i -1-10i ]; %blue x's
```

Generalized (Acker's) Method

$$K_c = \left[\left(p_1 - a_1 \right) \quad \left(p_2 - a_2 \right) \quad \left(p_3 - a_3 \right) \quad \left(p_4 - a_4 \right) \right] \tilde{w}_r w_r^{-1} \qquad k_r = \frac{-1}{\left(C \left(A - BK \right)^{-1} B \right)}$$

$$\begin{aligned} CE(A) &= (s+0)(s+0)(s+0.05+i)(s+0.05-i) \\ &= (s^2)(s+2s+2) \\ &= s^4 + 0.1s^3 + 1.0025s^2 \end{aligned}$$

$$\begin{aligned} CE(A-BK_c) &= (s+2)(s+1)(s+1+i)(s+1-i) \\ &= (s^2+3s+2)(s+2s^2+2) \\ &= s^4 + 5s^3 + 10s^2 + 10s + 4 \end{aligned}$$

```
A = [ 0 1 0 0;
    -k/J1 -c/J1 k/J1 c/J1;
    0 0 0 1;
    k/J2 c/J2 -k/J2 -c/J2 ];
B = [ 0; kI/J1; 0; 0; ];
C = [1 0 0 0];

p = [5 10 10 4];
a = [.1 1.0025 0 0 ];
r = [-2 -1 -1+1i -1-1i];

Wr = B(:,1);
for( i=2:length(A))
     Wr(:,i) = A*Wr(:,i-1);
end
Wr;
```

```
Ac = [ a(1) a(2) a(3) a(4);
       1 0 0 0;
       0 1 0 0;
       0 0 1 0; ];
Bc = [ 1; 0; 0; 0];

Wrc = Bc(:,1);
for( i=2:length(Ac))
     Wrc(:,i) = Ac*Wrc(:,i-1);
end
Wrc

K = (p-a)*Wrc*inv(Wr)
```

Acker's continued….

What I get from my code….

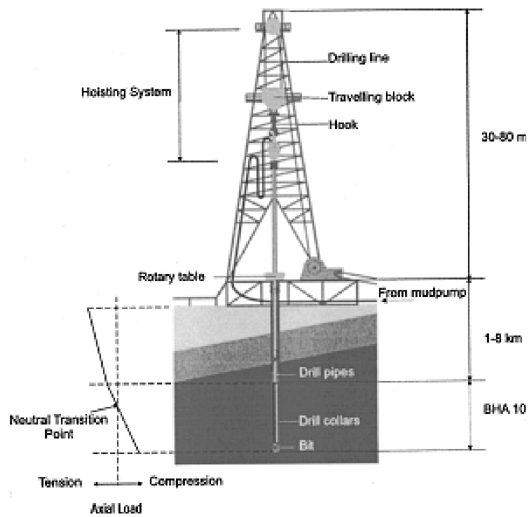
```
>> Prob6_11

