State Feedback

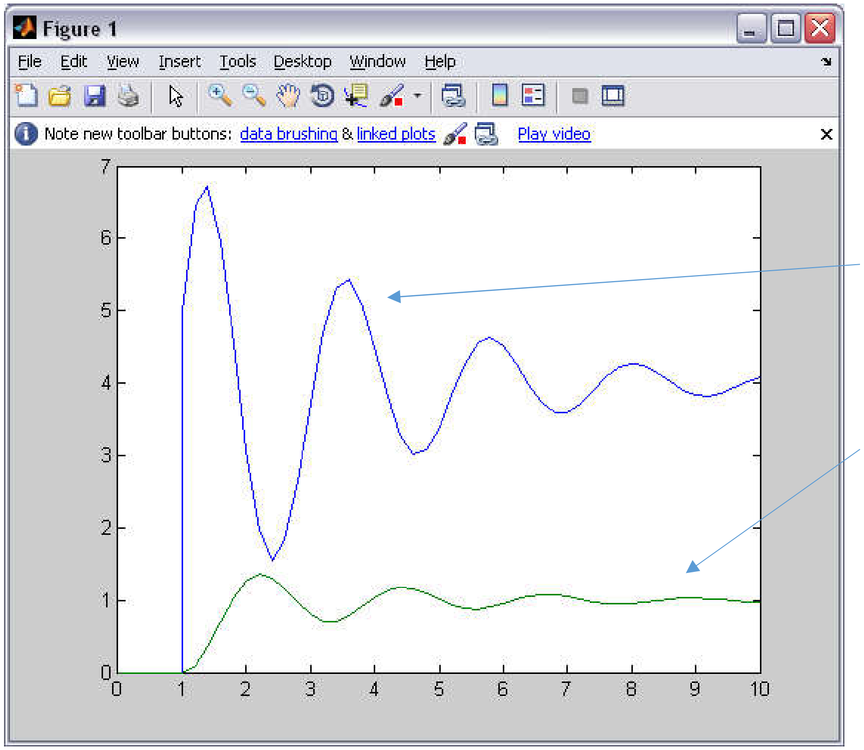
Mitch Pryor

Lesson objective

- Use knowledge of the state values (**z**) of a system (**A, B, C, D**) to select a control input (**u**) that gives us a desired system output (**y**).

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

$$\mathbf{y} = \mathbf{Cz} + \mathbf{Du}$$

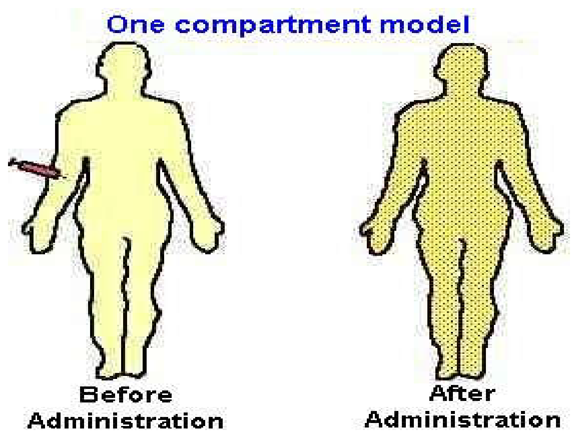


An input u (blue)…

…to get the output to a desired value of 1 (green)

How to pick **u**?

Start with an example: drug administration



simple 1st order model

$$V\frac{dc}{dt} = -qc \qquad c(0) = c_o$$

where

V=: volume of the vessel (mL_{vessel})

c =: drug concentration (mL_{solute}/mL_{vessel})

q=:outflow rate (mL_{solute}/s)

therefore,

$$c(t) = c_o e^{-\frac{qt}{V}}$$

if we add an input…

$$V\frac{dc}{dt} = -qc + c_d u$$

where

c_d =: concentration of the drug ($mL_{solute}/mL_{solution}$)
u=:intravenous flow rate ($mL_{solution}/s$)

$$\frac{dc}{dt} = -\frac{q}{V}c + \frac{c_d}{V}u$$

$$\frac{dc}{dt} = -kc + b_d u$$

where

k =: concentration flow rate (q/V)
b_d =: intravenous concentration flow rate (c_d/V)

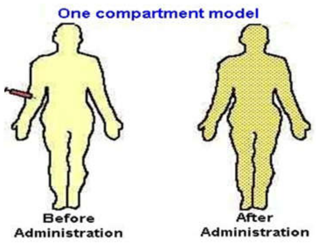
Consider two input options

Administered with a shot…

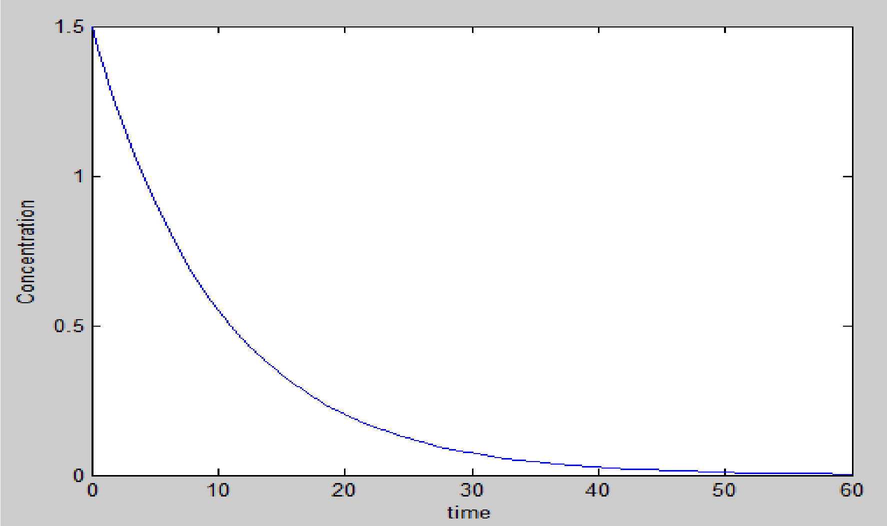
$$u(t) = \begin{cases} u_o & t = t_0 \\ 0 & t \neq 0 \end{cases}$$

or intravenously…

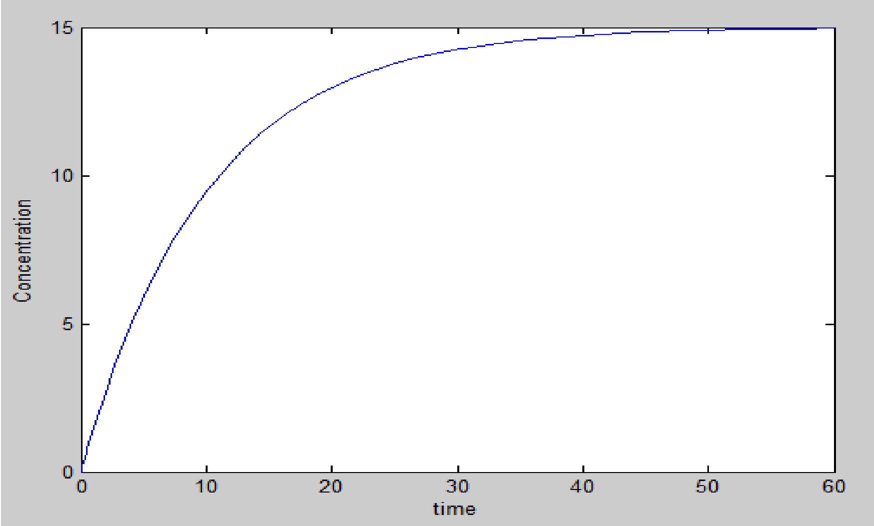
$$u(t) = \begin{cases} 0 & t < t_0 \\ u_o & t \geq 0 \end{cases}$$



