State Feedback

Dr. Mitch Pryor

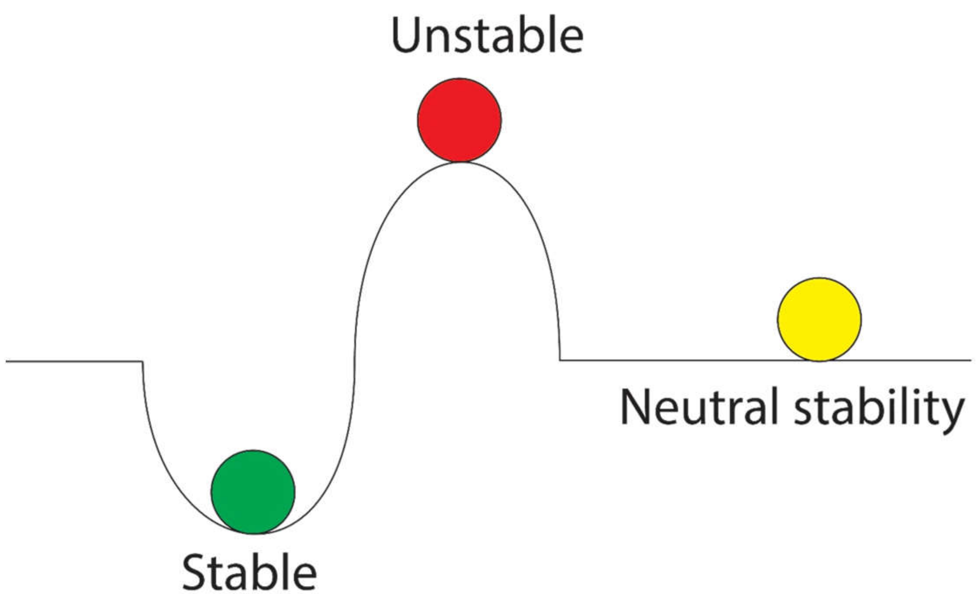
Lesson Objectives

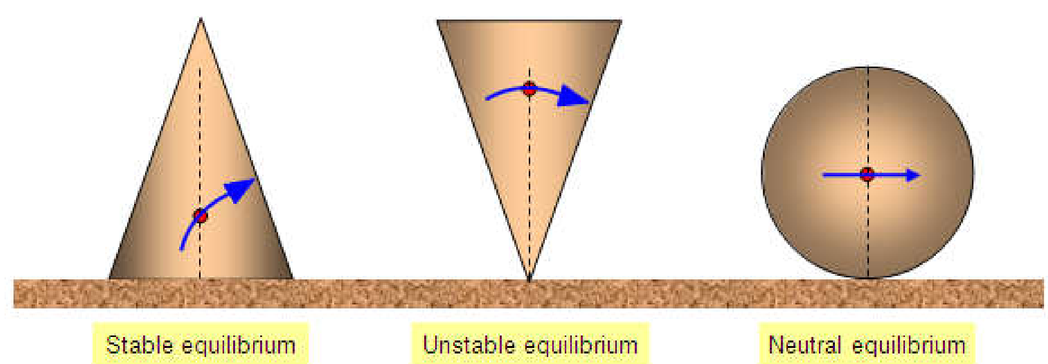
- Formally define stability
- For linear systems
  - We have already seen a strong pattern between a system's eigenvalues. In this lesson, we will:
    - Review system response with respect to stability
    - Provide additional insight for determining which systems are stable (or under what conditions).
- For nonlinear systems
  - Linearization and stability
  - Lyapunov functions for evaluating stability (not covered here)
- Example of how we can use state values in our input to make a system stable.

Formal definition of stability

A system is <u>stable</u> if that system's response stays arbitrarily near some value, $\mathbf{z}_a$, for all of time greater than some value, t_f.

$$\left\|\mathbf{z}_a - \mathbf{z}_b\right\| < \delta \Rightarrow \left\|\mathbf{z}(t;\mathbf{z}_b) - \mathbf{z}(t;\mathbf{z}_a)\right\| < \varepsilon \quad \text{for all } t > 0$$

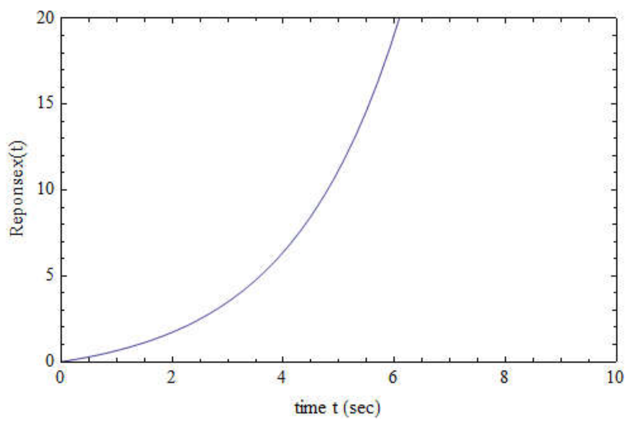




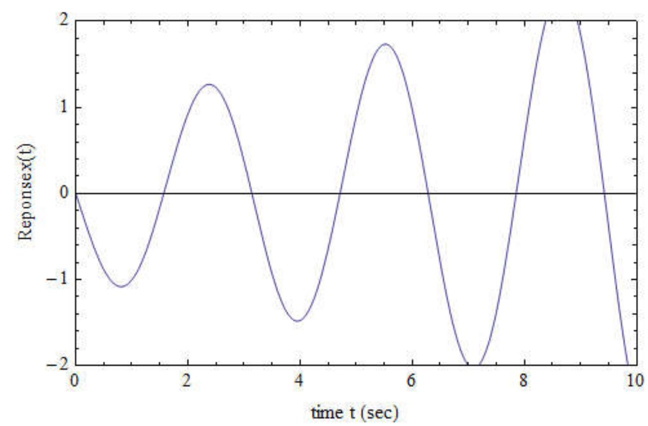
http://www.schoolphysics.co.uk/age11-14/Mechanics/Statics/text/Stability_/index.html

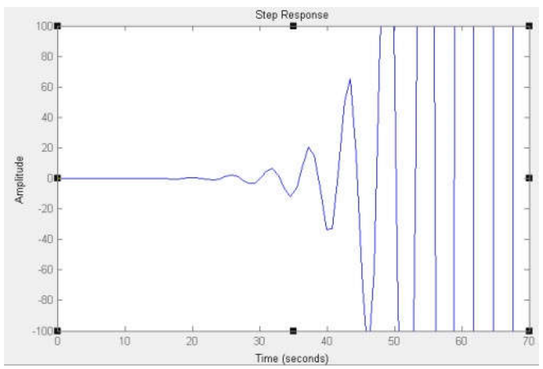
Unstable, neutrally stable responses

$$e^{+\alpha t}$$

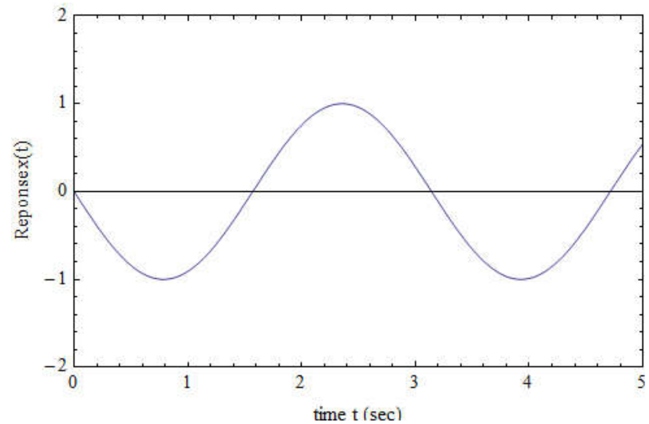


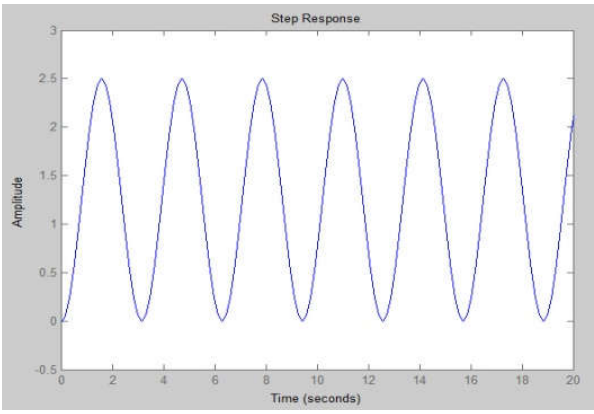
$$e^{+\alpha t}\left(c_1 \sin bt + c_2 \cos bt\right)$$





$$e^{0t}\ \left(c_1 \sin bt + c_2 \cos bt\right)$$





Common 2nd order example

Given:

$$\ddot{x} + a_1\dot{x} + a_2 x = bu(t)$$

Where:

$$x(0) = w \quad b = 1$$

$$\dot{x}(0) = v \quad u = \left.\begin{matrix} 0 \\ 1 \end{matrix}\right\} \begin{matrix} t < t_0 \\ t \ge t_0 \end{matrix}$$

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

Find:

what happens as a_1 and a_2 vary?

