State-Space Model Representation

Dr. Mitch Pryor

Lesson Objective

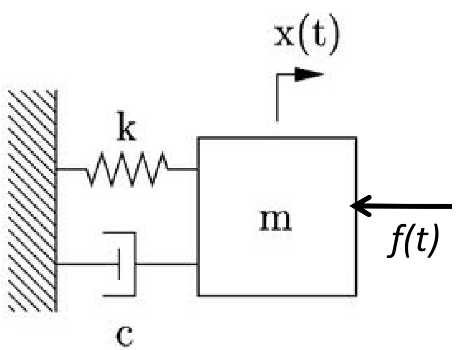
- Learn how to convert any system model (say a set of derived differential equations of motion) to state-space form.

Modeling

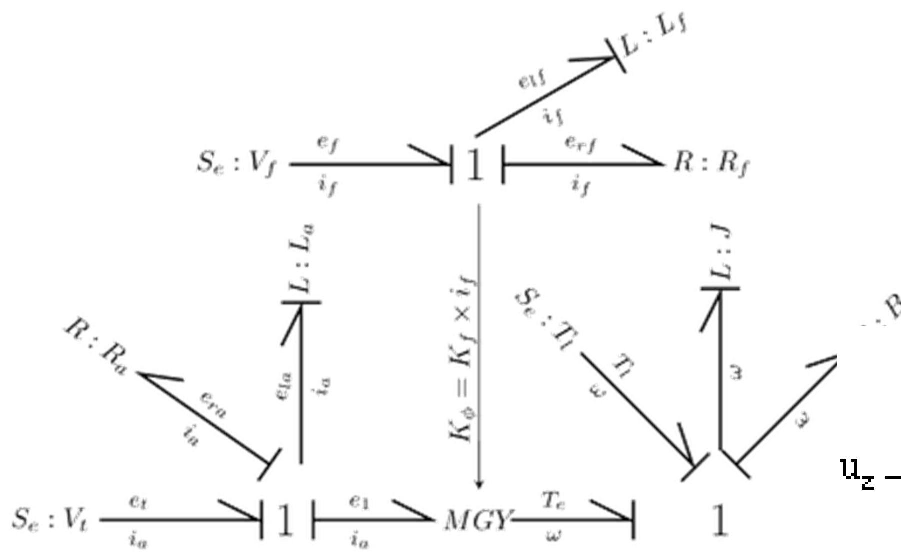
Model: A representation of something as a:
- Visualization
- Text description
- Equations
- Computer Program
- Bond Graph
- etc.

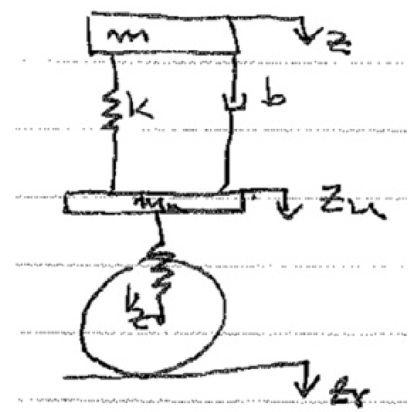
What a model is:
- A tool for analysis, comprehension, visualization
- A necessary simplification of the modeled system
- An abstraction of a real thing
- A useful component in a controller that improves the controller's performance

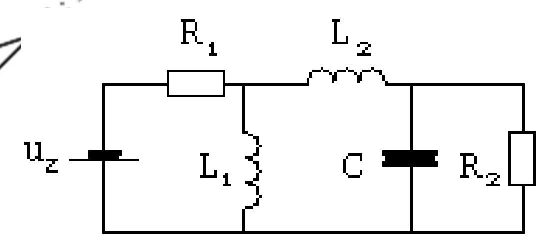
- What a model is **NOT**:
  - The real thing
  - the focus of this course



$$m\ddot{z} + b\dot{z} + kz = b\dot{z}_u + kz_u$$
$$m_u\ddot{z}_u + b\dot{z}_u + (k + k_t)z_u = k_t z_r + b\dot{z} + kz$$







Modeling terms

- **System**: a functional group of interrelated things
  – System model: a representation (often mathematical) of a system
- **State**: A changeable condition of the system regarding form, structure, location, thermodynamics, or composition of a system
  – **State Vector**: a collection of state variables that fully describes the object over time
- **Input**: an external object that acts upon a system with the possibility of changing its states
  – **Input Vector**: The set of all inputs that can impact a system
- **Output**: a dependent variable (often, but not always a state) from within the system that can be measured or quantified.
  – **Output Vector**: the set of outputs that can be measured for a system
- **Parameters:** Fixed values or properties for a given system
- **Dynamics**: a process through which the state variables change over time.