a.k.a. the Impulse Function

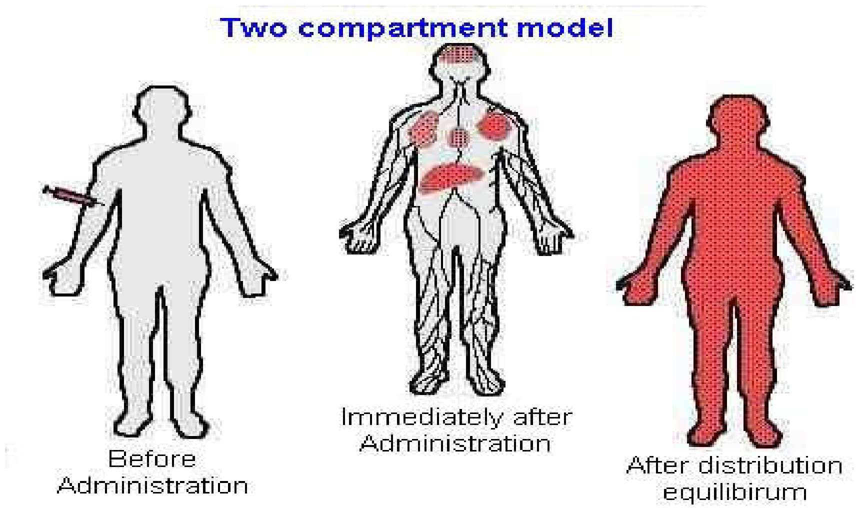


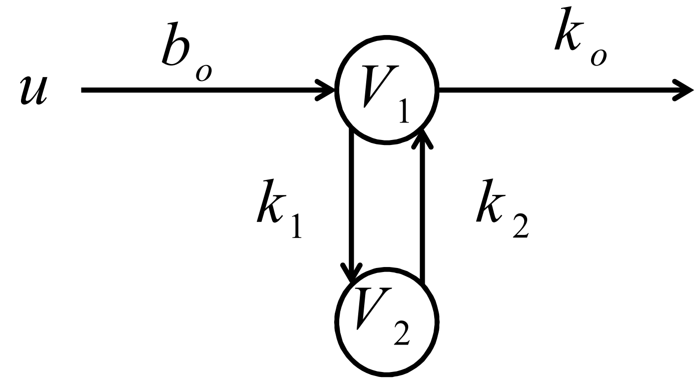
a.k.a. the step function



(note: to reproduce these graphs in MATLAB: t_o = 0.0, k = 0.1, u_o=1.5 and b_d = 1.0)

Example: 2 Vessel model





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

<u>The “equations of motion”…</u>

$$\frac{dc_1}{dt} = -k_1 c_1 + k_2 c_2 - k_0 c_1 + b_o u$$
$$\frac{dc_2}{dt} = k_1 c_1 - k_2 c_2$$

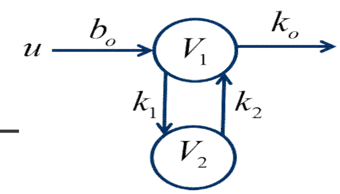
$$\Rightarrow$$

<u>To state space…</u>

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

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

Is the system stable?



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

$$= \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} c + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u \qquad\qquad = \begin{bmatrix} 0 \\ 1 \end{bmatrix} c + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \det \begin{bmatrix} \lambda + k_o + k_1 & -k_2 \\ -k_1 & \lambda + k_2 \end{bmatrix} = 0$$

$$(\lambda + k_o + k_1)(\lambda + k_2) - k_1 k_2 = 0$$

$$\lambda^2 + (k_0 + k_1 + k_2)\lambda + k_0 k_2 = 0$$

Note the cancelling terms…

For what values of the flow rates is the system stable?

Two requirements

$$\boxed{\begin{matrix} k_0 > 0 \\ k_2 > 0 \end{matrix}}$$

For example…

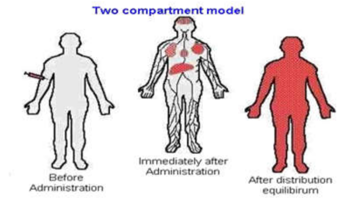
$$\begin{matrix} k_0 = 1 \\ k_1 = 1 \\ k_2 = 2 \end{matrix} \Rightarrow \lambda = \begin{bmatrix} -0.5858 \\ -3.4142 \end{bmatrix}$$

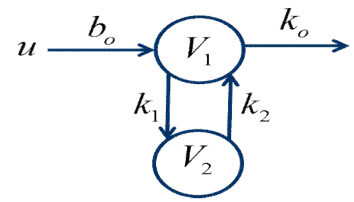
or…

$$\begin{matrix} k_0 = 0 \\ k_1 = 2 \\ k_2 = 2 \end{matrix} \Rightarrow \lambda = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$$

Now to control the concentration!

$$\frac{dc}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} c + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u \qquad y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$



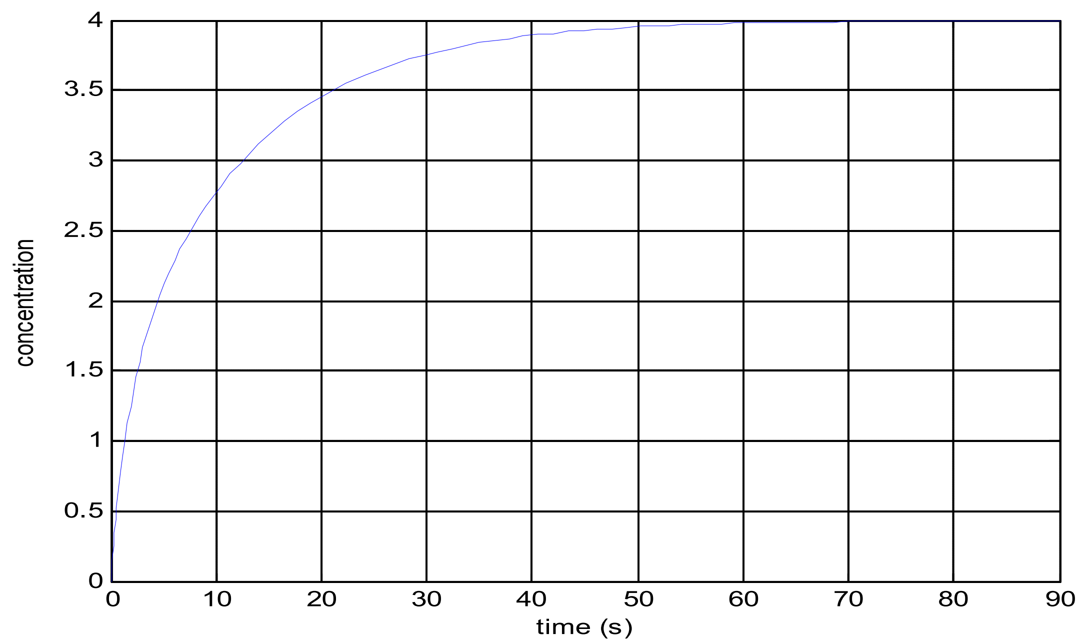


Find u such that the concentration of the drug in muscular system is 8 mL per 100mL

*Let's first find an **open loop controller***

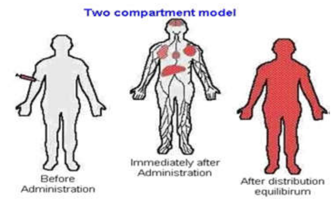
Let's start with a guess.

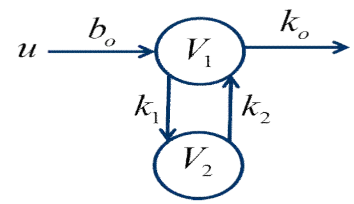
```
global u;
u = 1;

[t,z] = ode45('twoVolume', [0 90], [0 0]);
plot(t, z(:,2));


function cprime = twoVolume( t, c )
global u;

k0 = 0.1; k1 = 0.1;
k2 = 0.5; b0 = 1.5

A = [ -k0-k1 k2; k1 -k2 ];
B = [ 0; 1];

cprime = A*c + B*u;
```



2 vessel *open loop* control

$$\frac{dc}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} c + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u \qquad y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$