Solve:

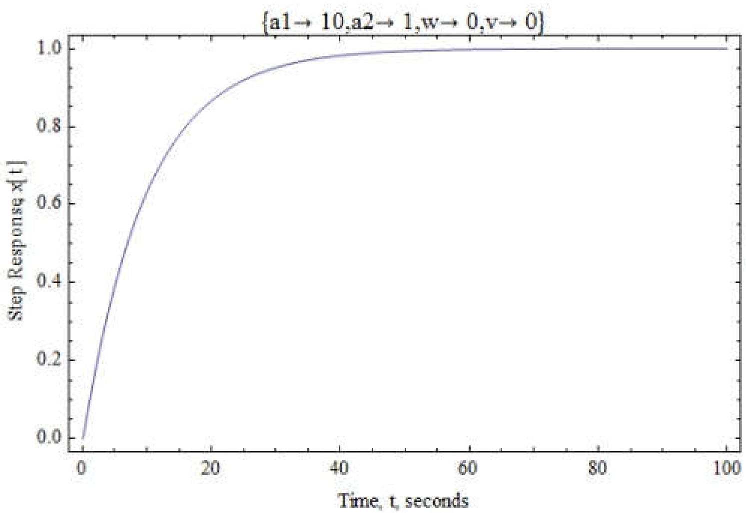
Using methods from previous lessons:

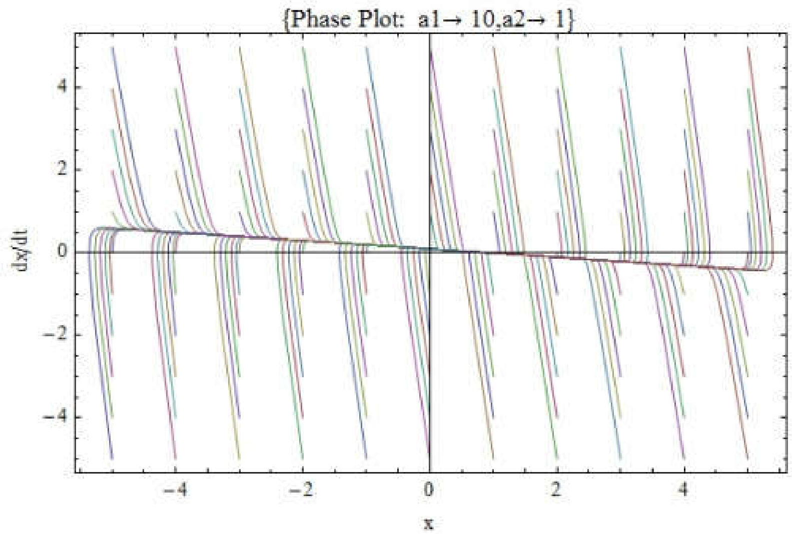
$$y(t) = \frac{1}{a_2} + C_1 e^{\frac{1}{2}(-a_1 - \sqrt{a_1^2 - 4a_2})t} + C_2 e^{\frac{1}{2}(-a_1 + \sqrt{a_1^2 - 4a_2})t}$$

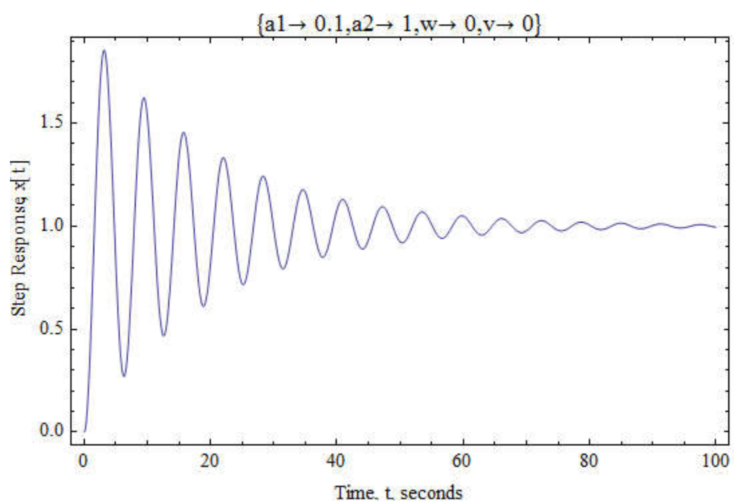
Which is used to generate the following examples for a variety of system parameters and initial conditions that illustrate common stability modalities.

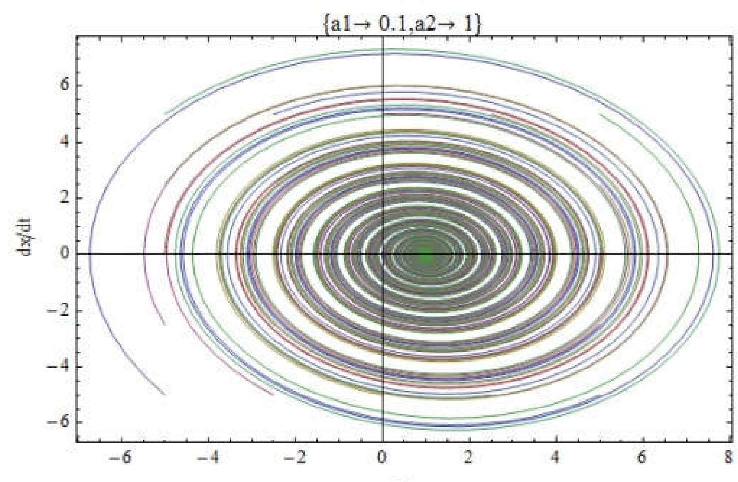
Asymptotically Stable Examples

$$x(t) = \frac{1}{a_2} + C_1 e^{\frac{1}{2}(-a_1 - \sqrt{a_1^2 - 4a_2})t} + C_2 e^{\frac{1}{2}(-a_1 + \sqrt{a_1^2 - 4a_2})t}$$