State-space model

- mathematical model of a system's inputs, outputs, and states represented as a set of 1st order ODEs.

Let,

$\mathbf{z}(\mathrm{t}) \in \mathbb{R}^n$	State vector
$\mathbf{u}(\mathrm{t}) \in \mathbb{R}^p$	Input vector
$\mathbf{y}(\mathrm{t}) \in \mathbb{R}^q$	Output (or measured) vector

In the general form, $\dfrac{d\mathbf{z}}{dt} = f(t, \mathbf{z}, \mathbf{u}) \quad \mathbf{y} = h(t, \mathbf{z}, \mathbf{u})$

If the system is linear *and* time-invariant (LTI)

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

If the system is also single-input single-output (SISO)

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

State-space model

- So for a LTI SISO system…

*n*x*n* matrix

*n*x1 state vector

*n*x1 vector

scalar input

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

How the states change due to the current values of the states and due to any inputs.

*1*x*n* row vector

*n*x1 state vector

1x1 “matrix”

scalar input

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

Provides a measured value(s) in terms of the states or inputs

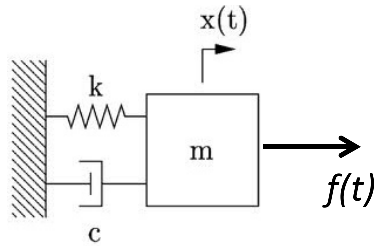
$$\mathbf{z}(\mathrm{t}) \in \mathbb{R}^n$$

$$\mathbf{u}(\mathrm{t}) \in \mathbb{R}^p$$

$$\mathbf{y}(\mathrm{t}) \in \mathbb{R}^q$$

Mass Spring Damper Example

Given: Convert the EOM (equations of motion) model for a mass-spring-damper (MSD) system to a state-space model where the position is the measured output.



$$ma = \overset{+}{\rightarrow} \sum F_x$$

$$m\ddot{x} = -c\dot{x} - kx + f(t)$$

Solve:

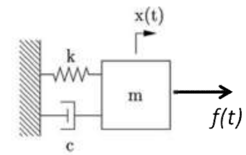
Step 1: Write the ODE(s) in the form:

$$\frac{d^n x}{dt^n} + a_1 \frac{d^{n-1}x}{dt^{n-1}} + a_2 \frac{d^{n-2}x}{dt^{n-2}} + \cdots + a_{n-1}\frac{dx}{dt} + a_n x = u$$

which in this case is…

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{f(t)}{m} = u$$

Mass Spring Damper Example



Result from step 1.

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{F(t)}{m} = u$$

<u>Step 2</u>: Define the state variables

Let, $z_1 = x$

$z_2 = \dot{x}$

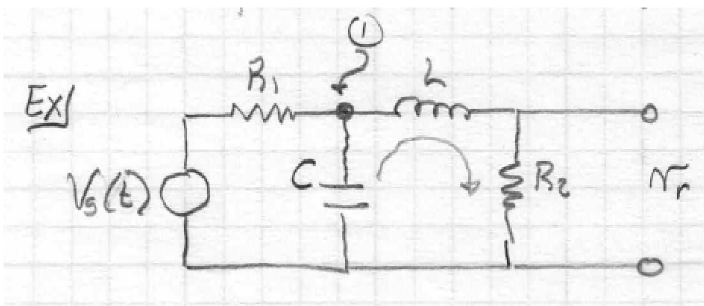
$$\dot{z}_2 = -\frac{k}{m}z_1 - \frac{c}{m}z_2 + u$$

<u>Step 3</u>: Rewrite in matrix form

$$\frac{d\mathbf{z}}{dt} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \end{bmatrix}u$$

Circuit Example

Given: Convert the EOM (equations of motion) model for an RLC circuit to a state-space model.



