Systems Analysis and Control

Matthew M. Peet
Illinois Institute of Technology

Lecture 2: Systems Defined by Differential Equations

Introduction

In this Lecture, you will learn:

How to use differential equations to define a *System*.

- Identify the inputs and outputs
- Model the dynamics
  - Newton's Laws
  - Voltage Laws
- Put in First-Order (State-Space) Form

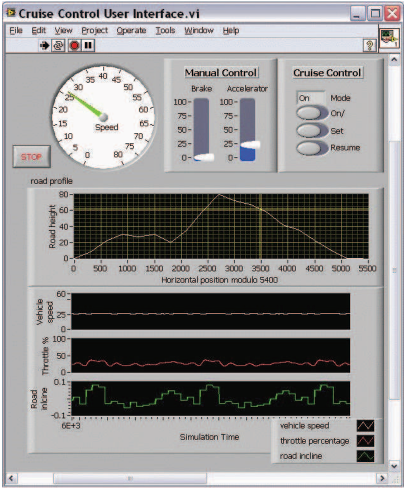
Later, we'll discuss linearization and the Laplace transform.

Plant:

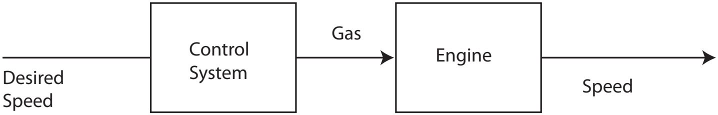
- **Input:** Throttle Position, θ_e.
- **Output:** Real Velocity, v_r.
- **Dynamics:** A simple proportional gain (no dynamics).

$$v_r = 10 \cdot \theta_e$$

The gain factor is $10mph/^\circ$



First lets start with open loop control



Actuator: Throttle

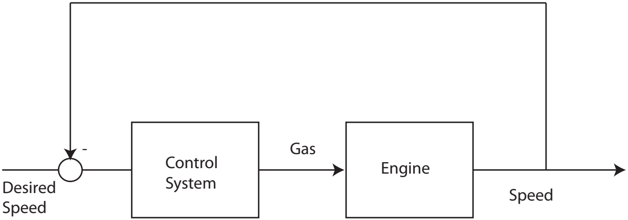
Controller:

- **Input:** Desired Velocity, v_d.
- **Output:** Throttle, θ_e.

Because the plant is simple, we will use a simple controller based on our understanding of the plant.

$$\theta_e = \frac{1}{10} v_d$$

Now lets try using closed loop control



Actuator: Throttle

Sensor: Real Velocity

Controller:

- **Input:** Error in Velocity, $e_v = v_r - v_d$.
- **Output:** Throttle, θ_e.

Our controller is static and uses no knowledge of the plant. It simply amplifies the error signal by a factor k. Any positive value of k will work.

$$\theta_e = -k \cdot e_v = -k \cdot (v_r - v_d)$$

Closed Loop vs. Open Loop

Open Loop: Two relations:

$$v_r = 10 \cdot \theta_e \qquad \text{and} \qquad \theta_e = \frac{1}{10} v_d$$

we have

$$v_r = 10 \frac{1}{10} v_d = v_d$$

So there is no error in the open-loop control

Closed Loop: We also have two relations:

$$v_r = 10 \cdot \theta_e \qquad \text{and} \qquad \theta_e = -k\left(v_r - v_d\right)$$

Combining these, we get $\mathbf{v_r} = -10 \cdot k(\mathbf{v_r} - v_d)$.
Solving for velocity, v_r, we get for $k = 10$,

$$v_r = \frac{10 \cdot k}{1 + 10 \cdot k} v_d = \frac{100}{101} v_d = .99 v_d.$$

Impact of Error and Disturbances

Comparison:

- Open Loop: No final error
- Closed Loop: Small final error
  - Error can be made arbitrarily by letting $k \to \infty$, which makes

$$v_r = \frac{10 \cdot k}{1 + 10 \cdot k} v_d \to v_d.$$

  - Error can be eliminated entirely using a dynamic controller.