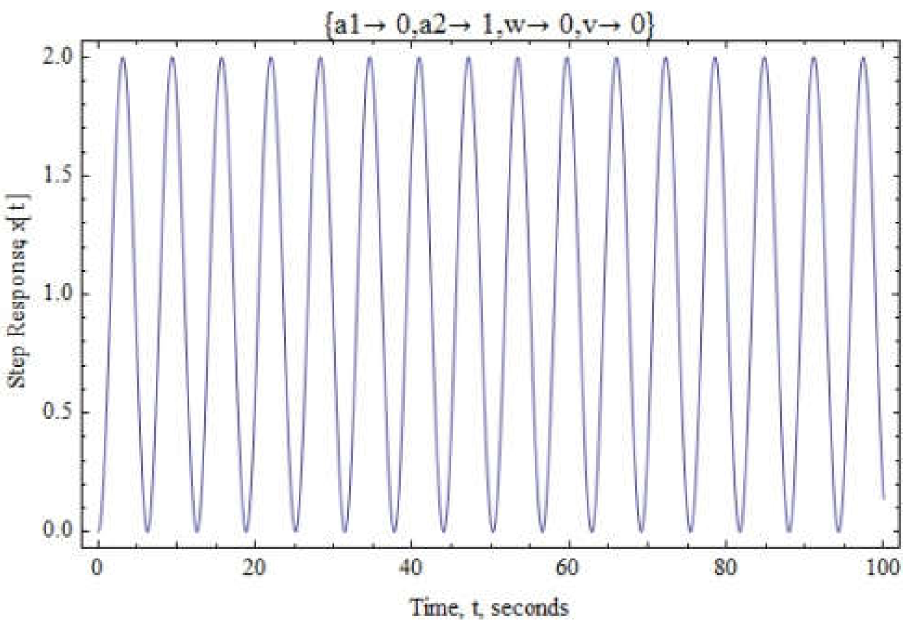


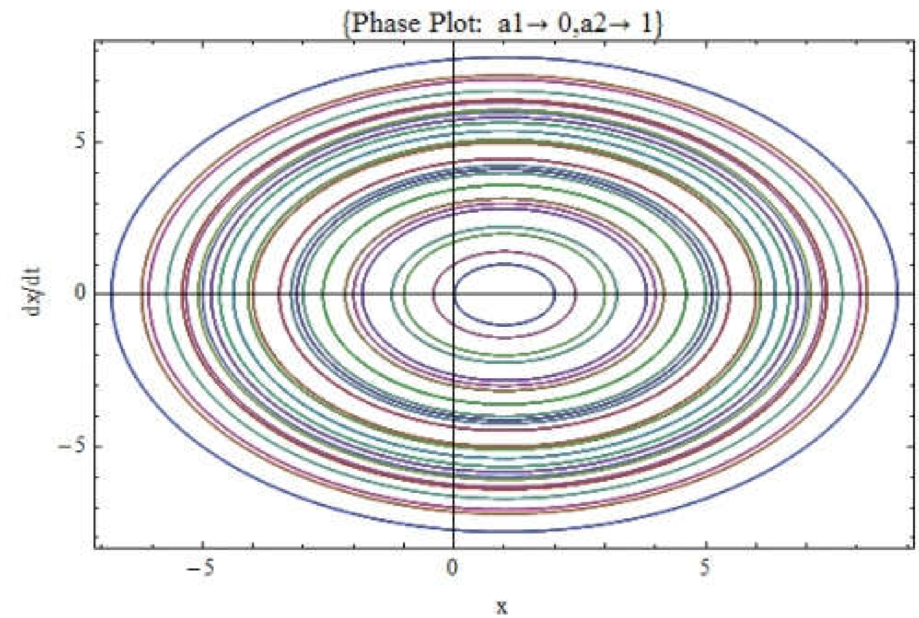




Neutrally stable examples

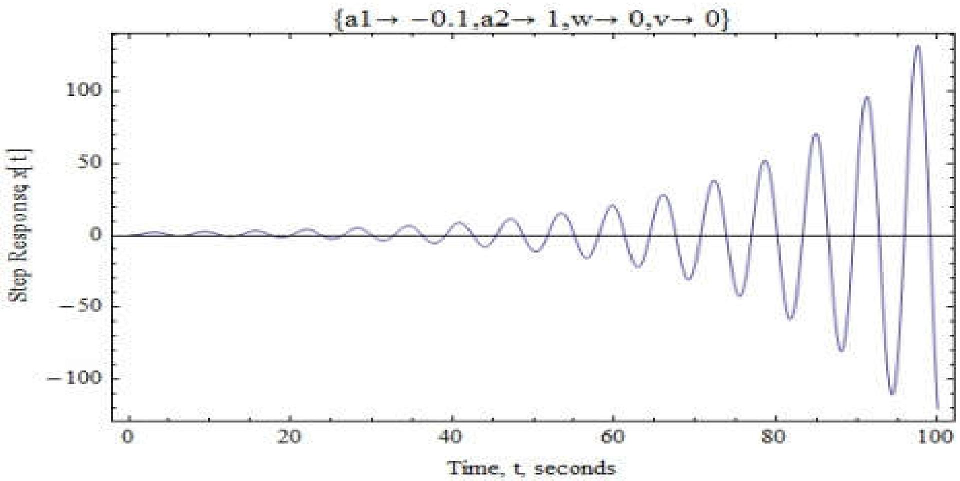
$$x(t) = \frac{1}{a_2} + C_1 e^{\frac{1}{2}(-a_1 - \sqrt{a_1^2 - 4a_2})t} + C_2 e^{\frac{1}{2}(-a_1 + \sqrt{a_1^2 - 4a_2})t}$$

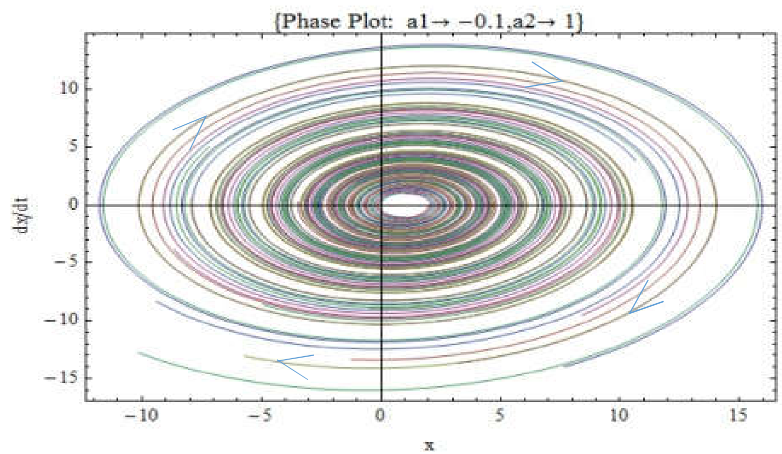


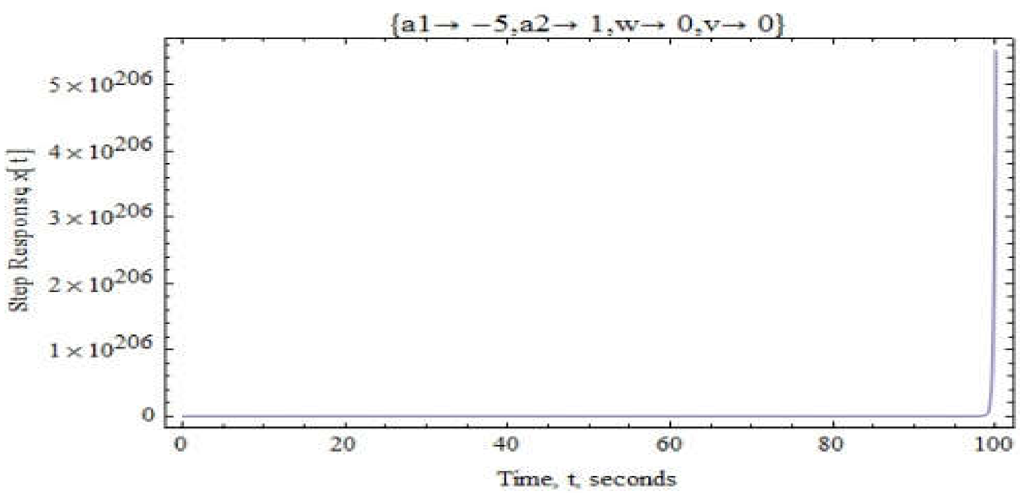


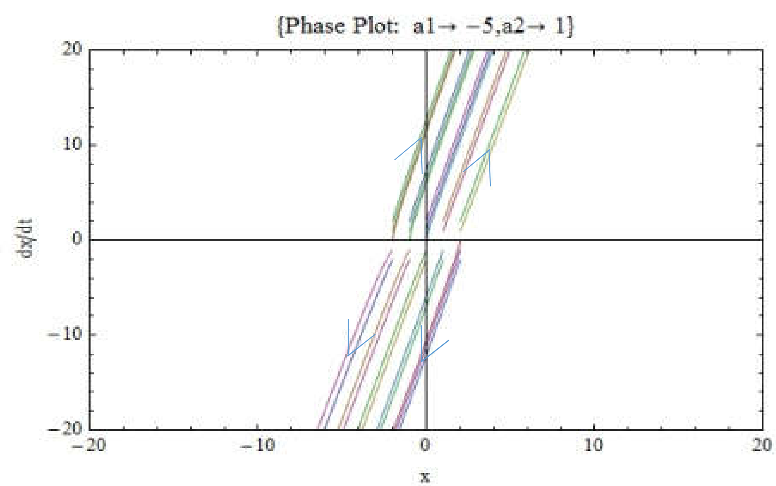
Unstable examples

$$x(t)=\frac{1}{a_2}+C_1 e^{\frac{1}{2}(-a_1-\sqrt{a_1^2-4a_2})t}+C_2 e^{\frac{1}{2}(-a_1+\sqrt{a_1^2-4a_2})t}$$









Stability in higher order systems

Example: For what values of α (if any) is the following system stable?

$$\frac{d\mathbf{z}}{dt}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & \alpha & -2\end{bmatrix}\mathbf{z}+\begin{bmatrix}0\\0\\1\end{bmatrix}u$$

$$y=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\mathbf{z}+\begin{bmatrix}4\end{bmatrix}u$$

Solve:

$$\det\left(\boldsymbol{\lambda}\mathbf{I}-\mathbf{A}\right)=\begin{bmatrix}\lambda & -1 & 0\\ 0 & \lambda & -1\\ 0 & -\alpha & \lambda+2\end{bmatrix}\mathbf{z}$$

$$\lambda\left(\lambda^2+2\lambda-\alpha\right)+0=0$$

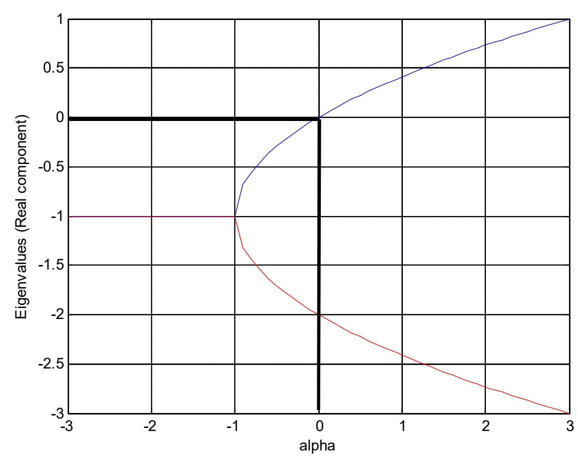
Therefore,

$$\lambda_1=0 \qquad \lambda_{2,3}=\frac{-2\pm\sqrt{4+4\alpha}}{2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

$$=\frac{-2\pm2\sqrt{1+\alpha}}{2}$$

$$=-1\pm\sqrt{1+\alpha}$$

The system is – at best – neutrally stable. Is there a range where the system is unstable?




```
a=[-3:.1:3];
for i=1:length(a)
    eig1(i) = real(-1+sqrt(1+a(i)));
    eig2(i) = real(-1-sqrt(1+a(i)));
end
```

So the system becomes unstable if α > 0.