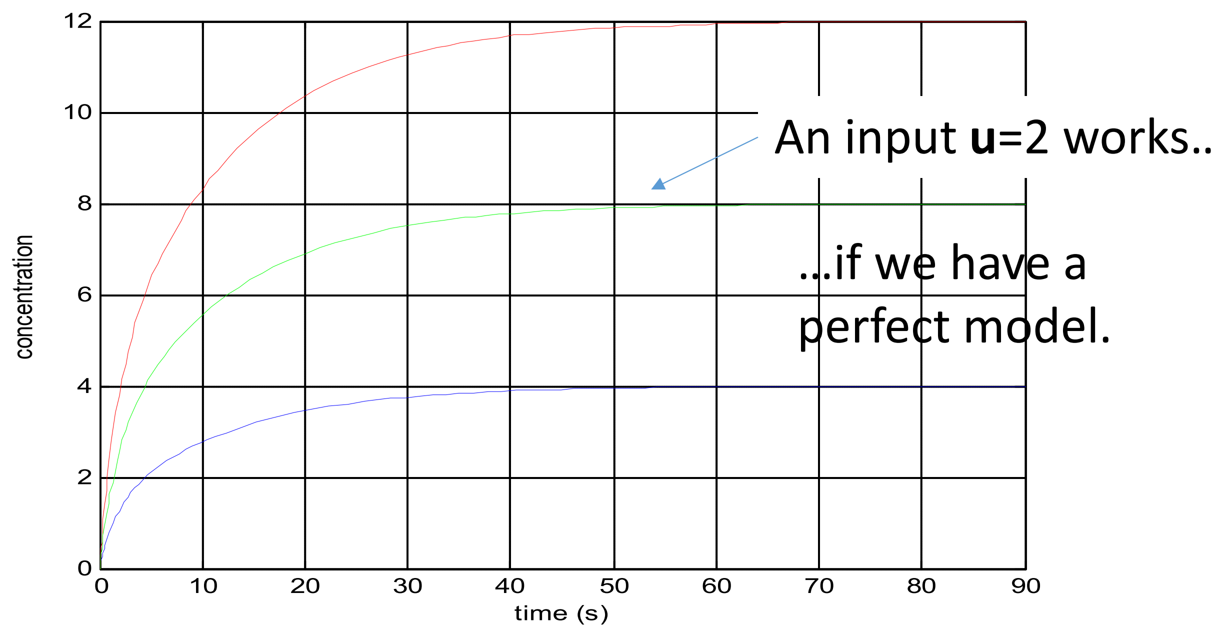


Find u such that the concentration of the drug in muscular system is 8 mL per 100mL

Via trial & error, we arrive at a solution....

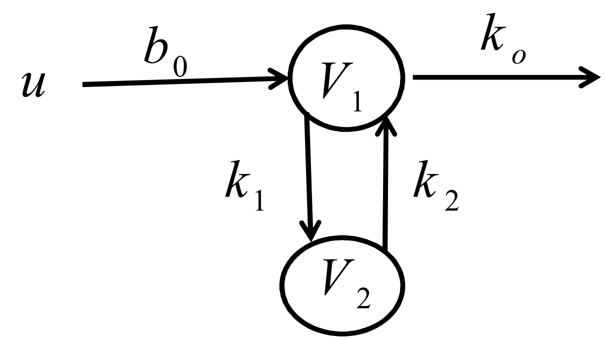
```
global u;
u = 2; %1 %3

[t,z] = ode45('twoVolume', [0 90], [0 0]);
plot(t, z(:,2), 'g');
```

```
function cprime = twoVolume( t, c )
global u;

k0 = 0.1; k1 = 0.1;
k2 = 0.5; b0 = 1.5

A = [ -k0-k1 k2; k1 -k2 ];
B = [ 0; 1];

cprime = A*c + B*u;
```

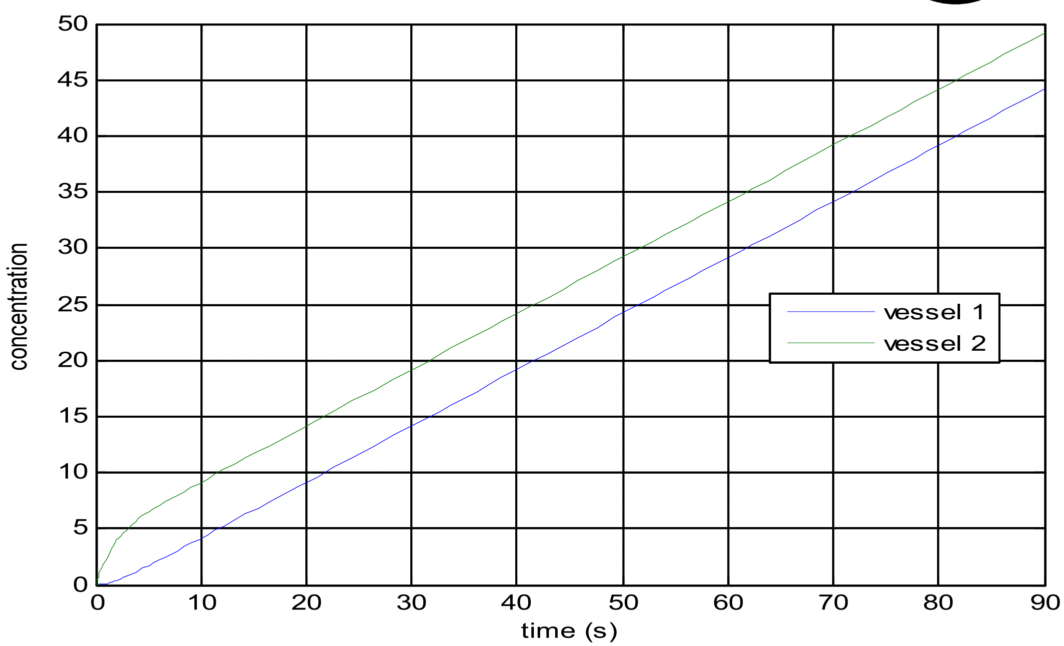


Unstable 2 Vessel Example

$$\frac{dc}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} c + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u \qquad y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} c$$




```
global u;
u = 3;

[t,z] = ode45('twoVolume', [0 90], [0 0]);
plot(t, z);


function cprime = twoVolume( t, c )
global u;

k0 = 0.0; k1 = 0.1;
k2 = 0.5; b0 = 1.5

A = [ -k0-k1 k2; k1 -k2 ];
B = [ 0; 1];

cprime = A*c + B*u;
```



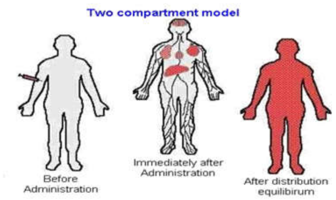
Open vs. closed loop control

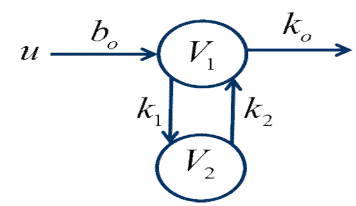
- **Open loop example results**
  - Trial & error found a *u* that gave us the desired concentration in vessel 2.
- **Open loop control**
  - No feedback. Only works if model is perfect and there are no disturbances.
  - Model is never perfect. There is almost always a disturbance.
- **Closed loop control**
  - Feedback. State or output value(s) are used to adjust system input.
  - **State feedback control** – Feedback the system's state values to determine the input.
    - Assumes all states are known or measured (not likely)
  - **Output feedback control –** Feedback the system's output value to determine the input.
    - Formulate *observers* that estimate the state information from the output signal.

Let's start with a simple example for our 2 vessel system.

State feedback example

$$\frac{dc}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} c + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u \qquad y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$





Objective: *create a* ***<u>feedback controller</u>*** *that adjusts the input such that vessel two maintains a desired concentration c_d. Of the three primary performance issues (rise time, overshoot, and steady state error), avoiding overshoot is the most important. (for this example)*

Define a controller:

$$u = k_p \left(c_d - c_2 \right) + u_0$$

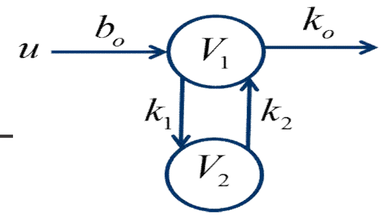
proportional control term to regulate difference between c_2 and c_d

Base level (minimum) rate of injection

Insert control law into the system:

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(k_p \left(c_d - c_2 \right) + u_0 \right)$$

State feedback example



Our system with the controller:

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}\mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix}\left(k_p\left(c_d - c_2\right) + u_0\right)$$

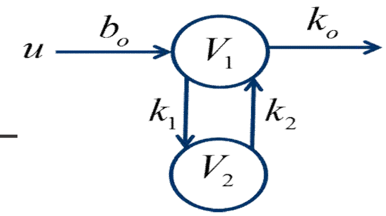
Separate the feedforward and feedback terms…

$$= \underbrace{\begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}}_{\text{A}}\mathbf{c} + \underbrace{\begin{bmatrix} b_0 \\ 0 \end{bmatrix}}_{\text{B}}\underbrace{\left(-k_p c_2\right)}_{\textit{negative state feedback}} + \underbrace{\begin{bmatrix} b_0 \\ 0 \end{bmatrix}}_{\text{B}}\underbrace{\left(k_p c_d + u_o\right)}_{\textit{reference and open loop terms}}$$

$$= \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix}\begin{bmatrix} 0 & -k_p \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix}\left(k_p c_d + u_o\right)$$

$$= \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}\mathbf{c} + \begin{bmatrix} 0 & -b_o k_p \\ 0 & 0 \end{bmatrix}\mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix}\left(k_p c_d + u_o\right)$$