Question: What happens when things aren't perfect?
Problems:

- Modeling Error: Suppose our model is off by 10%, so that

$$v_r = 11 \cdot \theta_e$$

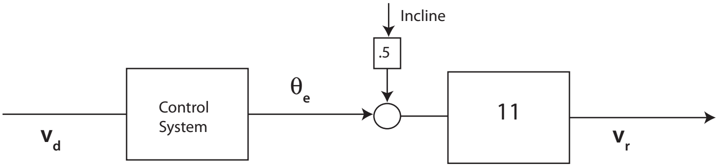
- Disturbance: An Incline, i_d will cause a decrease in throttle power of $.5/^\circ$.

$$\Delta\theta_e = -.5 \cdot i_d$$

Impact of Error and Disturbances

Open Loop

Let $v_d = 50mph$, $i_d = -1°$.
Recalculate for the open loop case:



$$v_r = 11(\theta_e - .5 \cdot i_d)$$
$$\theta_e = \frac{1}{10} v_d = 5$$

we have

$$v_r = 11(5 + .5) = 60.5mph$$

Which is **NOT ACCEPTABLE!!!**.

Impact of Error and Disturbances

Closed Loop

Recalculate for the closed loop case:

- **Real Plant with Disturbance:** $v_r = 11 \cdot (\theta_e - .5 \cdot i_d)$
- **Controller:** $\theta_e = -k\,(v_r - v_d) = -k(v_r - 50)$

Combine expressions and solve for v_r!!!

$$v_r = 11(-kv_r + 50k + .5) = -11kv_r + 11 \cdot 50 \cdot k + 5.5$$

Solving for v_r yields

$$v_r = \frac{11k + .11}{1 + 11k} 50 = \frac{110.11}{111} 50 = .991 * 50 = 49.6mph$$

Better than without disturbance!!!!

Note: Solving for v_r is called *Closing the Loop*. We will be doing this a lot in the section on block diagrams.

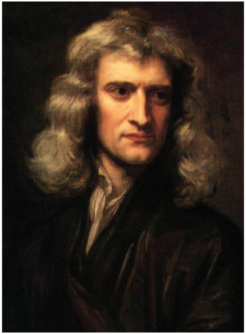
A Brief Review of Modeling

The previous model of an engine was a *static model*.
In this class, all models will be either

- static.
- differential equations.

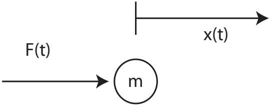
The modeling of physical systems using differential equations was introduced by Newton in 1684.

- I expect you to know how to derive Differential Equation models.
- Our treatment will be brief.



The first differential equation model was for a point mass.
Newton's Second Law:

$$\frac{d^2}{dt^2}x(t) = F/m$$



Review: Modeling

Differential Equations

The motion of dynamical systems can usually be specified using ordinary differential equations. e.g.

$$\frac{dx}{dt}(t) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t))$$

Where

- This is a first-order differential equation
- $u(t)$ is the input
- $y(t)$ is the output
- x is a *state variable*.
  - position, heading, velocity, etc.
- f, g are possibly nonlinear functions.

Note: Often, the equation is higher order.

Review: Equations of Motion

Linear Equations

Usually, our equations of motion will be linear. e.g.

$$\dot{x} = ax(t)$$

where

- a is a constant scalar.
- in this case $f(x) = ax$.

Linear equations are preferable because

- The motion of linear systems is much easier to visualize.
- Stability of linear systems is easy to determine
  - $\dot{x} = ax$ is stable if $a < 0$ and unstable if $a \geq 0$.

Review: Equations of Motion

Higher Orders or Multiple Variables

Most often, the dynamics will either

Be coupled with another variable:

$$\dot{x} = ax + bz$$
$$\dot{z} = cx + dz$$

where

- The motion of x affects the motion of y and vice-versa.

Be higher order:

$$\ddot{x} = a\dot{x} + bx$$

where

- Commonly obtained from Newton's Second law.

$$F = ma$$

or, in other words

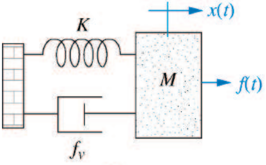
$$\ddot{x} = F/m.$$

Dynamic Model: Suspension System

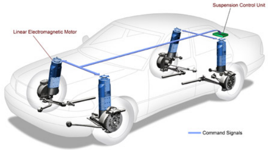
Mass-Spring Model

We wish to study the motion of the vehicle subject to disturbances.