So far…

- Presented formal definition of stability
- For Linear Systems
  - We have seen many examples.
  - Stability can be determined with respect to system parameters.
    - But method can get burdensome.
    - Note that "***all coefficients of the Characteristic Equation must be nonzero and have the same sign***" in order for the system to be asymptotically stable.
    - This is a <u>necessary</u>, but NOT <u>sufficient</u> condition for stability.

$$6\lambda^5 - 5\lambda^4 + 3\lambda^3 + 2\lambda^2 + 2\lambda + 3 = 0 \leftarrow NOT\_asymptotically\_stable$$

$$6\lambda^5 + 5\lambda^4 + 3\lambda^3 + 2\lambda^2 + 3 = 0 \leftarrow NOT\_asymptotically\_stable$$

$$6\lambda^5 + 5\lambda^4 + 3\lambda^3 + 2\lambda^2 + 2\lambda + 3 = 0 \leftarrow MAYBE\_asymptotically\_stable$$

- Still need to deal with stability of nonlinear systems.

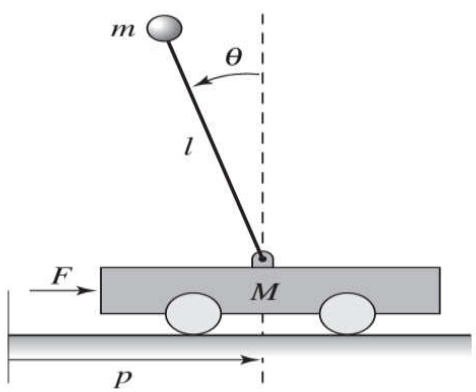
Nonlinear Systems: Multiple Options

- Determining stability for nonlinear systems using linearization.
- Exploit assumption that a system is properly controlled.
  - This allows us to treat some nonlinear systems as linear.
- Apply Lyapunov stability analysis to determine if a solution to a nonlinear dynamical system is stable.

Alexandr Lyapunov (1857-1918)

Inverted Pendulum Example

Given: A inverted pendulum on a moving cart:



Determine: If the inverted pendulum system shown is <u>stable</u> if the pendulum is initially perpendicular to the ground.

Solution:

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

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

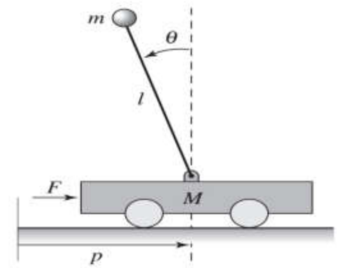
F is the input, linearize at θ =0º (i.e. cos(θ)=1 & sin(θ)= θ.)

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

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.

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

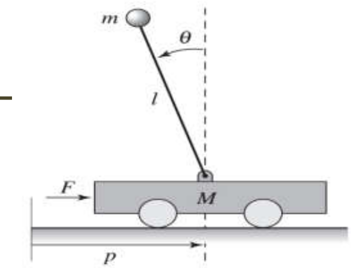
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{(M+m)(J+ml^2)-m^2l^2}\begin{bmatrix} (J+ml^2) & ml \\ ml & (M+m) \end{bmatrix}\left[\begin{bmatrix} 0 \\ mgl\theta \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix}\right]$$

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

Let's define the states as.

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

Inverted Pendulum Example



Note, in this case that: $\mathbf{y} = \mathbf{C}\,\mathbf{z} + \mathbf{D}\,u = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}\mathbf{z}$

And our system is….

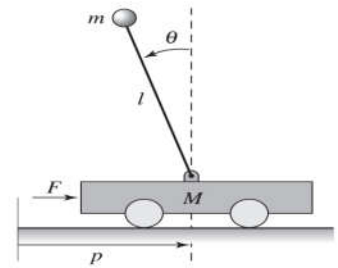
$$\mathbf{z} = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T$$

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

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

This system is linearized at θ=0 *assuming that the angular velocity is small. So is the system stable?*

Inverted Pendulum Example



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

This system is linearized at θ=0 *assuming that the angular velocity is small. So is the system stable?*

$$\det(\boldsymbol{\lambda}\mathbf{I} - \mathbf{A}) = \det \begin{bmatrix} \lambda & 0 & -1 & 0 \\ 0 & \lambda & 0 & -1 \\ 0 & -\frac{m^2 l^2 g}{\mu} & \lambda & 0 \\ 0 & -\frac{(M+m)mgl}{\mu} & 0 & \lambda \end{bmatrix}$$

$$\begin{aligned} CE &= \lambda\left(\lambda\left(\lambda^2\right) - 1\left(-\frac{(M+m)mgl}{\mu}\lambda\right)\right) - 1(0) \\ &= \lambda^4 - \lambda^2\,\frac{(M+m)mgl}{\mu} \end{aligned}$$

From this we get that the system's eigenvalues at this equilibrium point are:

$$\lambda = 0, 0, \pm\sqrt{\frac{(M+m)mgl}{\mu}}$$

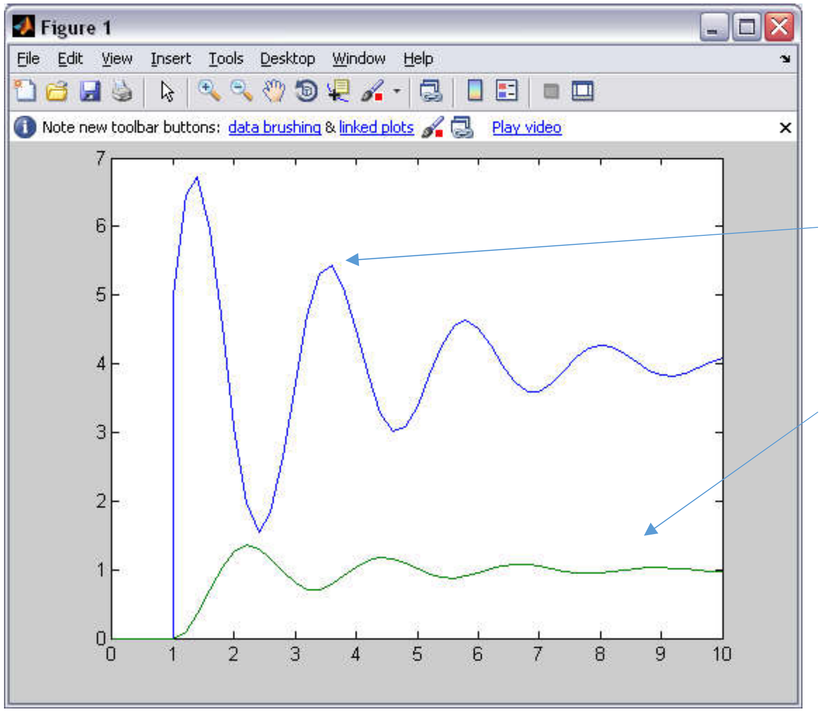
Therefore the system is unstable for any mechanical system qualifying as a pendulum!

Next 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**).

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

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



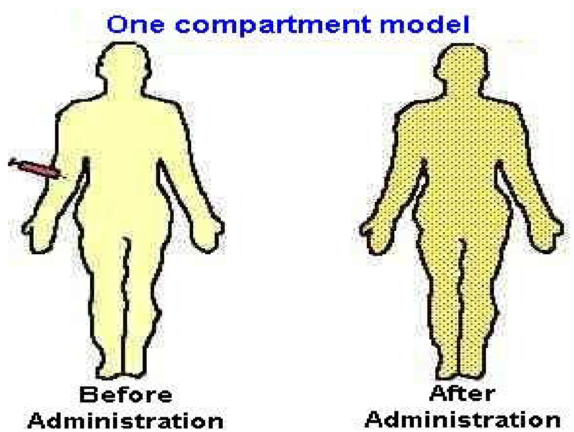
An input u (blue)…

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

How to pick **u**?

We have forshadowed our approach by looking at how varying elements of A impact stability….

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)

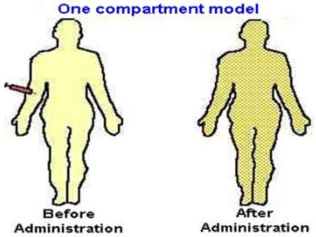
Consider two input options

Administered with a shot…

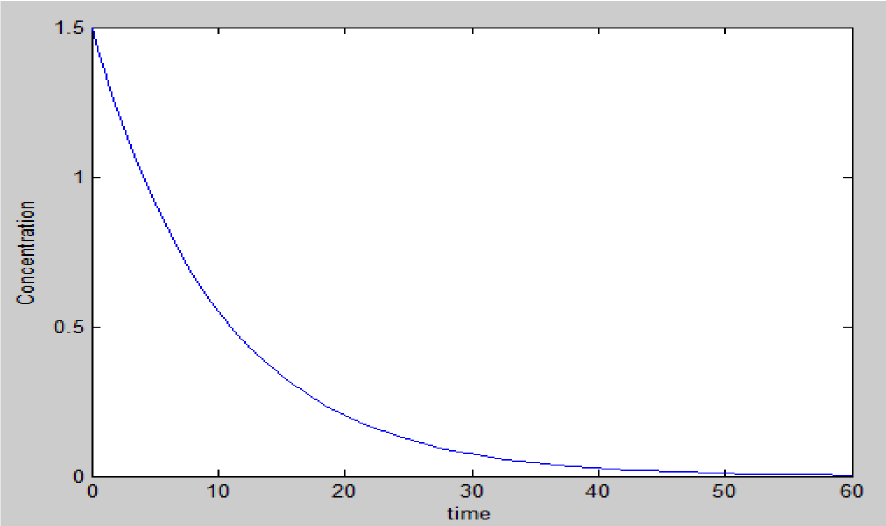
$$u(t) = \begin{cases} u_o & t = t_0 \\ 0 & t \neq 0 \end{cases}$$