Wrc =
            1           0.1        1.0125        0.2015
            0             1           0.1        1.0125
            0             0             1           0.1
            0             0             0             1
K =
       8.9647        5.4444        281.82         206.7
```

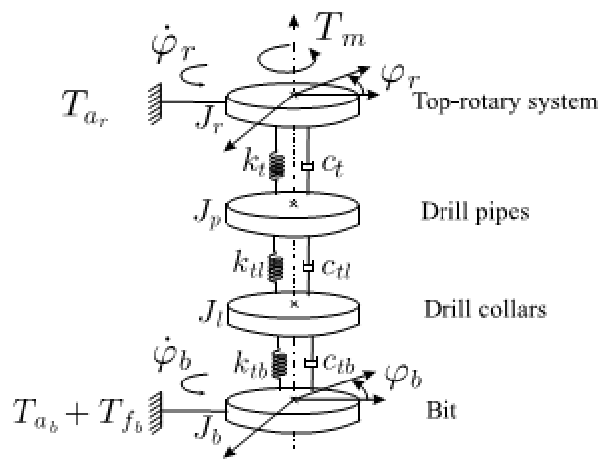
What I got from place( )

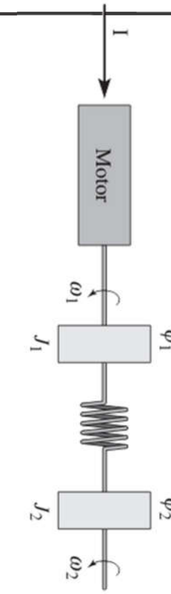
```
Kp = [ 8.9333 5.4444 35.511 101.22 ];
kr = 44.444;
```

Any questions?

So back to drilling…







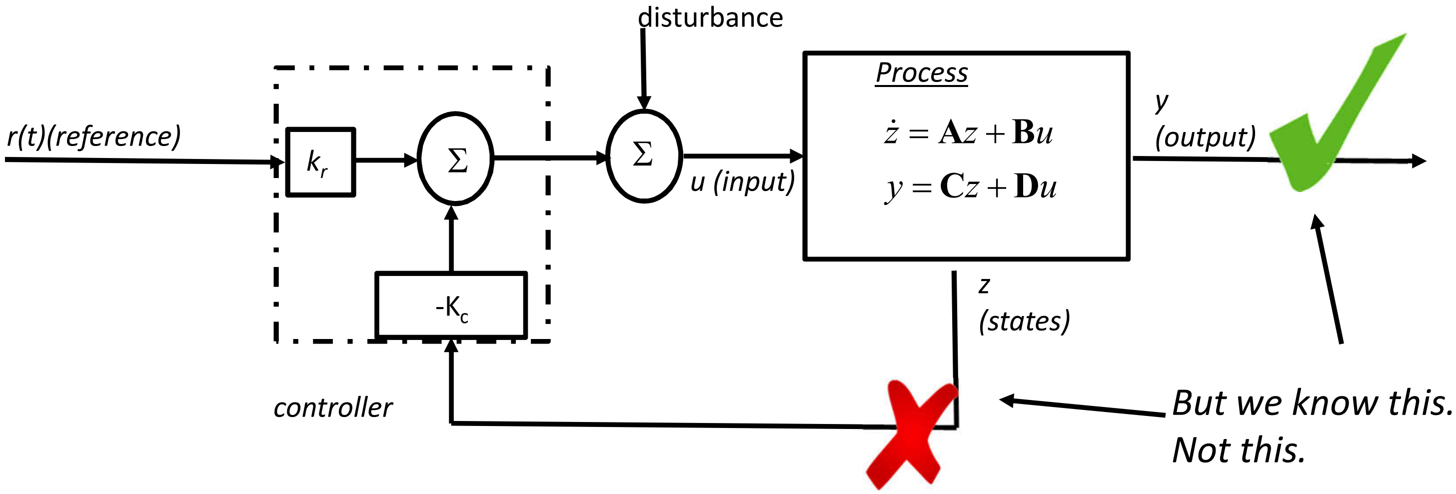
- What are we really interested in?
  - Want to control the velocity of the drill bit
  - But we can only measure the position (or velocity) of the pipe at the surface.

But do we know all the states?

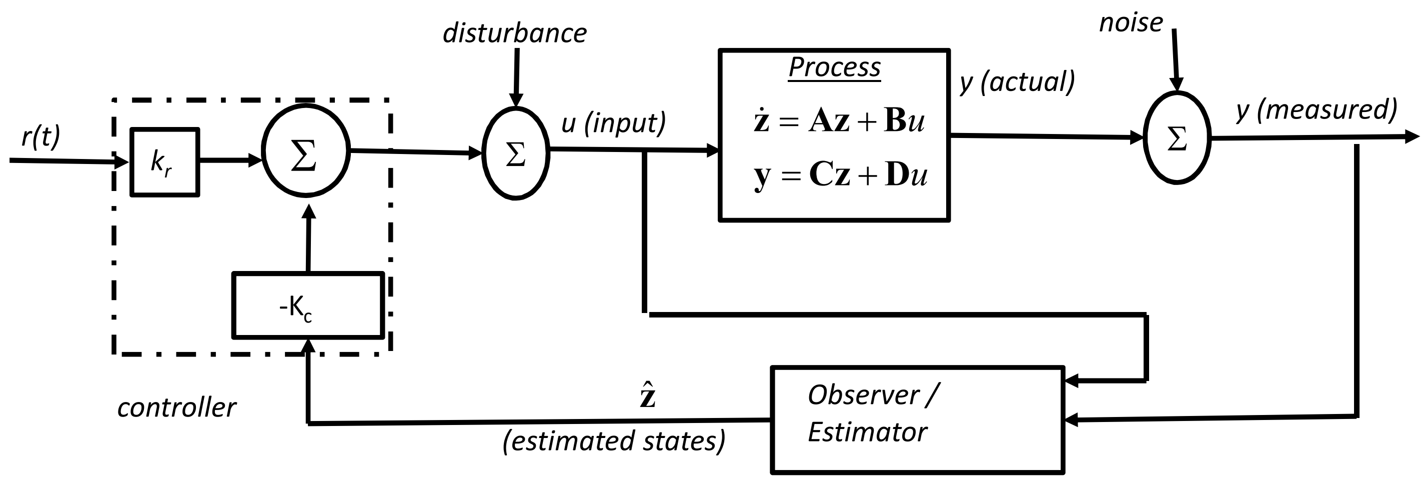
Given a system with the following dynamic model and output:

$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$\mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}u$$

Design a linear controller with a single input which is stable at an equilibrium point we define as z_e=0.



Observer/Estimator



Our objectives

- Determine if a system is observable
- Define the Observable Canonical Form (OCF)
- Create estimates of the states that allow us to continue to implement state feedback

Observability

Observability: The ability to determine the states of the system from the system output.

Start with our state-space model

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$\mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}u$$

The first test is simple…

$$\mathbf{z} = \mathbf{C}^{-1}\left(\mathbf{y} - \mathbf{D}u\right)$$

Whether we need the input or not, the system is observable as long as:

$$\det(\mathbf{C}) \neq \mathbf{0}$$

This is a sufficient, but not necessary condition. If C is not full rank (like for a SISO system), we have another option.

Note, observability is only a function of the dynamics, not the input.

Consider a system without an input

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z}$$
$$\mathbf{y} = \mathbf{C}\mathbf{z}$$

The solution to this system is

$$\mathbf{y} = \mathbf{C}e^{\mathbf{At}}\mathbf{z}(0)$$

where

$$e^{\mathbf{At}} = \sum_{k=0}^{n-1} \frac{1}{k!}\mathbf{A}^k$$

And thus for n states…

$$\mathbf{y}(t) = \sum_{k=0}^{n-1} \frac{1}{k!}\mathbf{C}\mathbf{A}^k\mathbf{z}(0)$$
$$= \left(\tfrac{1}{0!}\mathbf{C}\mathbf{A}^0 + \tfrac{t}{1!}\mathbf{C}\mathbf{A} + \tfrac{t^2}{2!}\mathbf{C}\mathbf{A}^2 + \ldots + \tfrac{t^n}{(n-1)!}\mathbf{C}\mathbf{A}^{n-1} \right)\mathbf{z}(0)$$

Observability

Observability: The ability to determine the states of the system from the system output.

From the previous page…