State feedback example



Separate the feedforward and feedback terms…

$$= \begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} 0 & -b_o k_p \\ 0 & 0 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(k_p c_d + u_o \right)$$

Add the matrices together…

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_o + k(\mathrm{c}_d) \right)$$

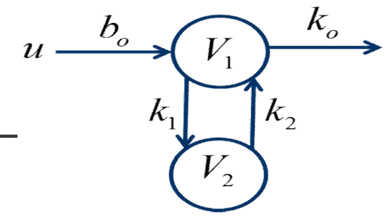
And this solution can be written more generally in the matrix form...

$$\frac{d\mathbf{c}}{dt} = [\mathbf{A} - \mathbf{B}\mathbf{K}]\mathbf{c} + \mathbf{B}\left(u_o + k_r \, \mathrm{c}_d \right)$$

We now have a set of feedback gains ***K****!*

$$\frac{d\mathbf{c}}{dt} = \left[\begin{bmatrix} -k_o - k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} - \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & k_p \end{bmatrix} \right] \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_o + k_r \, \mathrm{c}_d \right)$$

Is the state controlled system stable?



$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_o + k\left(\mathrm{c}_d\right)\right)$$

$$\frac{d\mathbf{c}}{dt} = \left[\mathbf{A} - \mathbf{B}\mathbf{K}\right]\mathbf{c} + \mathbf{B}\left(u_o + k_r \mathrm{c}_d\right)$$

Is our new "system" stable?

$$\det(\lambda \mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K})) = \det \begin{bmatrix} \lambda + k_o + k_1 & -k_2 - b_o k_p \\ -k_1 & \lambda + k_2 \end{bmatrix} = 0$$

$$= \lambda^2 + \left(k_0 + k_1 + k_2\right)\lambda + \left(k_0 k_2 + b_o k_2 k_p\right) = 0 \Rightarrow$$

System is stable for any $k_p > 0$!

*The eigenvalue (and thus stability) is now determined by the values of **K** since the set of first order differential equations we want to solve is **[A-BK]** and not just **A.***

System response with controller

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_d + k(y_d) \right)$$

```
[t,z] = ode45('twoVolume', [0 30], [0 0]);
plot(t, z);
```

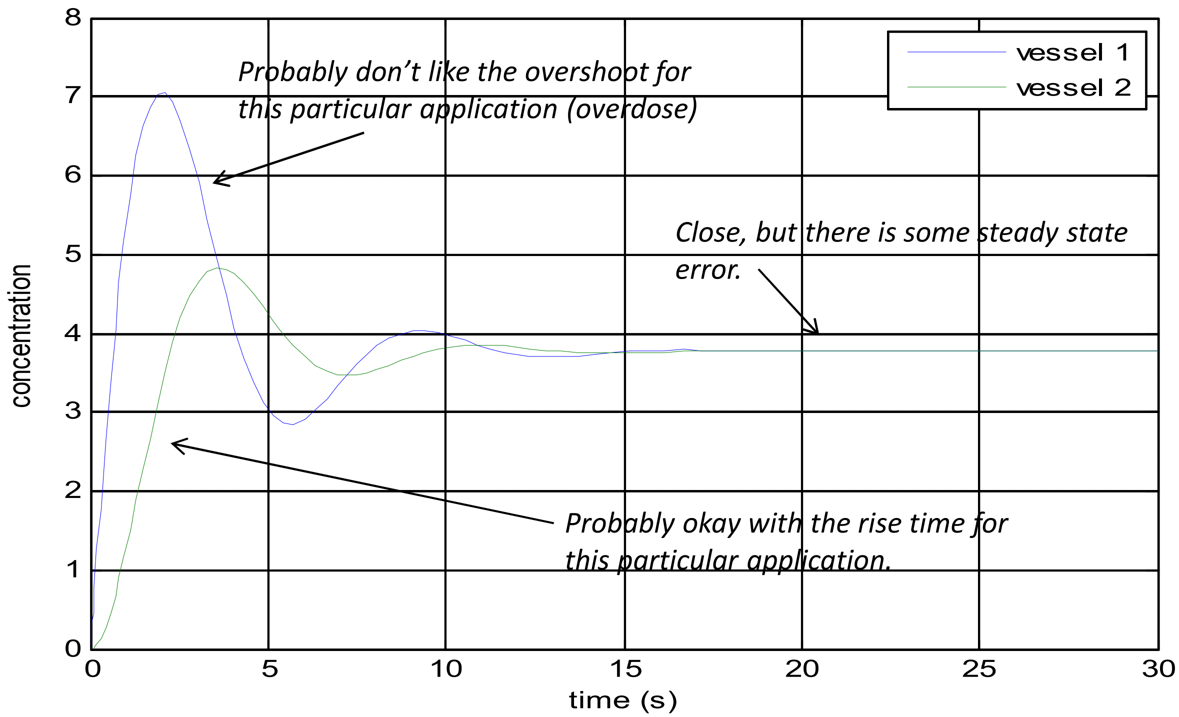
```
function cprime = twoVolume( t, c )

k0 = 0.1; k1 = 0.1;
k2 = 0.5; bo = 1.5;

kp = 1.1;
yd = 4.0;
ud = 0.0;

A = [ -k0-k1 k2-bo*k; k1 -k2 ];
B = [ bo; 0];
u = kp*yd + ud;

cprime = A*c + B*u;
```

system parameters

controller gain

desired output

default input

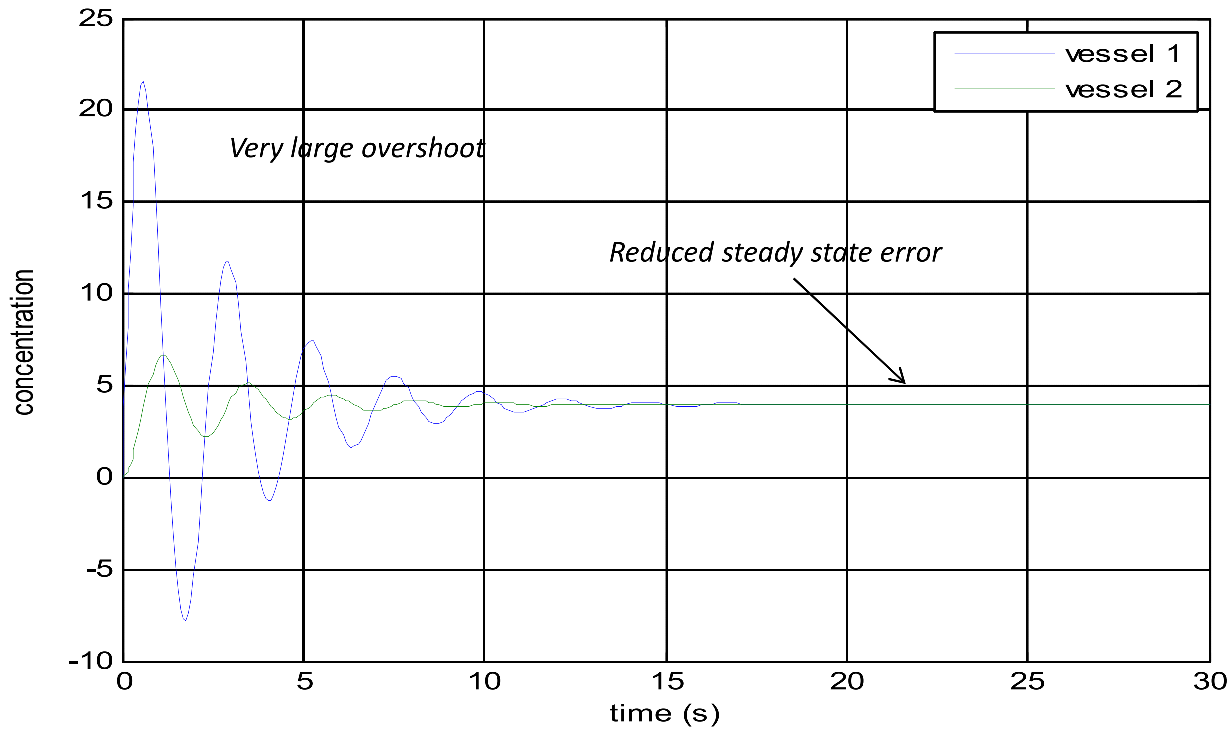


System response with controller

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_d + k(y_d) \right)$$

```
[t,z] = ode45('twoVolume', [0 30], [0 0]);
plot(t, z);


function cprime = twoVolume( t, c )

k0 = 0.1; k1 = 0.1;
k2 = 0.5; bo = 1.5;

kp = 10;
yd = 4.0;
ud = 0.0;

A = [ -k0-k1 k2-bo*k; k1 -k2 ];
B = [ bo; 0];
u = kp*yd + ud;

cprime = A*c + B*u;
```

