Lecture 2: Systems Defined by Differential Equations
Introduction

In this Lecture, you will learn:

1. Quantitative illustration of the benefits of Feedback
2. Some basics of modeling using differential equations.
3. The State-Space Framework.
CRUISE CONTROL

**Plant:** The Automobile (Car)

- **Input:** Throttle Position, $\theta_{\text{throttle}}$.
- **Output:** Real Velocity, $v_{\text{true}}$.
- **Dynamics:** No Dynamics! Speed is proportional to throttle (proportional gain).

\[
\frac{v_{\text{true}}}{\text{output}} = 10 \cdot \frac{\theta_{\text{throttle}}}{\text{Plant}} \cdot \frac{\text{input}}{
}
\]

The *gain factor* is 10 mph/°.
First let's start with open loop control.

**Actuator:** Throttle

**Controller:**
- **Input:** Desired Velocity, \( v_{\text{desired}} \).
- **Output:** Throttle, \( \theta_{\text{throttle}} \).

**Open Loop Controller** We use a simple controller based on our knowledge of the plant.

\[
\theta_{\text{throttle}} = \frac{1}{10} v_{\text{desired}}
\]
Cruise Control: Open-Loop vs. Closed-Loop

Closed Loop Control: The Error signal

Now let's try using closed loop control

Actuator: Throttle

Sensor: Speedometer

Look at the CONTROLLER:

- **Input:** Error in Velocity, $e_{velocity} = v_{true} - v_{desired}$.
- **Output:** Throttle, $\theta_{throttle}$.

Proportional Feedback: Amplify the error signal by a scalar gain $k$.

$$\theta_{throttle}^\text{output} = -k \cdot e_{velocity} = -k \cdot (v_{true} - v_{desired})$$
Closed Loop vs. Open Loop (Solving for $v_{\text{true}}$)

**Open Loop**: Two equations:

\[ v_{\text{true}} = 10 \cdot \theta_{\text{throttle}} \quad \text{and} \quad \theta_{\text{throttle}} = \frac{1}{10} v_{\text{desired}} \]

Combining, we get

\[ v_{\text{true}} = 10 \cdot \frac{1}{10} v_{\text{desired}} = v_{\text{desired}}. \]

Open-loop control has **No Error!**

**Closed Loop**: Two equations:

\[ v_{\text{true}} = 10 \cdot \theta_{\text{throttle}} \quad \text{and} \quad \theta_{\text{throttle}} = -k \left( v_{\text{true}} - v_{\text{desired}} \right). \]

Combining these, we get $v_{\text{true}} = -10 \cdot k (v_{\text{true}} - v_{\text{desired}})$.

Lets **Choose** $k = 10$.

\[ v_{\text{true}} = \frac{10 \cdot k}{1 + 10 \cdot k} v_{\text{desired}} = \frac{100}{101} v_{\text{desired}} = 0.99 v_{\text{desired}}. \]

Closed-loop control has **1% Error**.
Impact of Error and Disturbances

Comparison:
- Open Loop: No error
- Closed Loop: Small error
  - Error gets very small if $k \to \infty$, since
  \[
  v_{\text{true}} = \frac{10 \cdot k}{1 + 10 \cdot k} v_{\text{desired}} \to v_{\text{desired}}.
  \]

Question: Why use feedback?
- Answer: Life is Messy.

Problems:
- Modeling Errors: What if our model is wrong by 10%, so
  \[
  v_{\text{true}} = 11 \cdot \theta_{\text{throttle}}
  \]
- Disturbances: An Incline of $i_{\text{disturbance}}$ degrees will reduce the throttle by 
  $0.5/\degree$.
  \[
  \Delta \theta_{\text{throttle}} = -0.5 \cdot i_{\text{disturbance}}
  \]
Let $v_{\text{desired}} = 50\, \text{mph}$, $i_{\text{disturbance}} = -1^\circ$.

Now recalculate the \textit{Open Loop} output: $V_{\text{true}}$

\[
v_{\text{true}} = 11 \left( \theta_{\text{throttle}} - 0.5 \cdot i_{\text{disturbance}} \right)
\]

\[
\theta_{\text{throttle}} = \frac{1}{10} v_{\text{desired}} = 5
\]

we have

\[
v_{\text{true}} = 11(5 + 0.5) = 60.5\, \text{mph}
\]

Which is \textbf{NOT ACCEPTABLE}!!!
Recalculate the \textit{Closed Loop} output: $V_{\text{true}}$