$$\mathbf{y}(t) = \left(\tfrac{1}{0!}\mathbf{CA}^0 + \tfrac{t}{1!}\mathbf{CA} + \tfrac{t^2}{2!}\mathbf{CA}^2 + \cdots + \tfrac{t^k}{k!}\mathbf{CA}^k + \cdots \right)\mathbf{z}(0)$$

Given y, we know the states, z, if the material in the parentheses is invertible.

$$\mathbf{z} = \left(\tfrac{1}{0!}\mathbf{CA}^0 + \tfrac{1}{1!}\mathbf{CA} + \tfrac{1}{2!}\mathbf{CA}^2 + \ldots + \tfrac{1}{(\mathrm{n}-1)!}\mathbf{CA}^{n-1} \right)^{-1} \mathbf{y}(t)$$

To simplify, recognize the coefficients are not necessary to perform the test, and recall that for a single output C is 1x*n* and A is *nxn….*

$$\mathbf{w_o} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{C}^T & \mathbf{A}^T\mathbf{C}^T & \mathbf{A}^{2^T}\mathbf{C}^T & \cdots & \mathbf{A}^{(n-1)^T}\mathbf{C}^T \end{bmatrix}^T$$

This is know as the ***observability matrix*** *and if and only if it is invertible can we say the matrix in the equation above is also invertible, and thus, the system is observable.*

Proof of Observability Rank Condition

Theorem: A linear system is observable if and only if the observability matrix, w_o is full rank

Proof (sufficiency): Start with the output in form of the convolution integral…

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau + \mathbf{D}u(t)$$

Since we know u(t), we can subtract its contribution, and write the new output as…

$$\tilde{\mathbf{y}}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0)$$

Differentiate the new output and evaluate at t=0….

$$\begin{aligned}
\tilde{\mathbf{y}}(0) &= \mathbf{C}\mathbf{z}(0) \\
\dot{\tilde{\mathbf{y}}}(0) &= \mathbf{C}\mathbf{A}\mathbf{z}(0) \\
\ddot{\tilde{\mathbf{y}}}(0) &= \mathbf{C}\mathbf{A}^2\mathbf{z}(0) \\
&\vdots \\
\tilde{\mathbf{y}}^{(n)}(0) &= \mathbf{C}\mathbf{A}^{n-1}\mathbf{z}(0)
\end{aligned}$$

Which we can rewrite in matrix form and solve for the states.

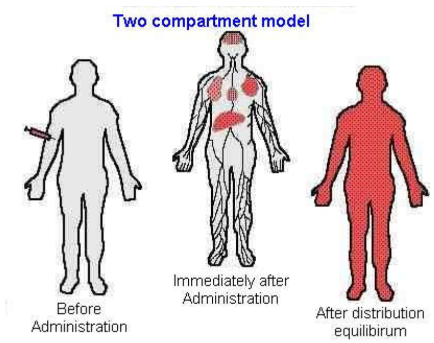
$$\mathbf{z}(0) = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\mathbf{y}}(0) \\ \dot{\tilde{\mathbf{y}}}(0) \\ \ddot{\tilde{\mathbf{y}}}(0) \\ \\ \tilde{\mathbf{y}}^{(n)}(0) \end{bmatrix} = \mathbf{w}_0^{-1} \begin{bmatrix} \tilde{\mathbf{y}}(0) \\ \dot{\tilde{\mathbf{y}}}(0) \\ \ddot{\tilde{\mathbf{y}}}(0) \\ \\ \tilde{\mathbf{y}}^{(n)}(0) \end{bmatrix}$$

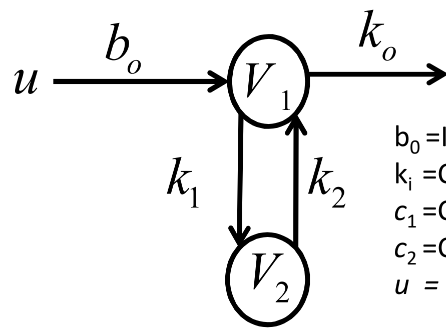
Thus if the observability matrix is full rank, we can determine the states that produce a given output.

So we solve for z(0) and then find z(t) using.

$$\mathbf{z}(t) = e^{\mathbf{A}t}\mathbf{z}(0)$$

Example: 2 Vessel model





b_0 =Intravenous concentration flow rate
k_i =Concentration flow rate between vessels
c_1 =Concentration in circulatory system
c_2 =Concentration in muscular system
u = Intravenous flow ($mL_{solution}/s$)

Recall our state-space model…