Input: $V_s(t)$

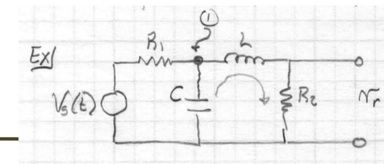
Output: v_r

Solution: Apply KCL for node 1: $$\frac{V_s - V_c}{R_1} - C\frac{dV_c}{dt} - i_L = 0$$

Apply KVL to right hand mesh: $$V_c - L\frac{di_L}{dt} - R_2 i_L = 0$$

Here the EOM's are a set of two 1st order differential equations.

Circuit Example



Let,

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} V_c \\ i_L \end{bmatrix} \qquad u = V_s(\mathrm{t})$$

Rewriting our equations…

KCL:

$$\frac{V_s - V_c}{R_1} - C\frac{dV_c}{dt} - i_L = 0$$

$$\frac{u}{R_1} - \frac{z_1}{R_1} - C\dot{z}_1 - z_2 = 0$$

$$\dot{z}_1 = -\frac{1}{CR_1}z_1 - \frac{1}{C}z_2 + \frac{1}{CR_1}u$$

KVL:

$$V_c - L\frac{di_L}{dt} - R_2 i_L = 0$$

$$z_1 - L\dot{z}_2 - R_2 z_2 = 0$$

$$\dot{z}_2 = \frac{1}{L}z_1 - \frac{R_2}{L}z_2$$

$$\frac{d\mathbf{z}}{dt} = \begin{bmatrix} -\frac{1}{CR_1} & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R_2}{L} \end{bmatrix}\mathbf{z} + \begin{bmatrix} \frac{1}{R_1 C} \\ 0 \end{bmatrix}u$$

$$y = \begin{bmatrix} 0 & R_2 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \end{bmatrix}u$$

Epidemic Disease Example

Given: Find the state-space model to simulate the spread of a disease throughout a population

Solution: In some cases, it is easier to define the states prior to determining the system model equations.

States:

z_1 = number NOT infected but susceptible to disease

z_2 = number of people infected

z_3 = number of people cured or immunized

z_4 = number of people who die

Note: Different assumptions lead to different answers. There may not be a "correct" answer when developing a model.

Inputs:

u_1 = new uninfected (but susceptible) people (born, immigrated, etc.)

u_2 = new infected people (born infected, immigrated infected, etc.)

Epidemic Disease Example

With the states defined, we can then determine the relationships between those states.

z_1 = # NOT infected

z_2 = # infected

z_3 = # immunized

z_4 = # immunized

a = healthy who die*
b = healthy who are infected
c = healthy who are immunized

d = infected who die
e = infected who are cured

f = immune who do

$$\dot{z}_4 = az_1 + dz_2 + fz_3$$
$$\dot{z}_3 = cz_1 + ez_2 - fz_3$$
$$\dot{z}_2 = bz_1 - dz_2 - ez_2 + u_2$$
$$\dot{z}_1 = -az_1 - bz_1 - cz_1 + u_1$$

$$\frac{d\mathbf{z}}{dt} = \mathbf{Az} + \mathbf{Bu}$$
$$= \begin{bmatrix} -a-b-c & 0 & 0 & 0 \\ b & -d-e & 0 & 0 \\ c & d & -f & 0 \\ a & d & f & 0 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{u}$$

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

*rated in #/100/day.

Epidemic Disease Example

From the previous slide…

$$\frac{d\mathbf{z}}{dt} = \mathbf{Az} + \mathbf{Bu}$$
$$= \begin{bmatrix} -a-b-c & 0 & 0 & 0 \\ b & -d-e & 0 & 0 \\ c & d & -f & 0 \\ a & d & f & 0 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{u}$$

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

Given the units on the coeeficients, it makes more sense to think of this as a discrete system.