or intravenously…

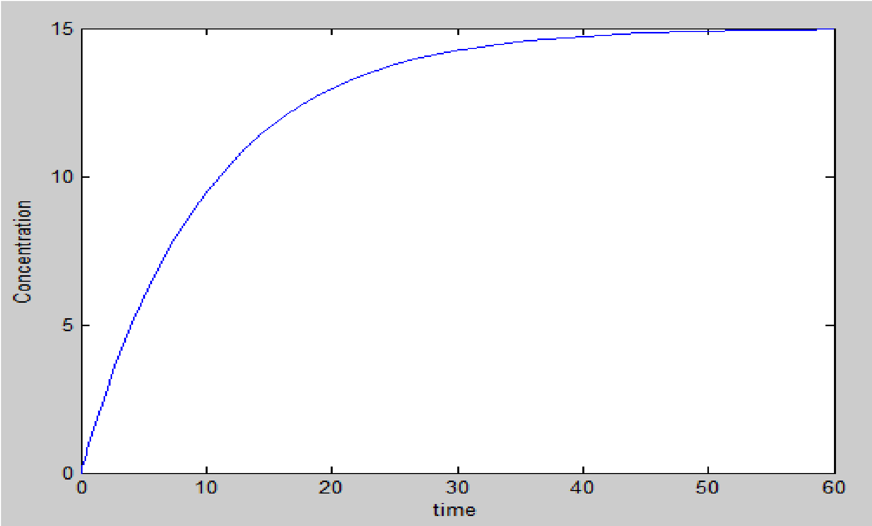
$$u(t) = \begin{cases} 0 & t < t_0 \\ u_o & t \geq 0 \end{cases}$$



a.k.a. the Impulse Function

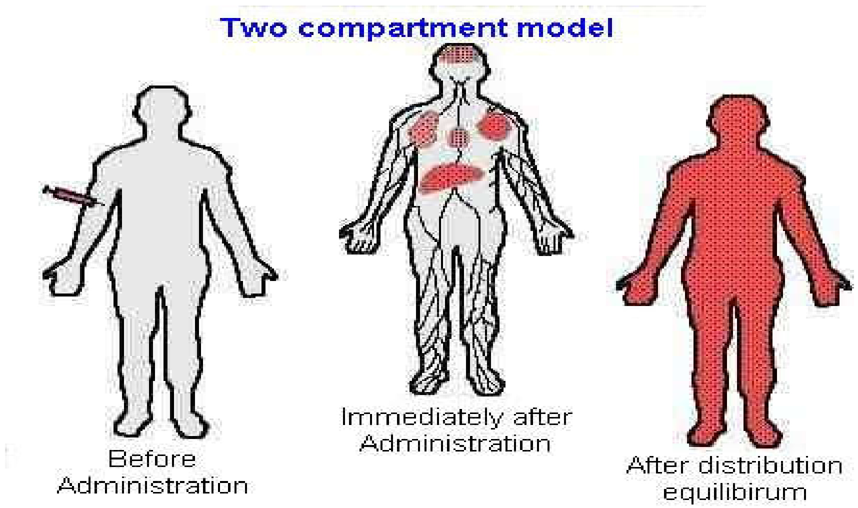


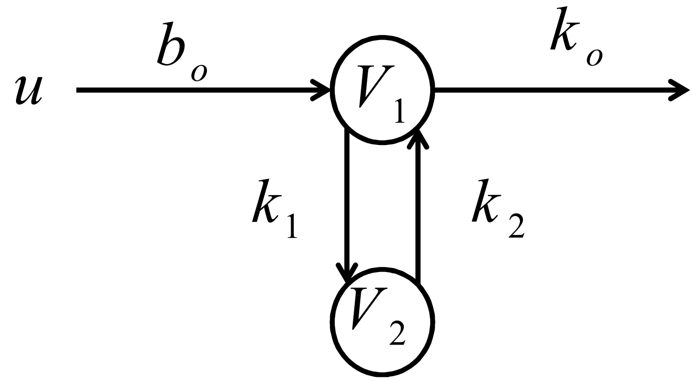
a.k.a. the step function



(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 ($mL_{solution}/s$)

The “equations of motion”…

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

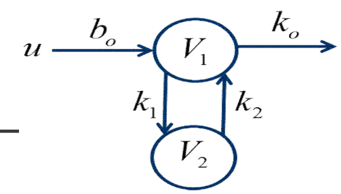
$$\Rightarrow$$

To state space…

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

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

Is the system stable?



$$\dot{c} = \mathbf{A}c + \mathbf{B}u$$

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

$$y = \mathbf{C}c + \mathbf{D}u$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} c + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

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

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

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

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…

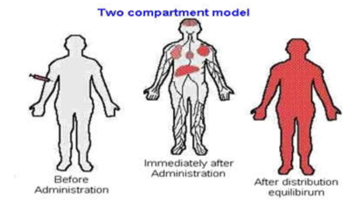
$$\begin{matrix} k_0 = 1 \\ k_1 = 1 \\ k_2 = 2 \end{matrix} \Rightarrow \lambda = \begin{bmatrix} -0.5858 \\ -3.4142 \end{bmatrix}$$

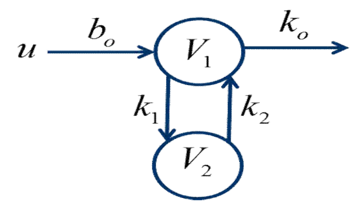
or…

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

Now to control the concentration!

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



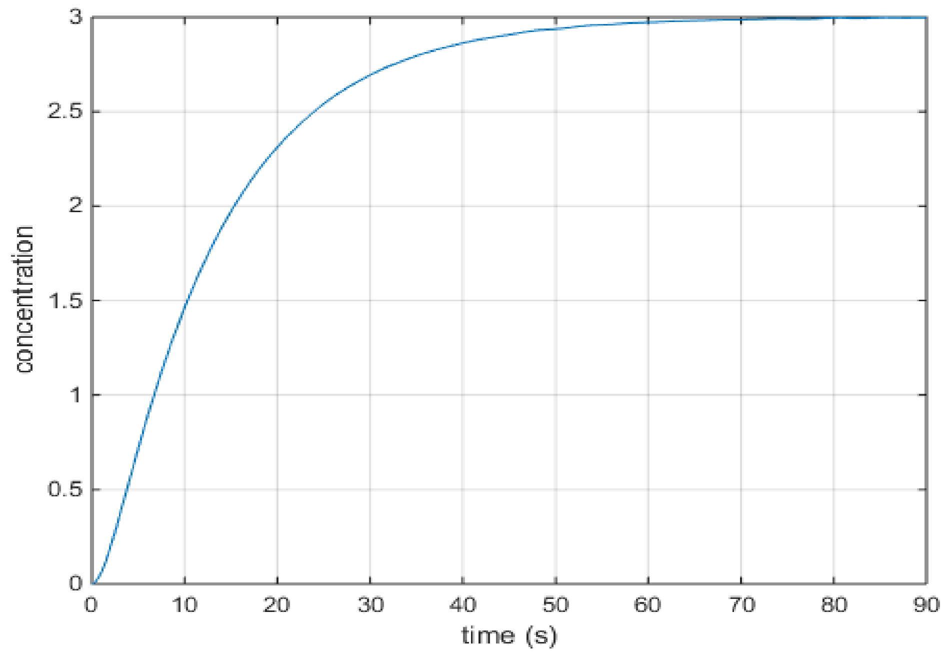


Find u such that the concentration of the drug in muscular system is 6 mL per 100mL

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

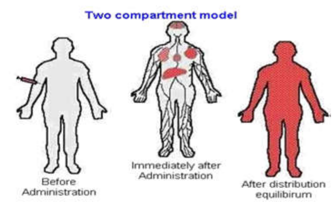
Let's start with a guess.