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_1 \\ k_2 & -k_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u$$

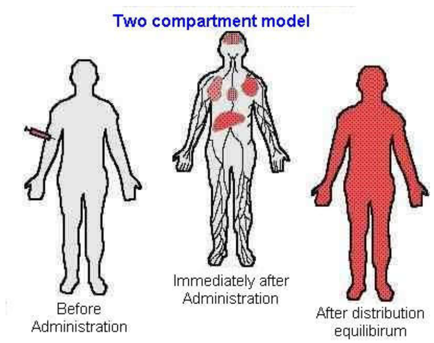
$$w_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}$$

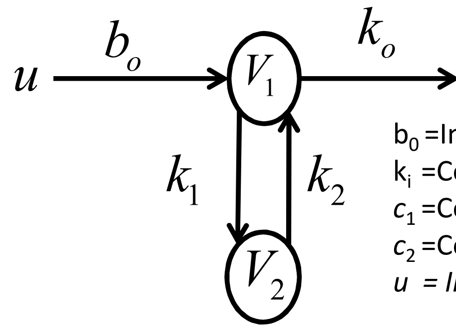
$$= \begin{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -k_o - k_1 & k_1 \\ k_2 & -k_2 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ k_2 & -k_2 \end{bmatrix}$$

Therefore, observable if $k_2 \neq 0$.

Example: 2 Vessel model





b_0 =Intravenous concentration flow rate
k_i =Concentration flow rate between vessels
c_1 =Concentration in circulatory system
c_2 =Concentration in muscular system
u = Intravenous flow ($mL_{solution}/s$)

$$w_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -k_o - k_1 & k_1 \\ k_2 & -k_2 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ -k_o - k_1 & k_1 \end{bmatrix}$$

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_1 \\ k_2 & -k_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u$$

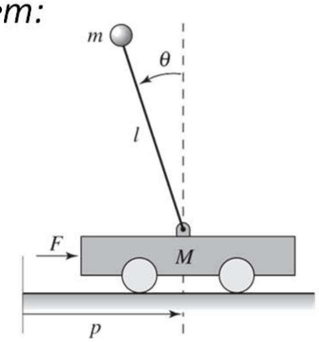
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u$$

Therefore, observable if $k_1 \neq 0$.

Now measure concentration in vessel 1.

Recall our Inverted Pendulum

In the state feedback lectures, we derived our equations of motion for an inverted pendulum system:



$$\sum F_i = (M+m)\ddot{x}$$

$$\sum \tau_i = I\ddot{\theta}$$

$$(M+m)\ddot{x} = ml\cos(\theta)\ddot{\theta} - c\dot{x} - ml\sin(\theta)\dot{\theta}^2 + F$$

$$(J+ml^2)\ddot{\theta} = ml\cos(\theta)\ddot{x} - \gamma\dot{\theta} + mgl\sin(\theta)$$

And our state-space model was….

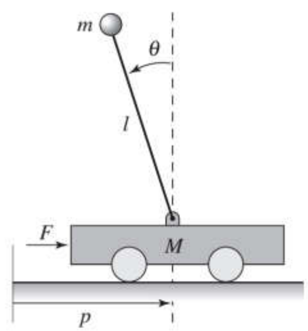
$$\mathbf{z} = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T \qquad \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{\mu}\begin{bmatrix} (J+ml^2) & -ml \\ ml & (M+m) \end{bmatrix}\left[\begin{bmatrix} 0 \\ mgl\theta \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix}\right]$$

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2l^2g}{\mu} & 0 & 0 \\ 0 & \frac{(M+m)mgl}{\mu} & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{J+ml^2}{\mu} \\ \frac{lm}{\mu} \end{bmatrix}u$$

Example: Inverted Pendulum

Determine the observability of the inverted pendulum derived in previous lecture.

$$M = 10kg$$
$$m = 80kg$$
$$l = 1m$$
$$J = 100\frac{kgm^2}{s^2}$$



Recall...

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2 l^2 g}{\mu} & 0 & 0 \\ 0 & \frac{(M+m)mgl}{\mu} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{J+ml^2}{\mu} \\ \frac{lm}{\mu} \end{bmatrix} u$$

The output of interest is the angular position...

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{z}$$

We can use MATLAB to calculate the observability matrix.