- **Plant with Disturbance:** $v_{\text{true}} = 11 \cdot (\theta_{\text{throttle}} - 0.5 \cdot i_{\text{disturbance}})$
- **Controller:** $\theta_{\text{throttle}} = -k \left( v_{\text{true}} - v_{\text{desired}} \right) = -k(v_{\text{true}} - 50)$

Combine these equations and solve for $v_{\text{true}}$.

$$v_{\text{true}} = 11(-k \cdot v_{\text{true}} + 50 \cdot k + 0.5) = -11 \cdot k \cdot v_{\text{true}} + 11 \cdot 50 \cdot k + 5.5$$

Solving for $v_{\text{true}}$ yields

$$v_{\text{true}} = \frac{11k + 0.11}{1 + 11k} \cdot 50 = \frac{110.11}{111} \cdot 50 = 0.991 \cdot 50 = 49.6 \text{ mph}$$
Models can be
- static ($x = Ku$).
- dynamic ($\dot{x} = -x + u$).

Physics-based Modeling
- Mechanics and circuits define ODEs
- Identify states (position, voltage, et c.)
- Identify governing differential equations.

Newton invented ODE models in 1684:
- Newton’s Second Law: ($x$ is position)
  \[ \frac{d^2}{dt^2} x(t) = F(t)/m \]
- $x(t)$ is a state (can also be a signal (output))
- $F(t)$ is a signal (input)
Nonlinear Differential Equations:

\[ \dot{x}(t) = f(x(t), u(t)) \]
\[ y(t) = g(x(t), u(t)) \]

Where

- This is a first-order differential equation
- \( u(t) \) is the Input
- \( y(t) \) is the Output
- \( x(t) \) is the state variable.
  - position, heading, velocity, etc.
- \( f, g \) are functions (possibly vector-valued).
Review: Equations of Motion (EOMs)

Linear Equations

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

\[ \ddot{x}(t) = b\dot{x}(t) + ax(t) \]

where

- \( a \) and \( b \) are constants.

Linear equations are better because

- The Laplace Transform exists (Lecture 4)
- Stability is easy
  - \( \dot{x} = ax \) is stable if \( a < 0 \) and unstable if \( a \geq 0 \).

Nonlinear Equations should be Linearized (Lecture 3)!!!
Types of EOM:

**Coupled Differential Equations:**

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

- The motion of \( x \) affects the motion of \( z \) and vice-versa.

**Higher Order Derivatives:**

\[ \ddot{x} = a\ddot{x} + b\dot{x} + cx \]

- Commonly obtained from Newton’s Second law.
  - Or anything with inertia....

AKA

\[ F = m\ddot{x} \]
\[ \ddot{x}(t) = \frac{1}{m}F(t). \]
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 \( x(t) \)

**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 $K$.
  
  \[ F_s(t) = -K x(t) \]

- **Damper Force:** Opposes motion in $\dot{x}$ with damping coefficient $c$
  
  \[ F_d(t) = -c \dot{x}(t) \]

- **Newton’s Second Law:**
  
  \[ m \ddot{x}(t) = F_s(t) + F_d(t) + f(t) \]

**System Model:**

\[
\ddot{x}(t) = -\frac{K}{m} x(t) - \frac{c}{m} \dot{x}(t) + \frac{1}{m} f(t) \\
y(t) = x(t)
\]
Once we have our dynamic model

\[
\ddot{x}(t) = -\frac{K}{m}x(t) - \frac{c}{m}\dot{x}(t) + \frac{1}{m}f(t)
\]

Differential Equations

\[
y(t) = x(t)
\]

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.

\[
s^2\hat{x}(s) = -\frac{K}{m}\hat{x}(s) - \frac{c}{m}s\hat{x}(s) + \frac{1}{m}\hat{f}(s)
\]

Differential Equations

\[
\hat{y}(s) = \hat{x}(s)
\]

Output Equation

which yields

\[
\hat{y}(s) = \frac{1}{ms^2 + cs + K}\hat{f}(s)
\]
Now, we add the dynamics of the wheel.

There are two states:

**States:**

- 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: Free Body 1

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

\[ m_c \ddot{x}_1(t) = F_{s1,c}(t) + F_{d,c}(t) \]
\[ = -K_1(x_1(t) - x_2(t)) - c(\dot{x}_1(t) - \dot{x}_2(t)) \]

**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:
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:

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

Combining the dynamics, we get the coupled system dynamics.