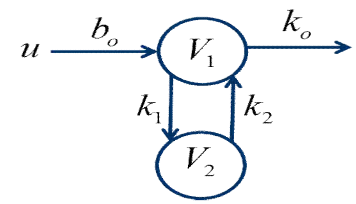
```
global u;
u = 1;

[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 = [ b0; 0];

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



2 vessel *open loop* control

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

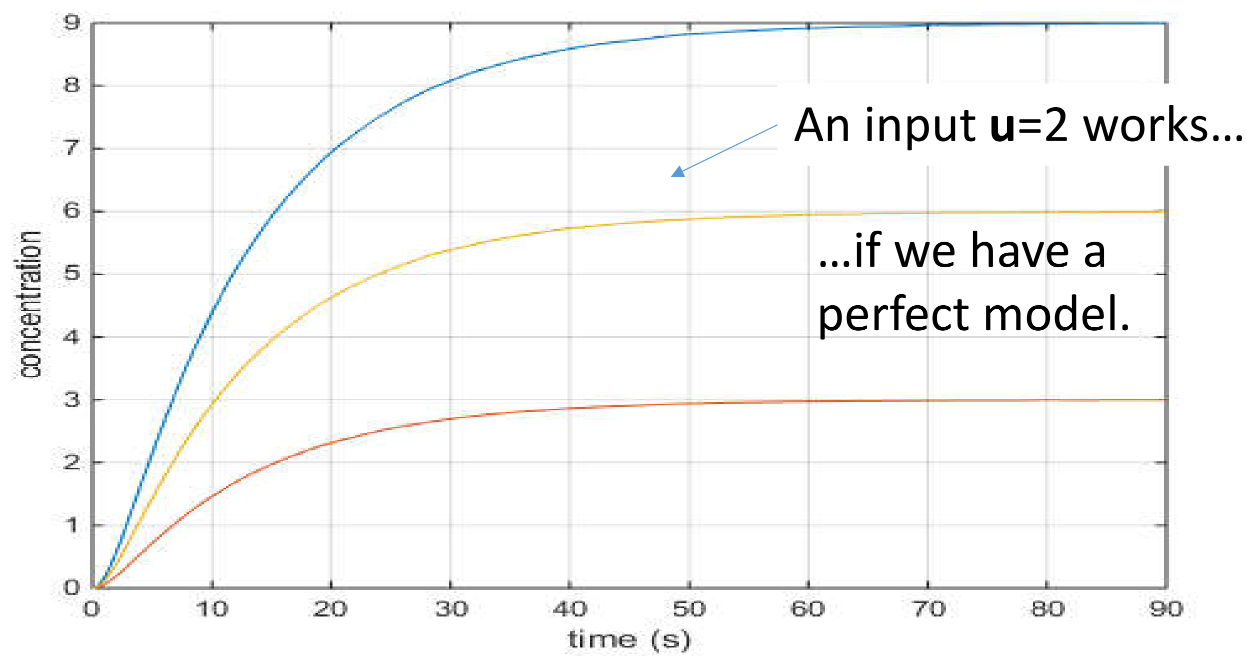




Find u such that the concentration of the drug in muscular system is 6 mL per 100mL

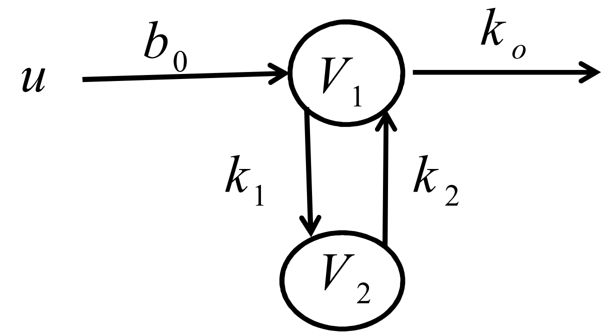
Via trial & error, we arrive at a solution….

```
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 = [ b0; 0];

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

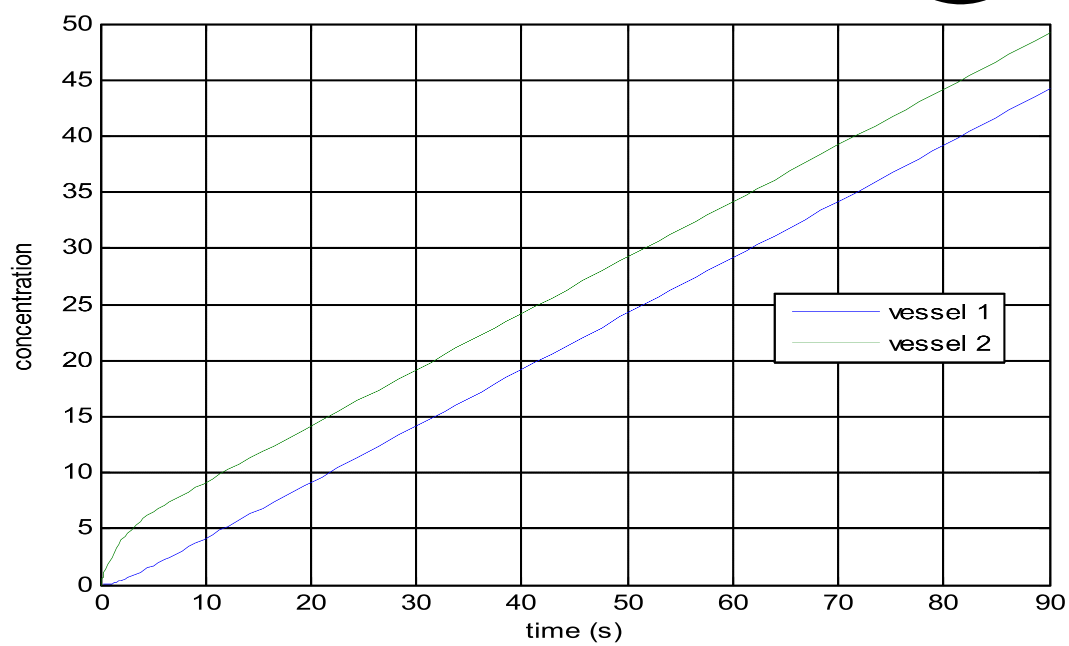


Unstable 2 Vessel Example

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




```
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 = [ b0; 0];

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



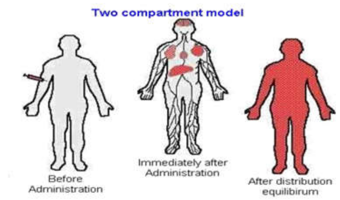
Open vs. closed loop control

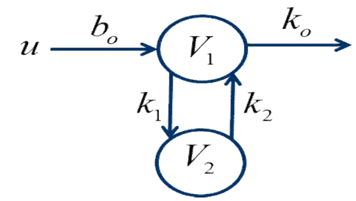
- **Open loop example results**
  - Trial & error found a *u* that gave us the desired concentration in vessel 2.
- **Open loop control**
  - No feedback. Only works if model is perfect and there are no disturbances.
  - Model is never perfect. There is almost always a disturbance.
- **Closed loop control**
  - 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:

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

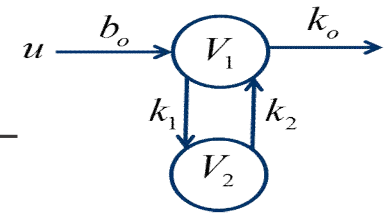
proportional control term to regulate difference between c_2 and c_d

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:

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

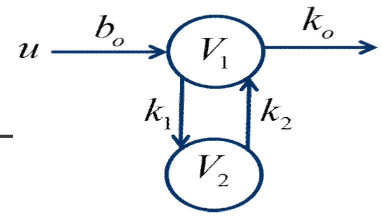
Separate the feedforward and feedback terms…

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

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

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

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…

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

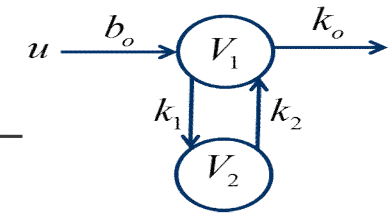
And this solution can be written more generally in the matrix form...

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

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?



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

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

Is our new "system" stable?

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

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

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

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

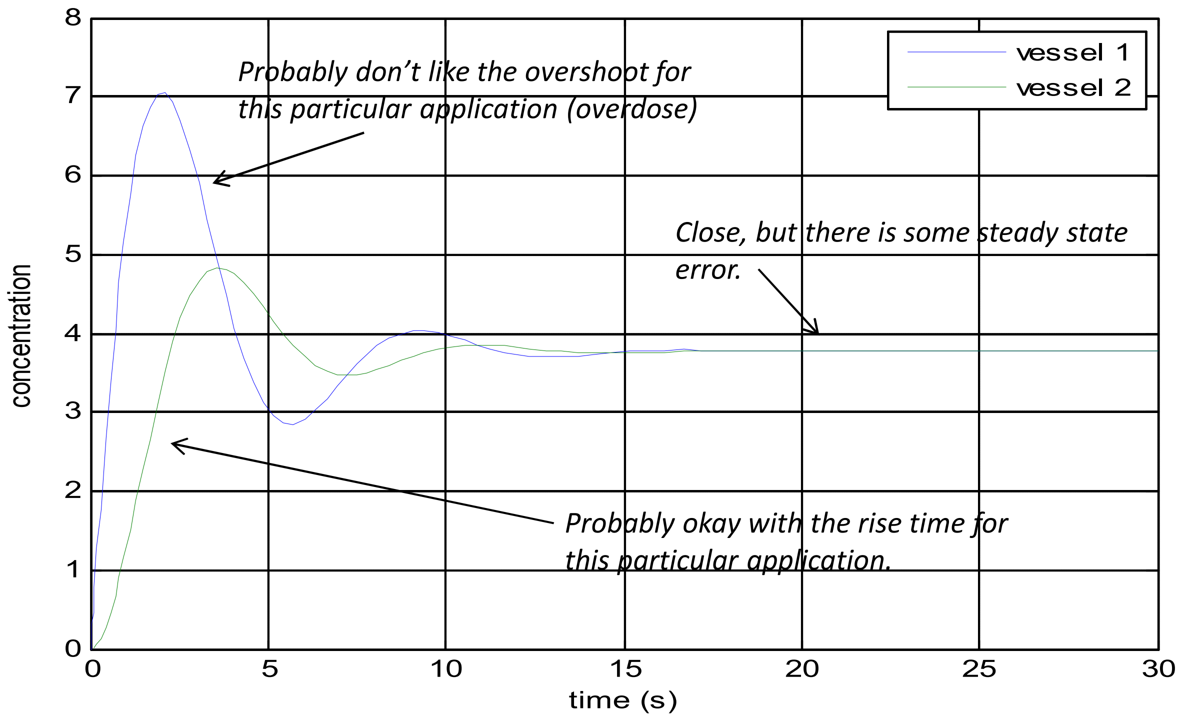
```
[t,z] = ode45('twoVolume', [0 30], [0 0]);
plot(t, z);


function cprime = twoVolume( t, c )