```
Mc = 10; Mb = 80; len = 1; J = 100; g = -9.81;

mu = (Mc+Mb)*(J+Mb*len^2)-Mb^2*len^2;

A = [ 0 0 1 0;
    0 0 0 1;
    0 Mb^2*len^2*g/mu 0 0;
    0 (Mb + Mc)*Mb*g*len 0 0; ];

C = [ 0 1 0 0];

wo = [ C; C*A; C*A*A; C*A*A*A]

rank(wo)
rank(obsv(A, C)) %observability MATLAB function
```

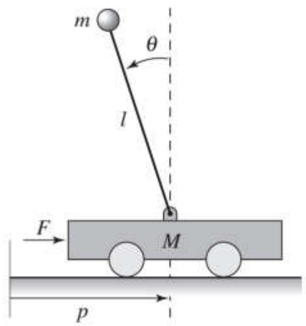
```
>> lecture11scratch

wo =
     0      1      0      0
     0      0      0      1
     0 -70632      0      0
     0      0      0 -70632

ans = 2
ans = 2
```

Therefore this system is NOT observable.

Example: Inverted Pendulum, cont'd

Determine the observability of the inverted pendulum derived in previous lecture.

$$M = 10kg \quad m = 80kg \quad l = 1m \quad J = 100\frac{kgm^2}{s^2}$$



Recall...

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2 l^2 g}{\mu} & 0 & 0 \\ 0 & \frac{(M+m)mgl}{\mu} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{J+ml^2}{\mu} \\ \frac{lm}{\mu} \end{bmatrix} u$$

but we can always add more sensors!

$$y = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}\mathbf{z} \quad \textit{or...} \quad \mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\mathbf{z}$$

We can use MATLAB to calculate the observability matrix.

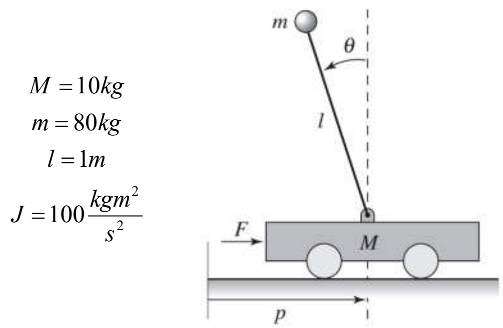
```
Mc = 10; Mb = 80; len = 1; J = 100; g = -9.81;

mu = (Mc+Mb)*(J+Mb*len^2)-Mb^2*len^2;