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

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

\[ \sum M(t) = I \alpha(t) = I \ddot{\theta}(t) \]

- \( \alpha(t) = \ddot{\theta}(t) \) is the angular acceleration.
- \( \theta(t) \) is the state (angular displacement).
- \( I \) is the moment of inertia.
- \( M(t) \) is the torque (moment).
Other Sources of Models

Voltage Laws

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

\[ \sum_{k} i_{k}(t) = 0 \]

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

\[ \sum_{k} V_{k}(t) = 0 \]

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

\[ V_{r}(t) = R i_{r}(t) \quad \frac{d}{dt} i_{L}(t) = \frac{1}{L} V_{L}(t) \quad \frac{d}{dt} V_{c}(t) = \frac{1}{C} i_{c}(t) \]
State-Space is a way of writing first order differential equation using matrices.

\[ \dot{\vec{x}}(t) = A\vec{x}(t) \]

where \( \vec{x}(t) \) 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

\[
\begin{align*}
\dot{x}_1 &= -x_1 + x_3 \\
\dot{x}_2 &= 2x_1 \\
\dot{x}_3 &= -x_2 + x_3
\end{align*}
\]

Writing equations in state-space has many advantages.
Consider the system

\begin{align*}
\dot{x} &= ax + by \\
\dot{y} &= cx + dy
\end{align*}

When we have multiple coupled equations: 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

\[
\dot{x} = Ax
\]

where

- \( x \) is a vector.
- \( A \) is a matrix.
Definition 2.

**State-Space Form** is a convenient way of representing linear multivariate or 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 compatible (the right size):

\[
A \in \mathbb{R}^{n \times n} \quad B \in \mathbb{R}^{n \times m} \\
C \in \mathbb{R}^{p \times n} \quad D \in \mathbb{R}^{p \times m}
\]

- \( u \in \mathbb{R}^m \) means \( u \) is a real vector of length \( m \).
- \( C \in \mathbb{R}^{p \times n} \) means \( C \) is a matrix with \( p \) rows and \( n \) columns.
Putting Things in State-Space Form

Reducing Higher Order Dynamics

When we have higher order derivatives,

\[ \ddot{x}(t) = a\dot{x}(t) + bx(t) + u(t) \]
\[ y(t) = x(t) + u(t) \]

we can put it in state-space form by

- Introducing new variables.

**Procedure:**
- Define a new variable for every derivative term except for the highest order one.
  - 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.

So, for example, we would have 3 equations

\[
\begin{align*}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_3(t) \\
\dot{x}_3(t) &= ax_2(t) + bx_1(t) + u(t)
\end{align*}
\]
Putting Things in State-Space Form

Using our first-order equations:

\[
\begin{align*}
\dot{x}_1(t) &= x_2(t); \\
\dot{x}_2(t) &= x_3(t) \\
\dot{x}_3(t) &= ax_2(t) + bx_1(t) + u(t)
\end{align*}
\]

We construct the matrix representation:

\[
\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ b & a & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)
\]

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

So that

\[
A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ b & a & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 \end{bmatrix}
\]
State-Space Form: 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 \( z_i \)

\[ z_1(t) = x_1(t) \quad z_2(t) = \dot{x}_1(t) \quad z_3(t) = x_2(t) \quad z_4(t) = \dot{x}_2(t) \]

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

\[
\begin{align*}
\dot{z}_1(t) & = z_2(t) \\
\dot{z}_2(t) & = -\frac{K_1}{m_c} z_1(t) - \frac{c}{m_c} z_2(t) + \frac{K_1}{m_c} z_3(t) + \frac{c}{m_c} z_4(t) \\
\dot{z}_3(t) & = z_4(t) \\
\dot{z}_4(t) & = \frac{K_1}{m_w} z_1(t) + \frac{c}{m_w} z_2(t) - \left( \frac{K_1}{m_w} + \frac{K_2}{m_w} \right) z_3(t) - \frac{c}{m_w} z_4(t) + \frac{K_2}{m_w} u(t)
\end{align*}
\]

\[
y(t) = \begin{bmatrix} z_1(t) \\ z_3(t) \end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{z}_1(t) \\
\dot{z}_2(t) \\
\dot{z}_3(t) \\
\dot{z}_4(t)
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
-\frac{K_1}{m_c} & -\frac{c}{m_c} & \frac{K_1}{m_c} & \frac{c}{m_c} \\
0 & 0 & 0 & 1 \\
\frac{K_1}{m_w} & \frac{c}{m_w} & -\left( \frac{K_1}{m_w} + \frac{K_2}{m_w} \right) & -\frac{c}{m_w}
\end{bmatrix}
\begin{bmatrix}
z_1(t) \\
z_2(t) \\
z_3(t) \\
z_4(t)
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
0 \\
\frac{K_2}{m_w}
\end{bmatrix} u(t)
\]

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