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

system parameters

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

controller gain
desired output
default input

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

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

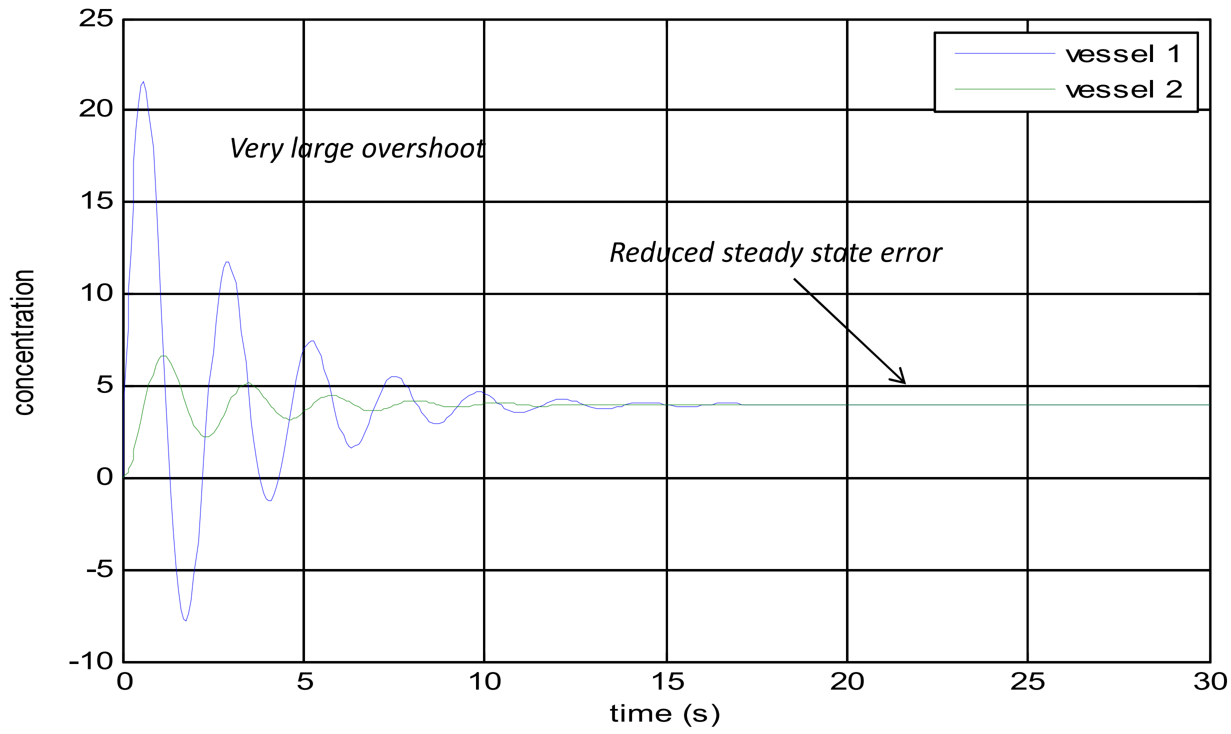


System response with controller

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

```
[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

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

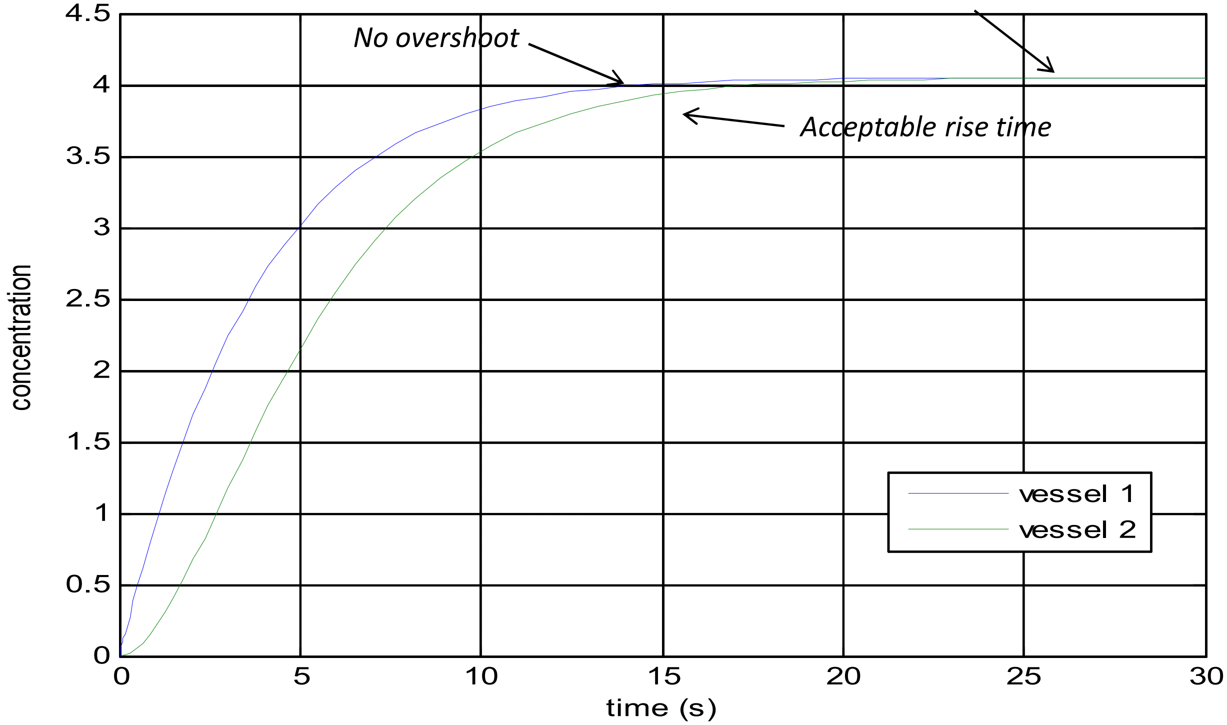
```
[t,z] = ode45('twoVolume', [0 30], [0 0]);
plot(t, z);
```