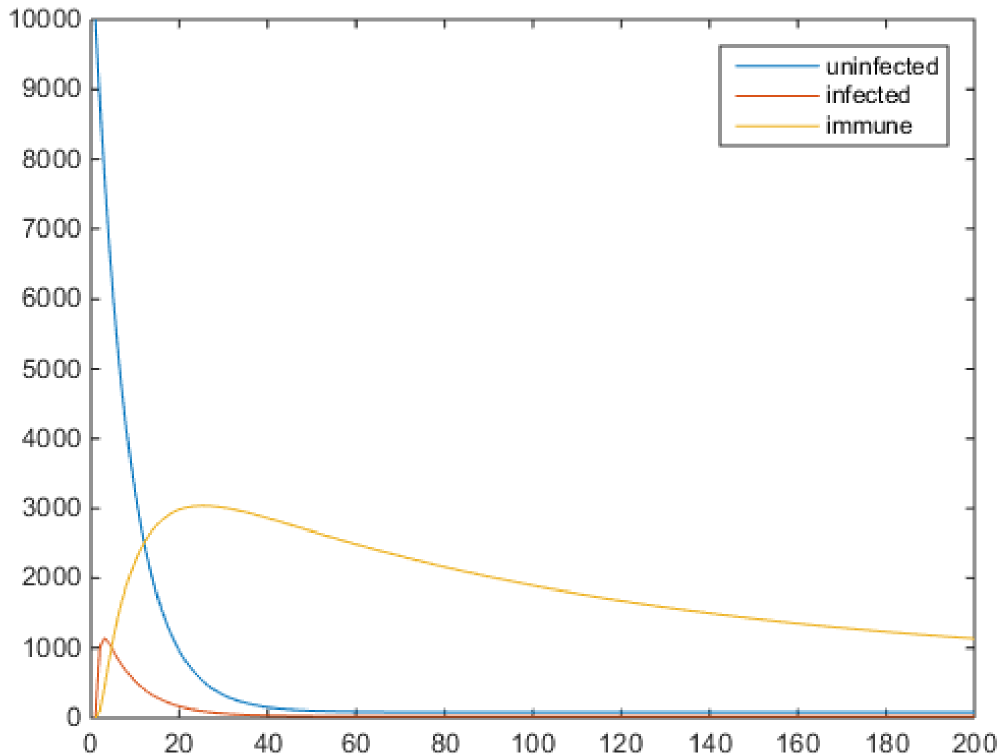
$$\mathbf{z}[i+1] - \mathbf{z}[i] = \begin{bmatrix} -a-b-c & 0 & 0 & 0 \\ b & -d-e & 0 & 0 \\ c & d & -f & 0 \\ a & d & f & 0 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{u}$$

*rated in #/100/day.

Epidemic Disease Example, MATLAB

Which makes it easy to utilize MATLAB to simulate our system

$$\mathbf{z}[i+1]=\mathbf{z}[i]+\begin{bmatrix} -a-b-c & 0 & 0 & 0 \\ b & -d-e & 0 & 0 \\ c & d & -f & 0 \\ a & d & f & 0 \end{bmatrix}\mathbf{z}+\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{u}$$