increased gain



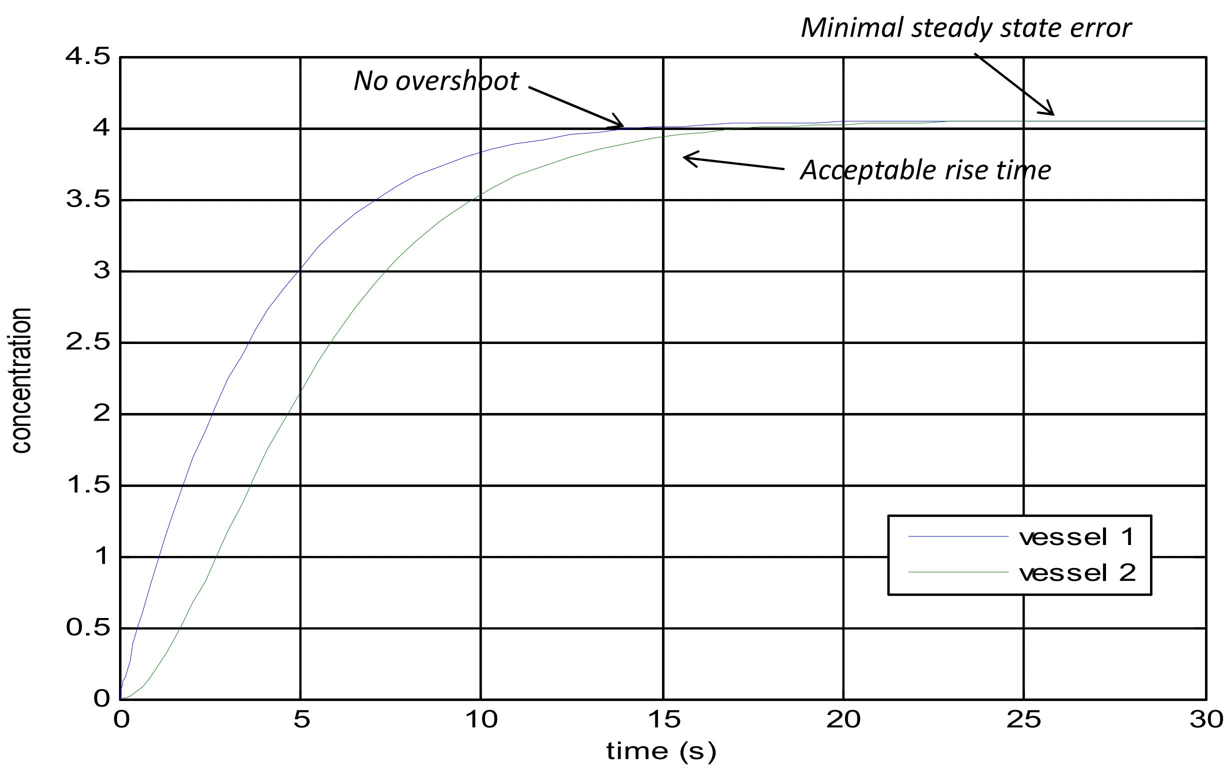
System response with controller

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_d + k(y_d) \right)$$

```
[t,z] = ode45('twoVolume', [0 30], [0 0]);
plot(t, z);


function cprime = twoVolume( t, c )

k0 = 0.1; k1 = 0.1;
k2 = 0.5; bo = 1.5;

k = .1;
yd = 4.0;
ud = 0.275;

A = [ -k0-k1 k1-bo*k; k2 -k2 ];
B = [ bo; 0];
u = k*yd + ud;

cprime = A*c + B*u;
```

decrease gain

Default input

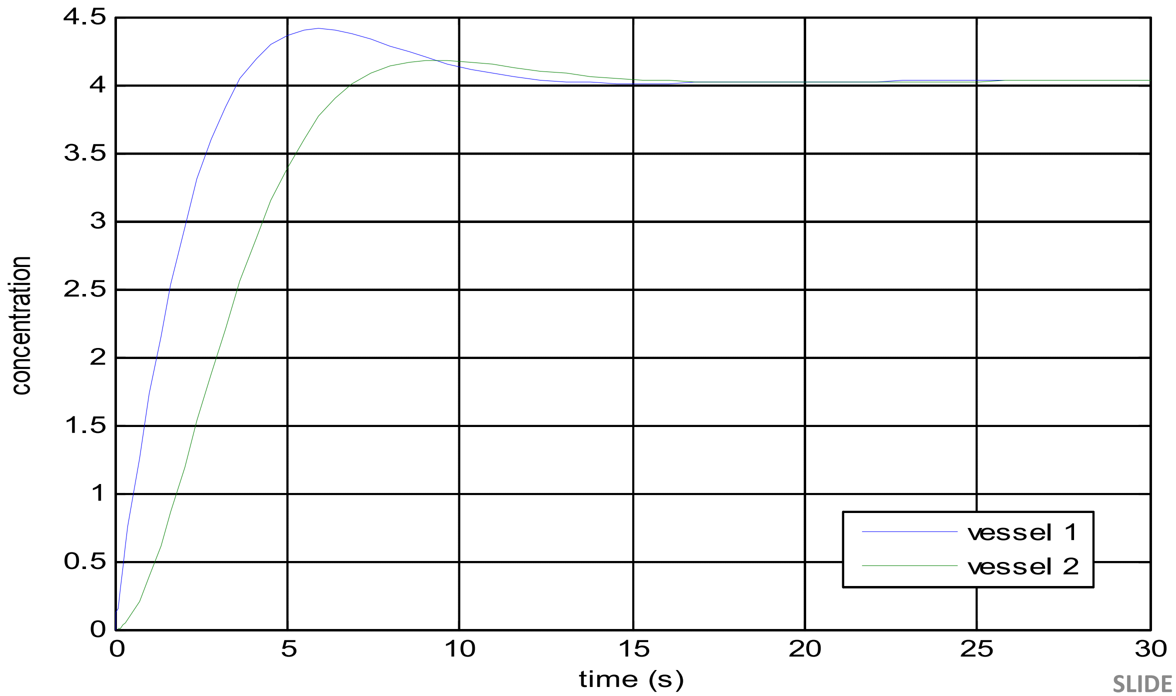


System response with controller

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_d + k(y_d)\right)$$

```
[t,z] = ode45('twoVolume', [0 30], [0 0]);
plot(t, z);


function cprime = twoVolume( t, c )

k0 = 0.1; k1 = 0.1;
k2 = 0.5; bo = 1.5;

k = .25;
yd = 4.0;
ud = 0.275;

A = [ -k0-k1 k1-bo*k; k2 -k2 ];
B = [ bo; 0];
u = k*yd + ud;

cprime = A*c + B*u;
```

slightly higher gain

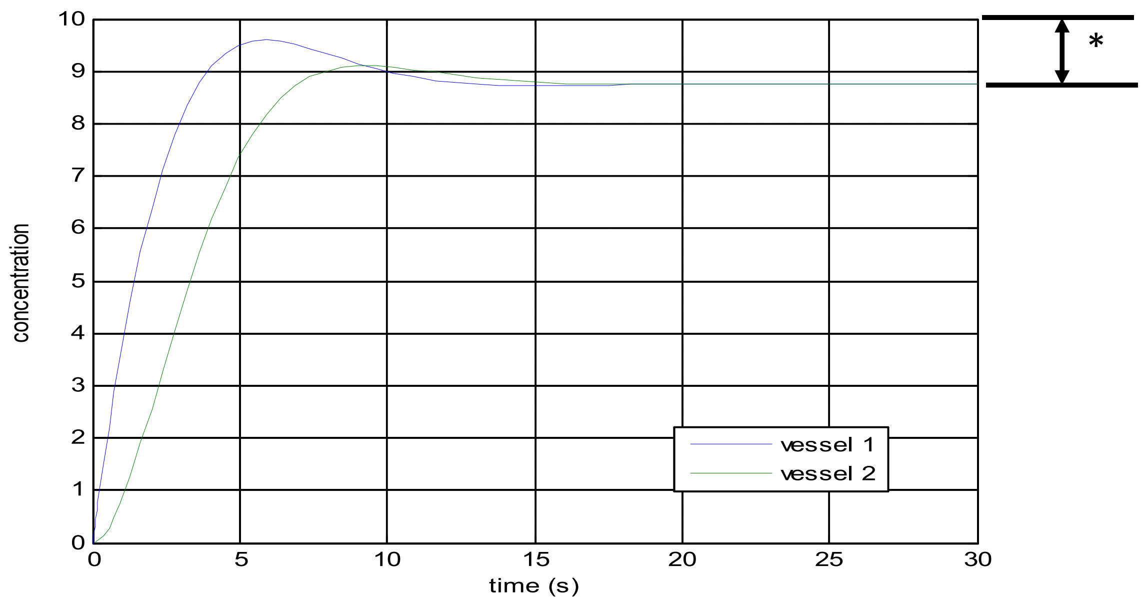


What if we change the c_d?