```
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

Minimal steady state error

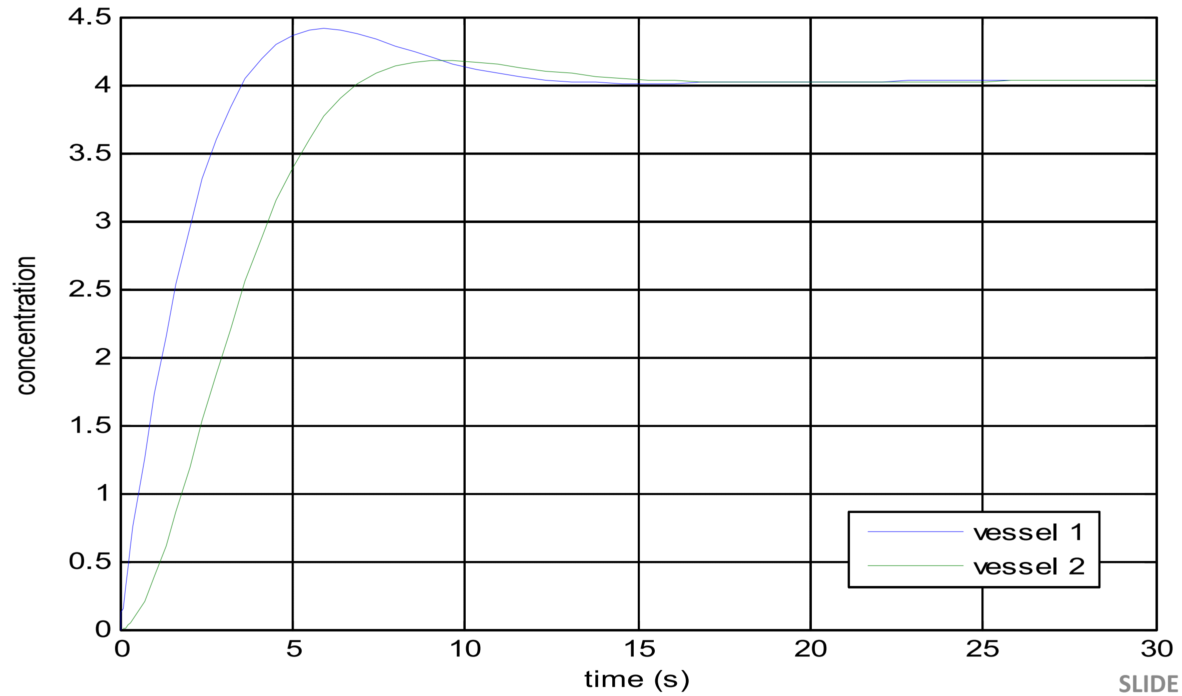


System response with controller

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

```
[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

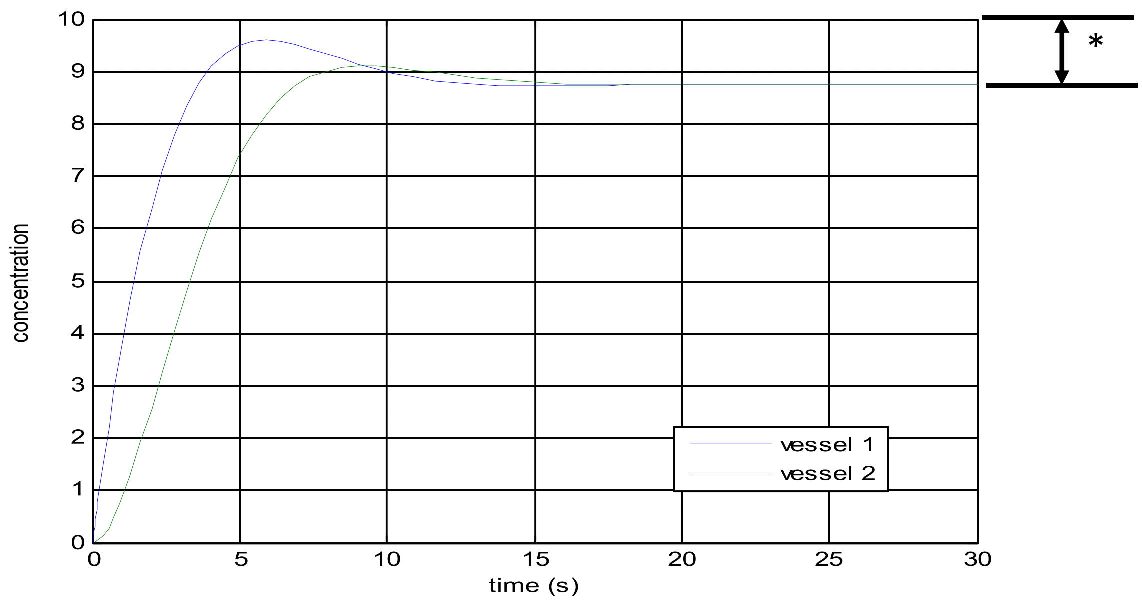


What if we change the c_d?

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

```
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.



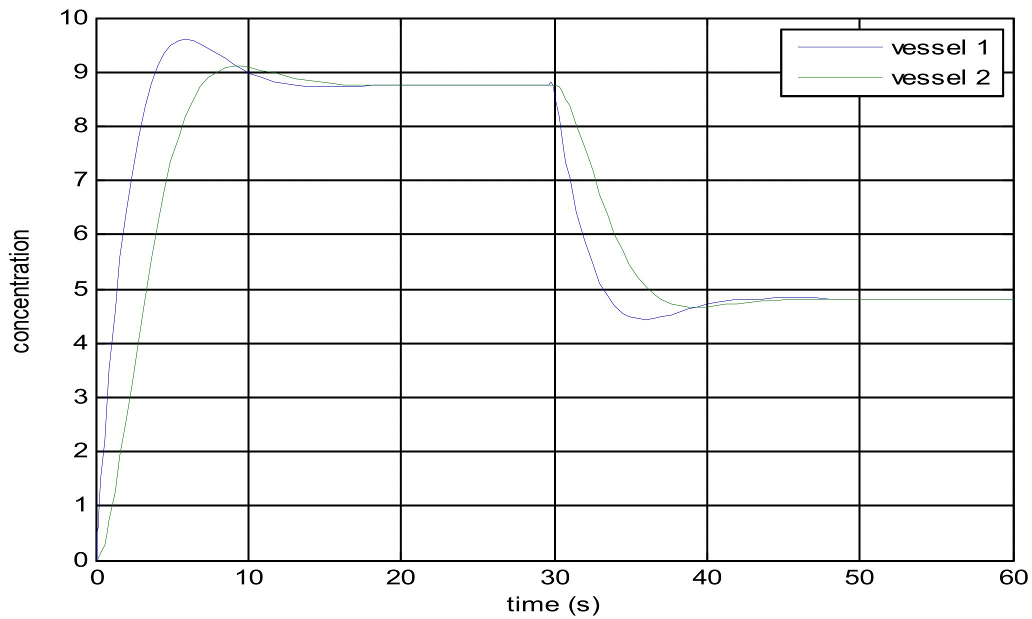
* *steady state error increased. How to eliminate this is future topic.*

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

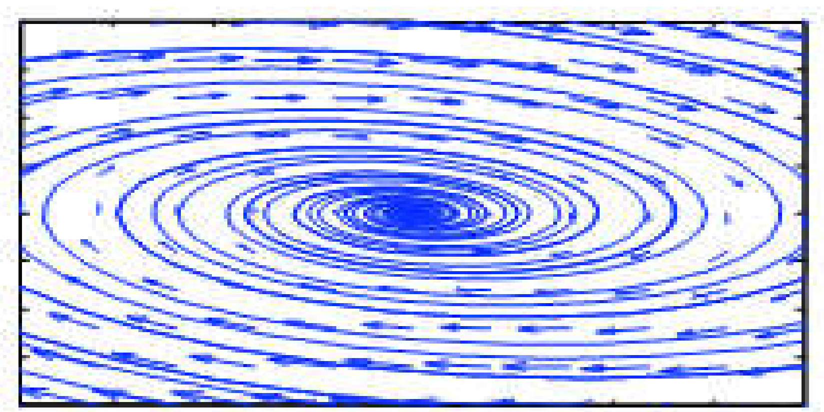


State Feedback – Defining Performance

- Stability

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

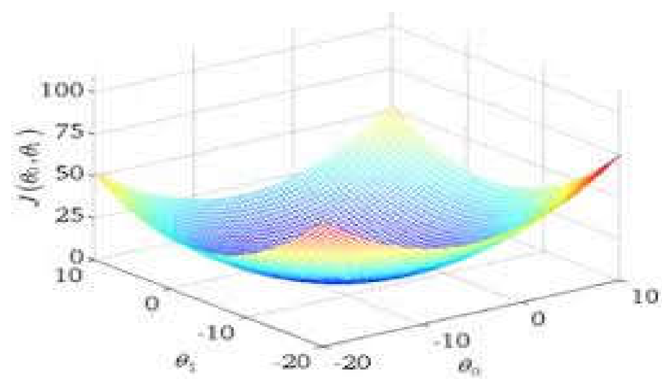
"The states of a system will approach equilibrium for the given initial states (global or local) (asymptotic or neutral)."



- 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.)"



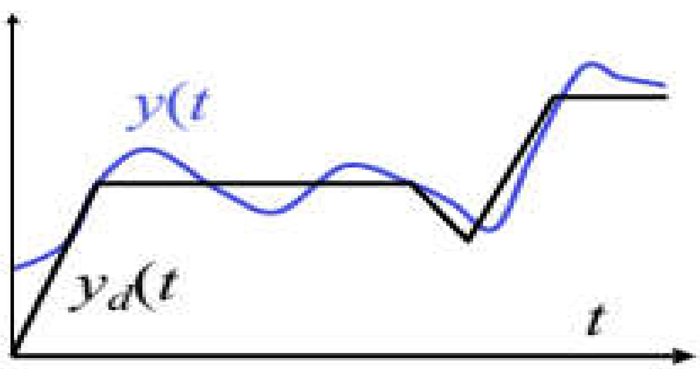
State Feedback – Key Definitions

- 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"

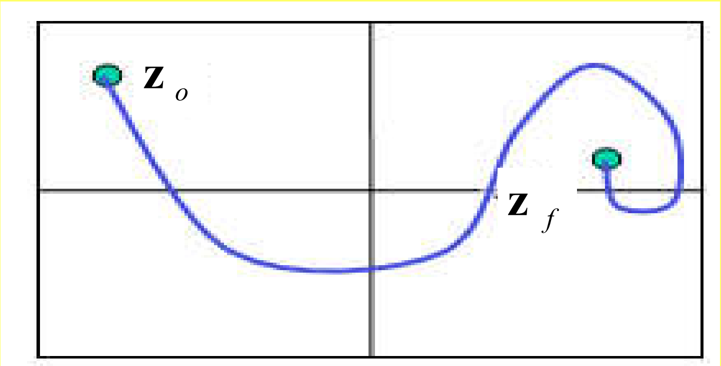


- Reachability (Controllability)

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

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

"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."



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
  - design and test state feedback controllers.

Next

Next Lesson