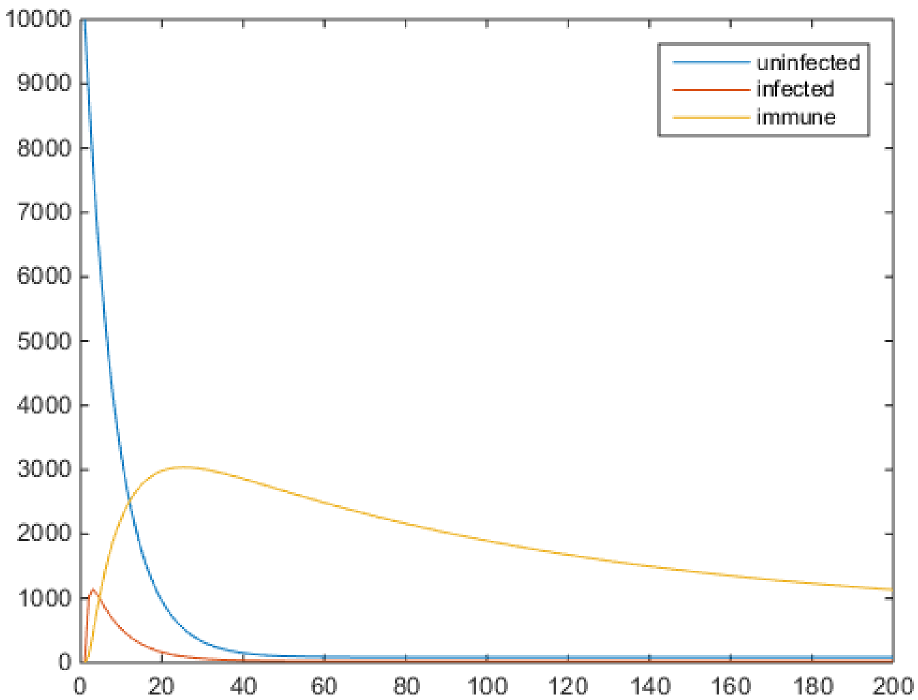



```
clear all;
z(1,1) = 10000; %initial uninfected pop
z(2,1) = 10; %initial infected pop
z(3,1) = 0; %initial immunized/cured
z(4,1) = 0; %dead

a=1; b=10; c=1; %#/100/day die, infected, immunized
d=50; e=25; %#/100/day of infected who die or are cured
f=1; %#/100/day of immune who die
u(1) = 10; u(2) = 10; %#/uninfected and infected added per day.

A = [ -a-b-c 0 0 0; b -d-e 0 0; c e -f 0; a d f 0; ]./100;
B = [ 1 0; 0 1; 0 0; 0 0 ];
C = [ eye(3) zeros(3,1) ];

day =1:200;
for c=1:length(day)-1
    z(:,c+1)= z(:,c) + A*z(:,c)+B*u';
    for(j=1:4)
        if z(j,c+1) < 0
            z(j,c+1) = 0;
        end
    end
end
plot(day,C*z)
legend('uninfected','infected','immune');
```

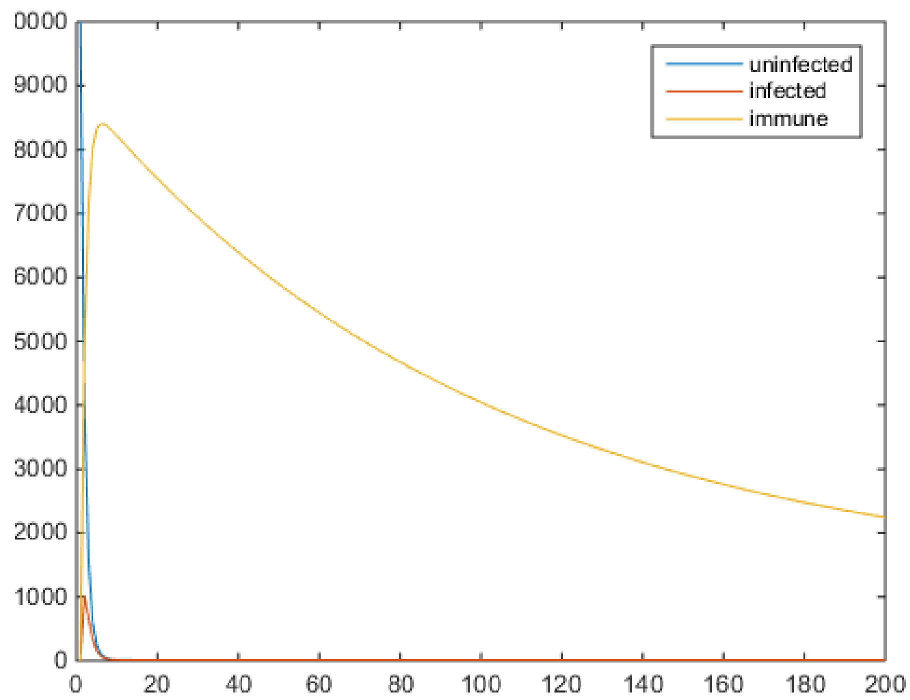
Easy to change parameters to see their impact.