- Model the car as a solid mass
- Control the vertical motion of the car $(x(t))$



(a)



Inputs: Force, $f(t)$.
Outputs: Displacement, $y(t) = x(t)$.

Definition 1.

A system with one input and one output is single-input, single-output (**SISO**). A system with more than one input *or* more than one output is multi-input multi-output (**MIMO**)

Dynamic Model: Suspension System

Mass-Spring Model

Plant Dynamics: Equations of Motion

- Spring Force: Opposes motion in x with spring constant K.

$$F_s(t) = -Kx(t)$$

- Damper Force: Opposes motion in $\dot{x}$ with damping coefficient f_v

$$F_d(t) = -f_v\dot{x}(t)$$

- Newton's Second Law:

$$m\ddot{x}(t) = F_s(t) + F_d(t) + f(t)$$

System Model:

$$\ddot{x}(t) = -\frac{K}{m}x(t) - \frac{f_v}{m}\dot{x}(t) + \frac{1}{m}f(t)$$
$$y(t) = x(t)$$

Standard Forms

Frequency Domain

Once we have our dynamic model

$$\ddot{x}(t) = -\frac{K}{m}x(t) - \frac{f_v}{m}\dot{x}(t) + \frac{1}{m}f(t) \qquad \text{Differential Equations}$$

$$y(t) = x(t) \qquad \text{Output Equation}$$

This model can be expressed in two standard forms

- Transfer Function
- State-Space

We will discuss these in more depth soon. For now:

Transfer Function: Apply the Laplace Transform to both equations and solve for the output.

$$s^2x(s) = -\frac{K}{m}x(s) - \frac{f_v}{m}sx(s) + \frac{1}{m}f(s) \qquad \text{Differential Equations}$$

$$y(s) = x(s) \qquad \text{Output Equation}$$

which yields

$$y(s) = \frac{1}{ms^2 + f_v s + K}u(s)$$

Suspension System with Wheel Dynamics

More Detailed Model

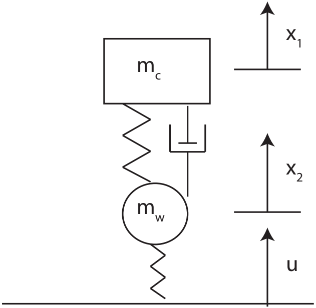
Now, we add the dynamics of the wheel.

There are two outputs:
Outputs:

- Vehicle Position, x_1
- Wheel Position, x_2

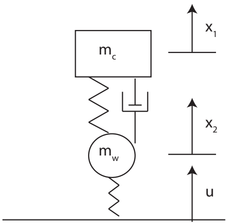
Our input is the position of the surface of the road.
Inputs:

- Road Surface, u



Suspension Model

This time we write the dynamics of both the wheel and the car.

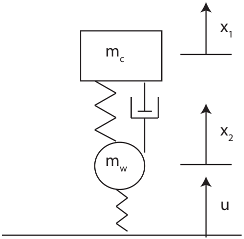


Car Dynamics: Equations of Motion

- Spring 1 Force on Car: $F_{s1,c}(t) = -K_1(x_1(t) - x_2(t))$
- Damper Force on Car: $F_{d,c}(t) = -c(\dot{x}_1(t) - \dot{x}_2(t))$
- Newton's Second Law:

$$\begin{aligned} m_c\ddot{x}_1(t) &= F_{s1,c}(t) + F_{d,c}(t) \\ &= -K_1(x_1(t) - x_2(t)) - c(\dot{x}_1(t) - \dot{x}_2(t)) \end{aligned}$$

Suspension Model



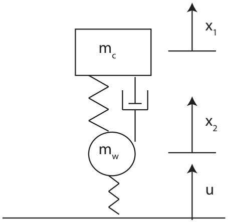
Wheel Dynamics: Equations of Motion

- Spring 1 Force on Wheel: $F_{s1,w}(t) = K_1(x_1(t) - x_2(t))$
- Spring 2 Force on Wheel: $F_{s2,w}(t) = -K_2(x_2(t) - u(t))$
- Damper Force on Wheel: $F_{d,w}(t) = c(\dot{x}_1(t) - \dot{x}_2(t))$
- Newton's Second Law:

$$\begin{aligned} m_w \ddot{x}_2(t) &= F_{s1,w}(t) + F_{s2,w}(t) + F_{d,w}(t) \\ &= K_1(x_1(t) - x_2(t) - K_2(x_2(t) - u(t))) + c(\dot{x}_1(t) - \dot{x}_2(t)) \end{aligned}$$