A = [ 0 0 1 0;
    0 0 0 1;
    0 Mb^2*len^2*g/mu 0 0;
    0 (Mb + Mc)*Mb*g*len 0 0; ];

C = [ 1 1 0 0];

wo = [ C; C*A; C*A*A; C*A*A*A]

rank(wo)
rank(obsv(A, C))
```

```
wo =  1.0e+04 *
 [ 0.0001   0.0001        0        0
        0        0   0.0001   0.0001
        0  -7.0638        0        0
        0        0        0  -7.0638 ]

ans =  4
ans =  4
```

Therefore this system can be redesigned to be observable.
(note: cond(w_o)=70,000!)

Example: Inverted Pendulum, cont'd

Determine the observability of the inverted pendulum derived in previous lecture.



Recall...

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m^2l^2g}{\mu} & 0 & 0 \\ 0 & \frac{(M+m)mgl}{\mu} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{J+ml^2}{\mu} \\ \frac{lm}{\mu} \end{bmatrix} u$$

What if y is a vector and not a sum?

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{z}$$

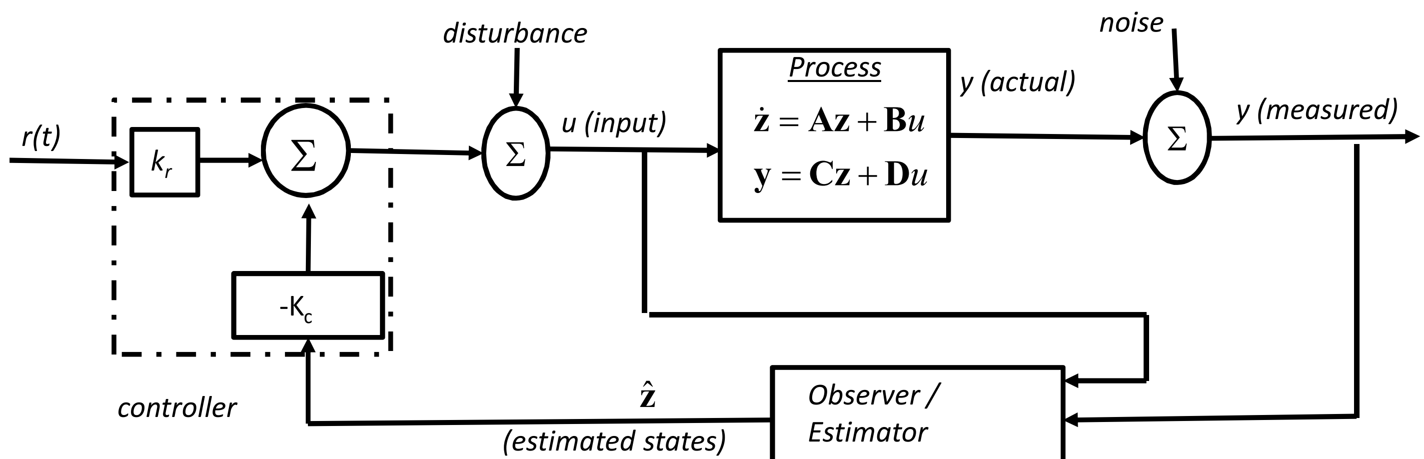
We can use MATLAB to calculate the observability matrix.

```
A = [ 0 0 1 0;
    0 0 0 1;
    0 Mb^2*len^2*g/mu 0 0;
    0 (Mb + Mc)*Mb*g*len 0 0; ];

C = [ 1 0 0 0; 0 1 0 0];

wo = [ C; C*A; C*A*A; C*A*A*A]

rank(wo)
rank(obsv(A, C))
```

```
wo = 1.0e+04 *

  0.0001        0        0        0
       0   0.0001        0        0
       0        0   0.0001        0
       0        0        0   0.0001
       0  -0.0006        0        0
       0  -7.0632        0        0
       0        0        0  -0.0006
       0        0        0  -7.0632

ans = 4
ans = 4
```

Note: same condition since cond(w_o)=70,000.

Observer/Estimator Summary



Our objectives

- Determine if a system is observable
- Define the Observable Canonical Form (OCF)
- Create Estimators that allow us to continue to implement state feedback