Easy to change parameters

```
a=1; b=10; c=1 %#/100/day die, infected, immunized
d=50; e=25; %#/100/day of infected who die or are cured
f=1; %#/100/day of immune who die
```

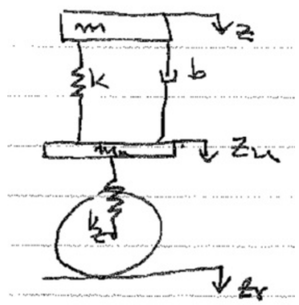



```
a=1; b=10; c=50; %#/100/day die, infected, immunized
d=50; e=25; %#/100/day of infected who die or are cured
f=1; %#/100/day of immune who die
```



Landing gear (multiple equations) example

<u>Given</u>: Convert the EOM (equations of motion) model for a plane's nose wheel to determine planes nose deflection after contact with a runway.



$$\begin{cases} m\ddot{z} + b\left(\dot{z} - \dot{z}_u\right) + k\left(z - z_u\right) = 0 \\ m_u\ddot{z}_u + b\dot{z}_u + (k + k_t)z_u = k_t z_r + b\dot{z} + kz \end{cases}$$

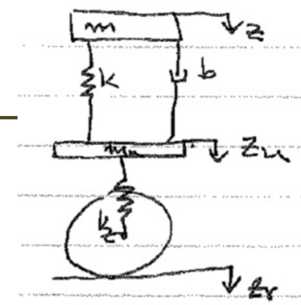
<u>Solution</u>: rewrite both equations in the correct format

$$\begin{cases} m\ddot{z} = -b\dot{z} - kz + b\dot{z}_u + kz_u \\ m_u\ddot{z}_u = -b\dot{z}_u - (k + k_t)z_u + k_t z_r + b\dot{z} + kz \end{cases}$$

Two 2nd order ODEs means states. Also, let *u* be the airfield deflection. Again, different modeling assumptions can lead to different EOMs or state-space models.

$\mathbf{z} =$ $\quad\quad\quad$ $u = z_r$

Landing gear example



Rewriting the equations as a set of first order ODE's

$$\begin{cases} m\ddot{z} = -b\dot{z} - kz + b\dot{z}_u + kz_u \\ m_u\ddot{z}_u = -b\dot{z}_u - (k+k_t)z_u + k_t z_r + b\dot{z} + kz \end{cases} \qquad \mathbf{z} = \begin{bmatrix} z & \dot{z} & z_u & \dot{z}_u \end{bmatrix}^T$$

$$\begin{cases} \dot{z}_1 = z_2 & \dot{z}_2 = -\dfrac{kz_1}{m} - \dfrac{bz_2}{m} + \dfrac{kz_3}{m} + \dfrac{bz_4}{m} \\ \dot{z}_3 = z_4 & \dot{z}_4 = \dfrac{kz_1}{m_u} + \dfrac{bz_2}{m_u} - \dfrac{(k+k_t)z_3}{m_u} - \dfrac{bz_4}{m_u} + \dfrac{k_t z_r}{m_u} \end{cases}$$

which can be put in state-space form below.

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{k}{m} & -\dfrac{b}{m} & \dfrac{k}{m} & \dfrac{b}{m} \\ 0 & 0 & 0 & 1 \\ \dfrac{k}{m_u} & \dfrac{b}{m_u} & -\dfrac{k+k_t}{m_u} & -\dfrac{b}{m_u} \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{k_t}{m_u} \end{bmatrix} z_r \qquad y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u$$

Summary

- We can put any linear model configured as a set of ordinary differential equations (ODEs) into state-space form.
- While most of the systems we will see in this class will be similar to examples given, the s-s form can be found for any set of linear ODEs. Try the following:

$$\ddot{x}_1 - 3x_2 + 2x_1 + \dot{x}_2 = 0$$
$$2\ddot{x}_2 - 3x_1 + 2\dot{x}_2 + u = 0$$

$$\ddddot{x} + 3\dddot{x} + 4x - 4u = 0$$

$$\ddddot{x} + 3\dddot{x} + 4x + y = 0$$
$$\dot{y} - 4\ddot{x} - 4u = 0$$

- Why State-space form?
  - Utilization of linear algebra for system analysis.
  - Examination of canonical systems represented in state-space form.
  - Can focus on control of systems in a particular form instead of modeling.
  - Application of numerical algorithms to solve systems in s-s form. (next!)