Equations of Motion

Combining the dynamics, we get the coupled system dynamics.



$$
\begin{aligned}
m_w \ddot{x}_2(t) &= K_1(x_1(t) - x_2(t)) - K_2(x_2(t) - u(t)) + c(\dot{x}_1(t) - \dot{x}_2(t)) \\
m_c \ddot{x}_1(t) &= -K_1(x_1(t) - x_2(t)) - c(\dot{x}_1(t) - \dot{x}_2(t)) \\
y(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}
\end{aligned}
$$

This is quite complicated.

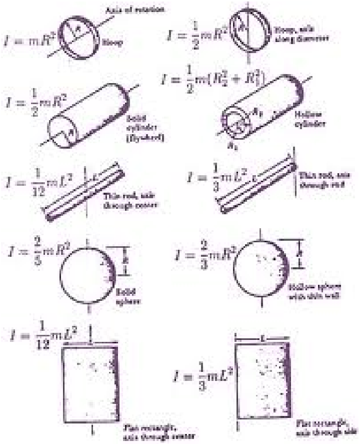
- To simplify, we would like to use a *Standard Form*.

Newton's Second Law Applied to Rigid Bodies

The rate of change of angular momentum is given by

$$\sum M_i = I\alpha = I\ddot{\theta}$$

- $\alpha = \ddot{\theta}$ is the angular acceleration in inertial coordinates.
- I is the moment of inertia, which varies by object.
- M_i are the moments applied to the body.

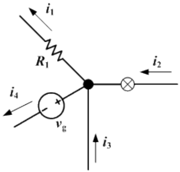


Other Sources of Models

Voltage Laws

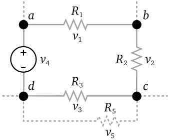
Kirchhoff's Current Law (KCL):
Current is conserved at each junction

$$\sum i_k = 0$$



Kirchhoff's Voltage Law (KVL): Net Voltage change around any loop is zero.

$$\sum_k V_k = 0$$



These are combined with standard voltage laws such as voltage drop across a resister, inductor and capacitor:

$$V_r(t) = R i_r(t) \qquad \frac{d}{dt} i_L(t) = \frac{1}{L} V_L(t) \qquad \frac{d}{dt} V_c(t) = \frac{1}{C} i_c(t)$$

State-Space is a way of writing first order differential equation using matrices. We write

$$\dot{\vec{x}} = A\vec{x}$$

where $\vec{x}$ is a vector and $A \in \mathbb{R}^{n\times n}$ is a square matrix.

Example:

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix} = \begin{bmatrix}-1 & 0 & 1\\ 2 & 0 & 0\\ 0 & -1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}$$

Is equivalent to writing the three differential equations

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_3\\ \dot{x}_2 &= 2x_1\\ \dot{x}_3 &= -x_2 + x_3\end{aligned}$$

Writing equations in state-space has many advantages

Review: Equations of Motion

Multiple Variables and State-Space

Consider the system

$$\dot{x} = ax + by$$
$$\dot{y} = cx + dy$$

When we have multiple coupled equations, the best option is: **Convert to State-Space:**

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

Which is easily expressed as

$$\dot{\mathbf{x}} = A\mathbf{x}$$

where

- $\mathbf{x}$ is a vector.
- A is a matrix.

The equation describes the motion of the vector.

Standard Forms: State-Space Form

Definition 2.

State-Space Form is a convenient way of representing multivariate or *linear* MIMO systems using 4 matrices.

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

- u is the vector of **Inputs**.
- y is the vector of **Outputs**.
- x is the **State**.

$u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, and $x \in \mathbb{R}^n$ can be vectors of any dimension. However, the matrices must be the right size:

$$A \in \mathbb{R}^{n\times n} \qquad B \in \mathbb{R}^{n\times m}$$
$$C \in \mathbb{R}^{p\times n} \qquad D \in \mathbb{R}^{p\times m}$$

- $u \in \mathbb{R}^n$ means u is a real vector of length n.
- $C \in \mathbb{R}^{p\times n}$ means C is a matrix with p rows and n columns.