$$\frac{d\mathbf{c}}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} \mathbf{c} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_d + k(y_d) \right)$$

```
function cprime = twoVolume( t, c )

k0 = 0.1; k1 = 0.1;
k2 = 0.5; bo = 1.5;

k = .25;
yd = 10.0;
ud = 0.275;

A = [ -k0-k1 k1-bo*k; k2 -k2 ];
B = [ bo; 0];
u = k*yd + ud;

cprime = A*c + B*u;
```

Changed desired output to 10 from 4.



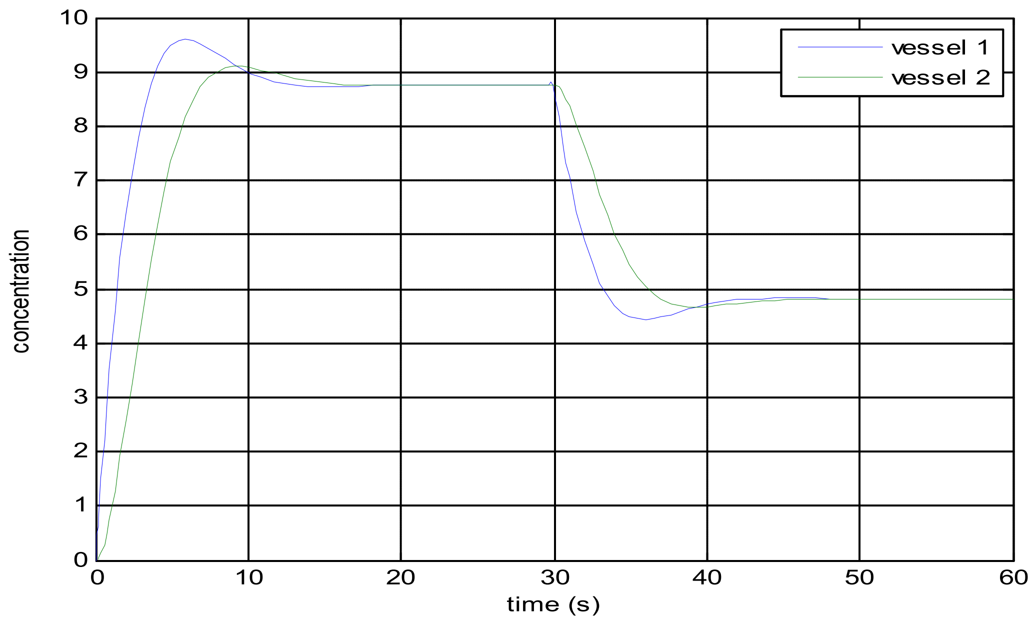
** steady state error increased. How to eliminate this is future topic.*

What if we dynamically change c_d?

$$\frac{dc}{dt} = \begin{bmatrix} -k_o - k_1 & k_2 - b_o k_p \\ k_1 & -k_2 \end{bmatrix} c + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \left(u_d + k(y_d) \right)$$

```
function cprime = twoVolume( t, c )

k0 = 0.1; k1 = 0.1;
k2 = 0.5; bo = 1.5;

k = .25;
if t < 30;
    yd = 10.0;
else
    yd = 5.0;
end
ud = 0.275;

A = [ -k0-k1 k2-bo*k; k1 -k2 ];
B = [ bo; 0];
u = k*yd + ud;

cprime = A*c + B*u;
```

Variable desired output



State Feedback Example

- What we learned…
  - Feedback made the system more robust
  - Allowed us to pick and change the concentration level (i.e. the state values)
    - The input value is determined by the controller
  - Trial & error is not necessary to find *u* every time the desired concentration (or other properties) change.
- But…
  - Used still trial & error to find one k *and* u_o, *and*
  - Trial & error for complicated systems may not be possible.
- What we will learn…
  - How to determine what systems are controllable,
  - to modify the eigenvalues w/ feedback to get the behavior we want,
  - Design controllers for a generalized system, and
  - How to eliminate steady state error.
- Our objective is to…
  - Determine if state feedback is possible,
  - Quantify a controller's performance, and

  Today

  - design and test state feedback controllers.

  Next Lesson

State Feedback – Defining Performance

- Stability

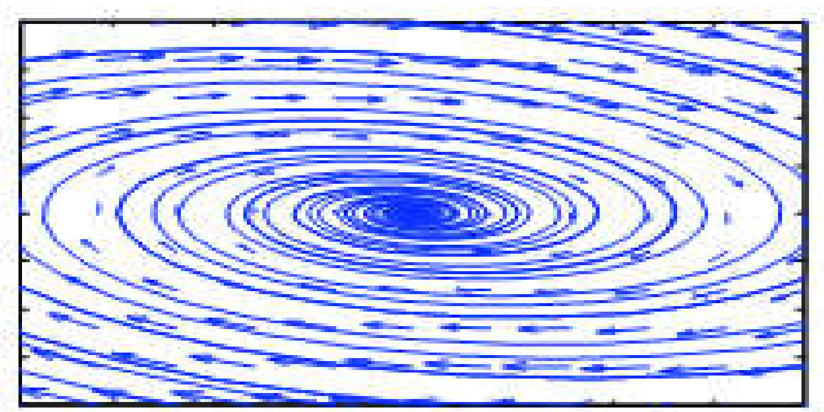
$$\lim_{t \to \infty} \mathbf{z}(t) = \mathbf{z}_e \forall \mathbf{z}(t_o) \in \mathbb{R}^n$$

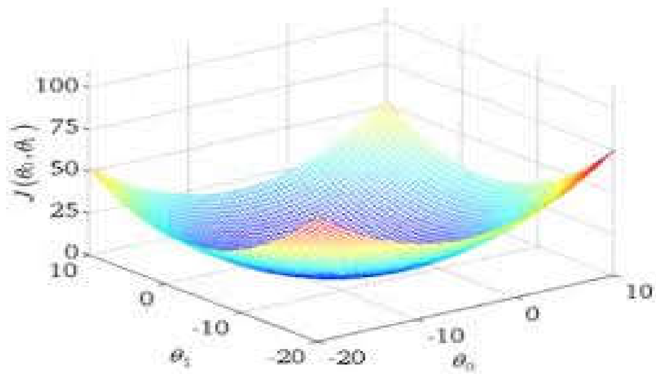
"The states of a system will approach equilibrium for the given initial states (global or local) (asymptotic or neutral)."

- Performance

find: $\mathbf{z}(t) \mid \min\left(\gamma_c\left(\mathbf{z}, u\right)\right)$

"Find a solution that minimizes a given performance criterion or criteria (i.e. minimize fuel consumed, minimize distance travelled, % overshoot, etc.)"