Review: Equations of Motion

Reducing Higher Order Dynamics

When we have higher order dynamics,

$$\dddot{x}(t) = a\dot{x}(t) + bx(t) + u(t)$$
$$y(t) = x(t) + u(t)$$

we can still use state-space form by

- Introducing new variables.

Procedure:

- Define a new variable for every Higher Order Term (HOT) except for the the highest.
  - e.g. Let $x_1 = x$, $x_2 = \dot{x}$ and $x_3 = \ddot{x}$.
- Add a new first order differential equation for each new variable.
  - e.g. $\dot{x}_1 = x_2$ and $\dot{x}_2 = x_3$
- Then put in state-space form.

Finally we have for our example

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = x_3(t)$$
$$\dot{x}_3(t) = ax_2(t) + bx_1(t) + u(t)$$

Using our first-order equations:

$$\dot{x}_1(t) = x_2(t); \qquad \dot{x}_2(t) = x_3(t)$$

$$\dot{x}_3(t) = ax_2(t) + bx_1(t) + u(t) \qquad y(t) = x_1(t) + u(t)$$

We construct the matrix representation:

$$\dot{x}(t) = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ b & a & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}(t) + \begin{bmatrix} 1 \end{bmatrix} u(t)$$

So that

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ b & a & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 1 \end{bmatrix}$$

Constructing State-Space Systems: Suspension System

Recall the dynamics:

$$m_w\ddot{x}_2(t) = K_1(x_1(t) - x_2(t)) - K_2(x_2(t) - u(t)) + c(\dot{x}_1(t) - \dot{x}_2(t))$$
$$m_c\ddot{x}_1(t) = -K_1(x_1(t) - x_2(t)) - c(\dot{x}_1(t) - \dot{x}_2(t))$$
$$y(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Define the new variables z_i

$$z_1 = x_1 \qquad z_2 = \dot{x}_1 \qquad z_3 = x_2 \qquad z_4 = \dot{x}_2$$

Which yields the following set of equations: $y(t) = \begin{bmatrix} z_1(t) \\ z_3(t) \end{bmatrix}$,

$$\dot{z}_1(t) = z_2(t)$$
$$\dot{z}_2(t) = -\frac{K_1}{m_c}(z_1(t) - z_3(t)) - \frac{c}{m_c}(z_2(t) - z_4(t))$$
$$\dot{z}_3(t) = z_4(t)$$
$$\dot{z}_4(t) = \frac{K_1}{m_w}(z_1(t) - z_3(t)) - \frac{K_2}{m_w}(z_3(t) - u(t))) + \frac{c}{m_w}(z_2(t) - z_4(t))$$

Constructing State-Space Systems

$$\dot{z}_1(t) = z_2(t)$$
$$\dot{z}_2(t) = -\frac{K_1}{m_c} z_1(t) - \frac{c}{m_c} z_2(t) + \frac{K_1}{m_c} z_3(t) + \frac{c}{m_c} z_4(t)$$
$$\dot{z}_3(t) = z_4(t)$$
$$\dot{z}_4(t) = \frac{K_1}{m_w} z_1(t) + \frac{c}{m_w} z_2(t) - \left(\frac{K_1}{m_w} + \frac{K_2}{m_w}\right) z_3(t) - \frac{c}{m_w} z_4(t) - \frac{K_2}{m_w} u(t)$$
$$y(t) = \begin{bmatrix} z_1(t) \\ z_3(t) \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_1}{m_c} & -\frac{c}{m_c} & \frac{K_1}{m_c} & \frac{c}{m_c} \\ 0 & 0 & 0 & 1 \\ \frac{K_1}{m_w} & \frac{c}{m_w} & -\left(\frac{K_1}{m_w} + \frac{K_2}{m_w}\right) & -\frac{c}{m_w} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{K_2}{m_w} \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}(t) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u(t)$$

Summary

What have we learned today?

A Static Model of Cruise-Control

- Simple static model and Control
- Open Loop Control
- Closed Loop Control
- Benefits of Feedback

Dynamic Models

- Including Inputs and Outputs
- Using Newton's Laws
- MIMO and SISO systems
- Other sources of models (Kirchhoff's Laws)

State-Space

- State-Space Form