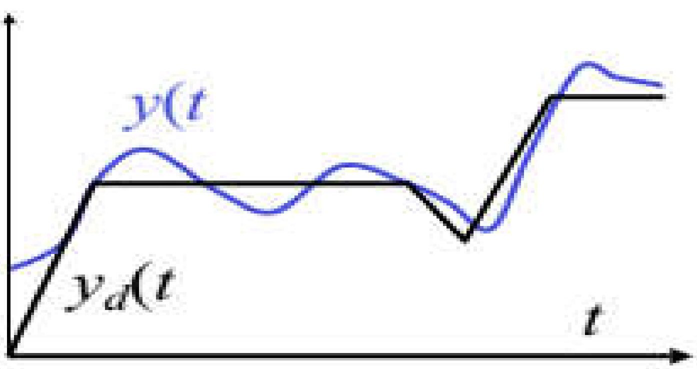
State Feedback – Key Definitions

- Tracking

given:

$$y_o(t)\exists u(\mathbf{z},t) \mid \lim_{t\to\infty}\left(y(t)-y_d(t)\right)=0\,\forall\,\mathbf{z}_o\in\mathbb{R}^n$$

"For a given output there exists an input that minimizes the error between the actual output and a desired output for every initial condition"

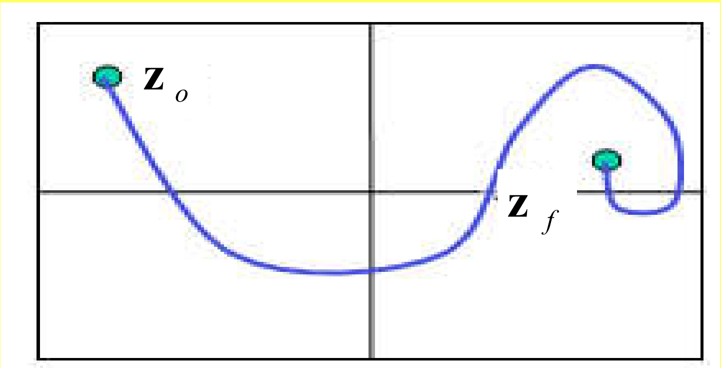


- Reachability (Controllability)

given: $\mathbf{z}_o,\mathbf{z}_f\in\mathbb{R}^n\exists u(t)\forall\,\dot{\mathbf{z}}=f(\mathbf{z},u)$

that takes: $\mathbf{z}_o\to\mathbf{z}(<T)=\mathbf{z}_f$

"Given an initial state and desired final state, there exists a controller that can attain the desired final states in a finite amount of time."



Reachability

A simple example…

Given:

$$\frac{dz_1}{dt} = -z_1 + u$$

$$\frac{dz_2}{dt} = -z_2 + u$$

where…

$$z_1(0) = 0$$

$$z_2(0) = 0$$

find…

$$u(t)$$

such that…

$$z_1(< T) = 1$$

$$z_2(< \mathrm{T}) = 2$$

Note: if I can attain the goal states in a finite amount of time, I can attain them in almost any amount of time assuming I can make my gain very large.

Solution:

There is none. The desired output is not reachable. Why not?

if z_1 and z_2 are initially the same, there is no input that will make them have final different values.

Reachability

Given a discrete time example…

$$\mathbf{z}[k+1] = \mathbf{A}\,\mathbf{z}[k] + \mathbf{B}\,u[k]$$

$$= \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{z}[k] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u[k] \qquad \mathbf{z}[0] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Are all the states reachable?

Solution: *Assume a unit input (i.e. u[k]=1).*

After one step:

$$\mathbf{z}[1] = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Reachability

Step 1

$$\mathbf{z}[1]=\begin{bmatrix}1&3&0&1\\1&0&1&0\\2&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}0\\0\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Step 2

$$\mathbf{z}[2]=\begin{bmatrix}1&3&0&1\\1&0&1&0\\2&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}+\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=\begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

But what if….

$$\mathbf{B}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}?$$

Step 3

$$\mathbf{z}[3]=\begin{bmatrix}1&3&0&1\\1&0&1&0\\2&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}+\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=\begin{bmatrix}2\\1\\2\\1\end{bmatrix}$$

Yes. All steps could be reached after three steps.

Reachability

Step 1

$$\mathbf{z}[1] = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 2

$$\mathbf{z}[2] = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \end{bmatrix}$$

Step 3

$$\mathbf{z}[3] = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 4 \\ 2 \end{bmatrix}$$

Yes. Again all states could be reached after three steps.

But what if….

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}?$$

Reachability

Step 1

$$\mathbf{z}[1] = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Step 2

$$\mathbf{z}[2] = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

Step 3

$$\mathbf{z}[3] = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \end{bmatrix}$$

No. Doesn't look like all states are reachable.

Step 1

$$\mathbf{z}[1] = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 2

$$\mathbf{z}[2] = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \end{bmatrix}$$

Step 3

$$\mathbf{z}[3] = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 4 \\ 2 \end{bmatrix}$$

But with a different B, this system IS reachable!

Reachability

Is there a quick way to determine reachability for a system in general?

Given a state space model and general solution…

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u \qquad \mathbf{z} \in \mathbb{R}^n$$
$$y = \mathbf{C}\mathbf{z} + \mathbf{D}u \qquad \mathbf{z}(0) = \mathbf{z}_o$$
$$u \in \mathbb{R}^1,\ \mathrm{y} \in \mathbb{R}^1$$

$$\Rightarrow \qquad \mathbf{z}(t) = e^{\mathbf{A}t}\mathbf{z}\big(0\big) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)\,\mathrm{d}\tau$$

Theorem: A linear system is reachable if and only if the Reachability Matrix (w_r) is full rank (i.e. invertible)

$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

- *Very simple. Easy to apply. (*MATLAB ctrb(A,B)*)*
- *If satisfied, we can assert that "the system (A,B) is reachable."*
- *The proof is difficult, but can get an idea of where it comes from…*

Reachability

Theorem: A linear system is reachable if and only if the Reachability Matrix (w_r) is full rank (i.e. invertible)

$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

Proof(?): The proof is difficult, but can get an idea of where it comes from…

Start with the general solution…

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

Only care about what we can impact with an input.

$$\mathbf{z}(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)\,\mathrm{d}\tau$$

Let's assume an impulse input….

$$\mathbf{z}_\delta(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\,\delta(\tau)\,\mathrm{d}\tau = e^{\mathbf{A}t}\mathbf{B}$$

Similarly, the response to the derivative of the impulse function is…

$$\mathbf{z}_{\dot\delta}(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\,\dot\delta(\tau)\,\mathrm{d}\tau = \mathbf{A}\,e^{\mathbf{A}t}\mathbf{B}$$

(etc.)

Does the derivative of an impulse (Dirac) function exist? Yes.

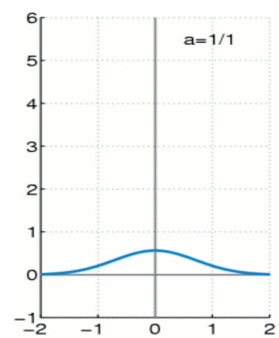
Why? Because the "engineering" definition is a cheat….

$$\delta(x) = \begin{cases} \infty & t = t_0 \\ 0 & t \neq t_0 \end{cases}$$

It only makes sense inside an integral

$$\int_{-\infty}^{\infty} \delta(t)\,dt = 1$$

The Dirac function is more formally defined as a distribution as the width goes to zero.



Reachability

Theorem: A linear system is reachable if and only if the Reachability Matrix (w_r) is full rank (i.e. invertible)

$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

Proof(?): The proof is difficult, but can get an idea of where it comes from…

So let's use the linearity and the principle of superposition to consider the following input.

$$\mathrm{u}(t) = \alpha_1 \delta(t) + \alpha_2 \dot{\delta}(t) + \alpha_3 \ddot{\delta}(t) + \cdots + \alpha_n \delta^{(n-1)}(t)$$

Plug in our solution to the ODEs for each input...

$$\mathbf{z}(t) = \alpha_1 e^{\mathbf{A}t}\mathbf{B} + \alpha_2 \mathbf{A} e^{\mathbf{A}t}\mathbf{B} + \alpha_3 \mathbf{A}^2 e^{\mathbf{A}t}\mathbf{B} + \cdots + \alpha_n \mathbf{A}^{n-1} e^{\mathbf{A}t}\mathbf{B}$$

Take the limit as t goes to zero…

$$\mathbf{z}(t) = \alpha_1 \mathbf{B} + \alpha_2 \mathbf{A}\mathbf{B} + \alpha_3 \mathbf{A}^2\mathbf{B} + \cdots + \alpha_n \mathbf{A}^{n-1}\mathbf{B}$$

So to reach an arbitrary set of states z(t), we must use some combination of these inputs….

$$\mathbf{z} = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_n \end{bmatrix} \quad \Rightarrow \quad w_r^{-1}\mathbf{z} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Examples revisited

$$\dot{\mathbf{z}} = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

```
clear all
A = [ 1 3 0 1;
    1 0 1 0;
    2 0 0 0;
    0 0 1 0 ];
B = [ 0 0 0 1 ]';

wr = [B A*B A*A*B A*A*A*B]
rank(wr)
```

```
>> Scratch
wr =
     0     1     1     4
     0     0     1     3
     0     0     2     2
     1     0     0     2

ans = 4
```

$$\dot{\mathbf{z}} = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

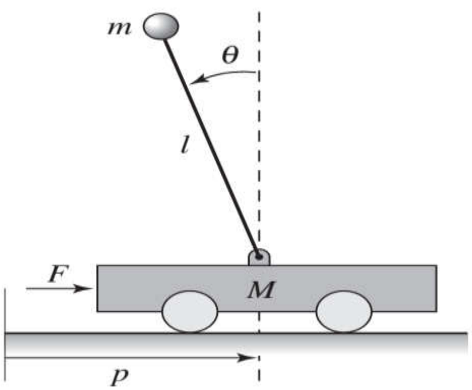
$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

```
clear all
A = [ 1 3 0 0;
    1 0 1 0;
    2 0 0 0;
    0 0 1 1 ];
B = [ 0 0 0 1 ]';

wr = ctrb( A, B )
rank( wr )
```

```
>> Scratch
wr =
     0     0     0     0
     0     0     0     0
     0     0     0     0
     1     1     1     1

ans = 1
```

Inverted Pendulum Example

Determine: If the inverted pendulum system shown below is controllable if the pendulum is initially perpendicular and above the platform if the mast has a length l.



Solution:

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

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

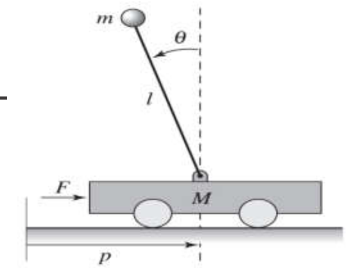
F is the input, linearize at θ =0º (i.e. cos(θ)=1 & sin(θ)= θ.)

$$(M + m)\ddot{x} = ml(1)\ddot{\theta} - (0)\dot{x} - ml\theta\dot{\theta}^2 + u$$
$$(J + ml^2)\ddot{\theta} = ml(1)\ddot{x} - (0)\dot{\theta} + mgl\theta$$

Put in matrix form…

$$\begin{bmatrix} (M + m) & -ml \\ -ml & (J + ml^2) \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -ml\theta\dot{\theta}^2 \\ mgl\theta \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix}$$

Inverted pendulum example



$$\begin{bmatrix} (M+m) & -ml \\ -ml & (J+ml^2) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -ml\theta\dot{\theta}^2 \\ mgl\theta \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix}$$

When controlled, the angular velocity should be close to zero, so we can ignore terms quadratic and higher angular velocity terms.

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

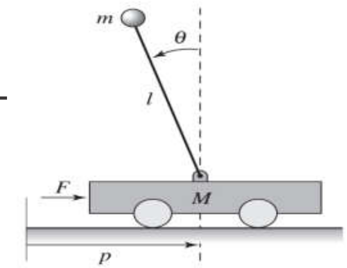
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{(M+m)(J+ml^2)-m^2l^2} \begin{bmatrix} (J+ml^2) & cml \\ ml & (M+m) \end{bmatrix} \left[\begin{bmatrix} 0 \\ mgl\theta \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix} \right]$$

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

For, control purposes, the measured values are x and ϑ, so let's define our states as.

$$\mathbf{z} = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T$$

Inverted Pendulum Example



In terms of our states, our outputs are:

$$\mathbf{y} = \mathbf{C}\,\mathbf{z} + \mathbf{D}\,u = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}$$

And our system is….

$$\mathbf{z} = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T$$

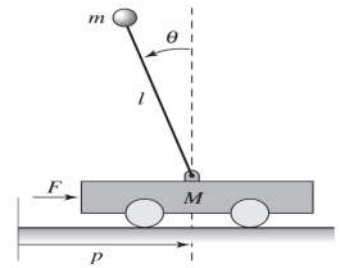
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{\mu} \begin{bmatrix} (J + m l^2) & -m l \\ m l & (M + m) \end{bmatrix} \left[\begin{bmatrix} 0 \\ m g l \theta \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix} \right]$$

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

And so, back to our question, is the system reachable (i.e. controllable?). Since n=4, we have…

$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\,\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^3\mathbf{B} \end{bmatrix}$$

Inverted Pendulum Example



Our system once again….

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

Examine the determinant to determine when the system will not be full rank.

Plugging into our reachability matrix…

$$\det(w_r) = \frac{g^2 l^4 m^4}{\mu^4} \neq 0$$

$$w_r = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix}$$

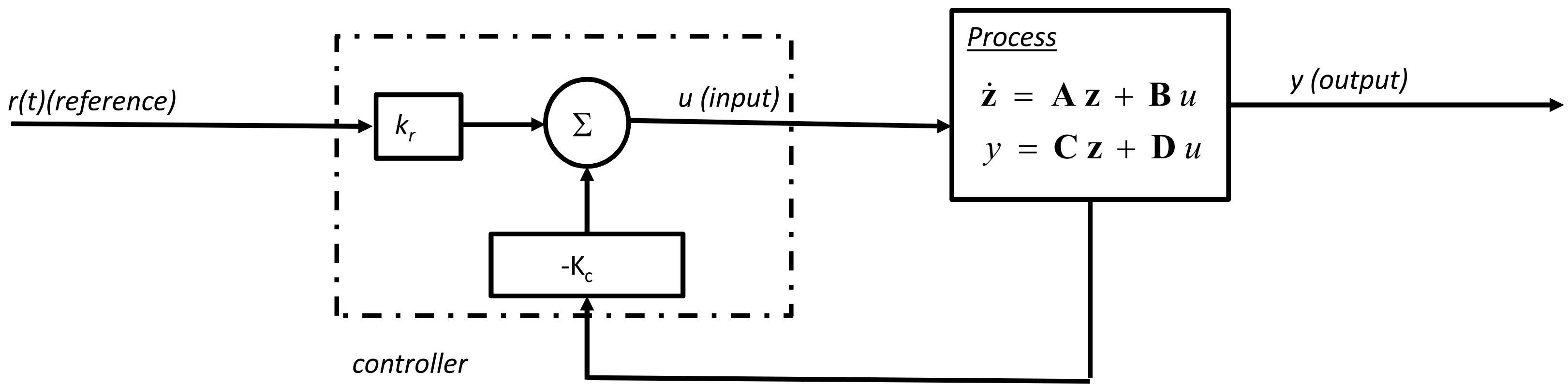
$$= \begin{bmatrix} 0 & \frac{J+ml^2}{\mu} & 0 & \frac{gl^3m^3}{\mu^2} \\ 0 & \frac{lm}{\mu} & 0 & \frac{gl^2m^2(m+M)}{\mu^2} \\ \frac{J+ml^2}{\mu} & 0 & \frac{gl^3m^3}{\mu^2} & 0 \\ \frac{lM}{\mu} & 0 & \frac{gl^2m^2(m+M)}{\mu^2} & 0 \end{bmatrix}$$

Where…

$$\mu = (M+m)(J+ml^2) - m^2 l^2$$

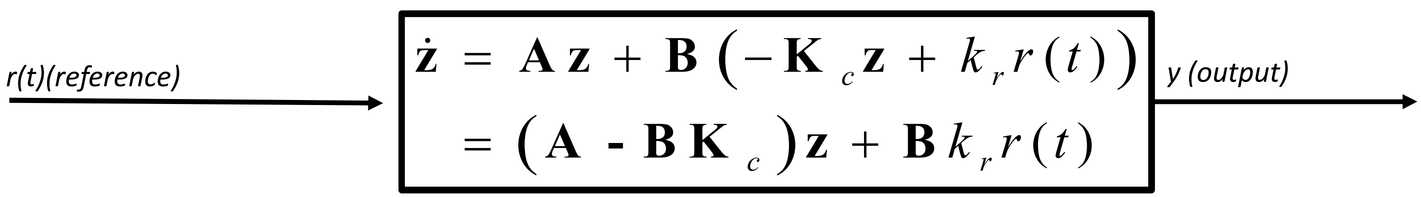
So, physically, when is the system NOT reachable?

State feedback control summary



The generalized state feedback control input is

$$u = -\mathbf{K}_c z + k_r y_r$$



Goal: For systems that are reachable, find K_c such that the system is stable and behaves how we want it to.

$$p(s) = \left(\lambda\mathbf{I} - \left[\mathbf{A} - \mathbf{B}\mathbf{K}_\mathbf{c}\right]\right) = 0 \qquad \textit{where...} \qquad \mathrm{Re}(\lambda_i) < 0$$

Note: stability is the baseline performance measure.