MEEN 432 –Automotive Engineering

Fall 2026

Instructor: Dr. Arnold Muyshondt

Acknowledgement: Most of the material for this class was developed by Dr. Swami Gopalswamy

Lecture 1: Introduction to Automotive (Systems) Engineering

- Course Introduction
  - What is Automotive Engineering
- Simulation-Based Analysis of a simple mechanical system
- Project 0 Roll-out

Course Description

- Introduction to vehicle dynamics; application of engineering mechanics principles to analysis of acceleration and braking, cornering and handling; analysis and design of drive train, suspension, brakes and tires to achieve desired performance. Introduction to Unconventional Powertrains (Electric, Hybrid-Electric and Fuel Cell vehicles)

General Information

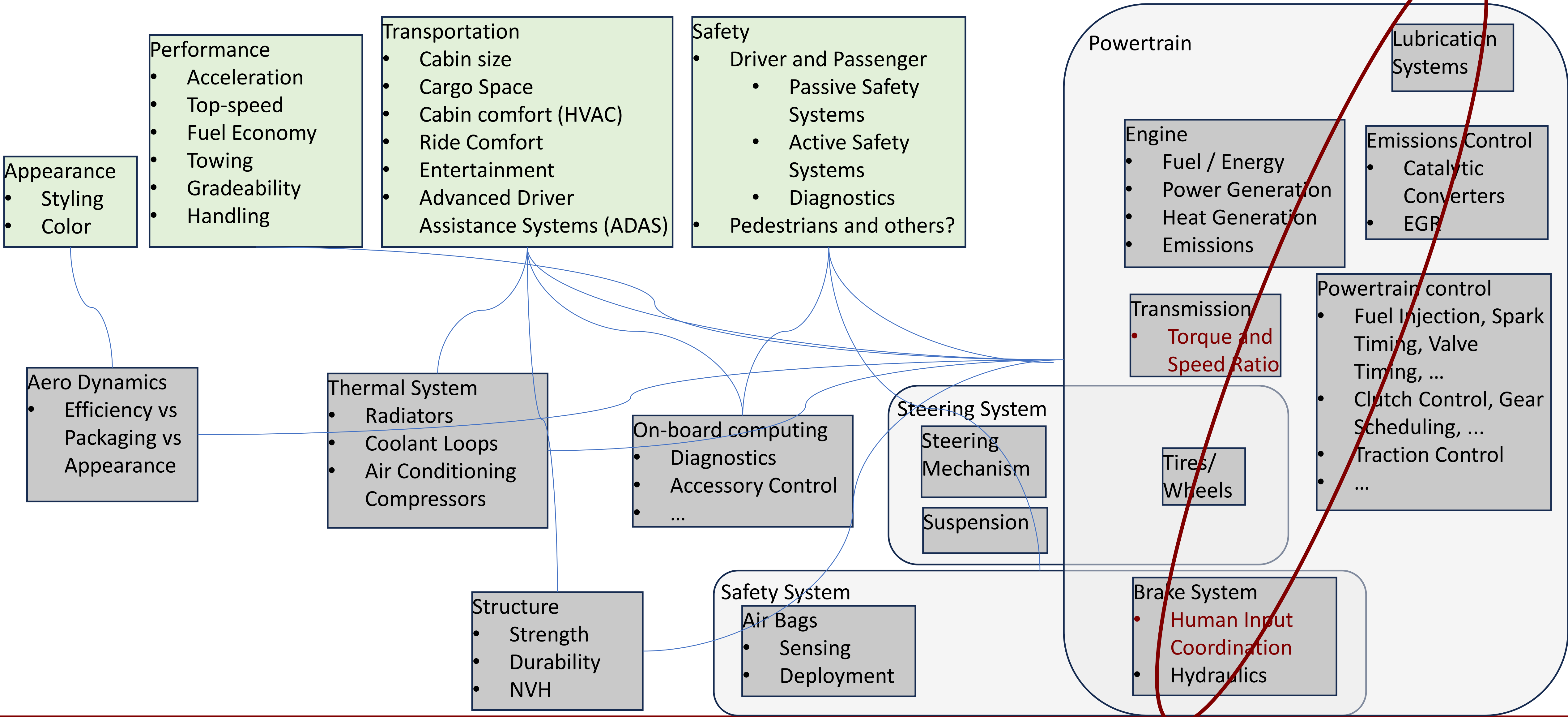
- **Instructor**:
  - Dr. Arnold Muyshondt, 531 MEOB, Phone: 505-977-1998, Email: amuyshondt@tamu.edu
  - Office Hours: By appointment most of the week 8:00 AM – 4:00 PM
- **Teaching Assistant** :
  - Allen George Phillip, y262u297@tamu.edu
  - Office Hours: TBD
- Lecture: MWF 9:10 – 10:00 AM Zach 218

General Information

- Project Based Course:
  - There will be four projects that will be the basis for grading with weights 10%, 25%, 25% and 40%. Timely completion of the projects is expected.
  - Project progress will be monitored through appropriate submission requirements on a regular (e.g. weekly) basis. Weekly progress on the project will constitute 15% of project scores.
  - Projects 1, 2, 3 and part of 4 will be performed in teams of ~4 people. Project 4 will also have an individual sub-project. For team projects, the project evaluation will also consist of a “peer-evaluation” component that could significantly influence each individual’s score relative to the overall project evaluation.
- General Outline (but will not necessarily be followed always):
  - M, W – Lectures
  - F – Demos, other project related materials, special lectures by TA

A user-centric view of an automobile

Scope of Class: Systems Engineering View of Automobile Engineering



Learning Outcomes

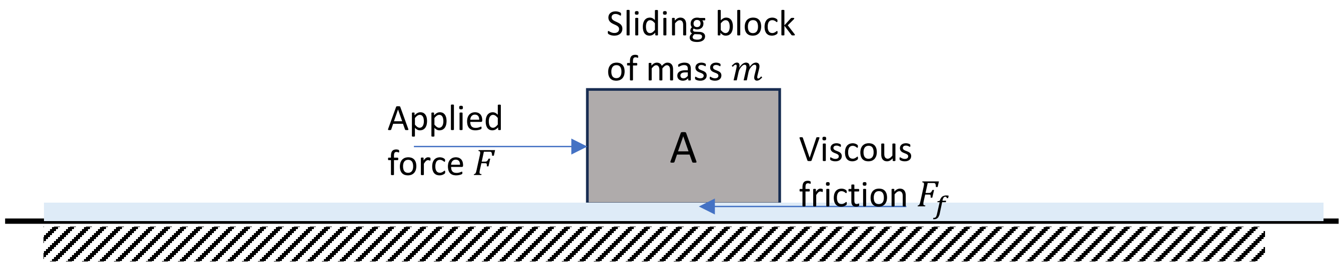
Course Goals: At the end of this course, the students should be able to

- Identify and build mathematical models for basic sub-systems of an automobile such as the power train and the steering;
- Simulate the dynamic response of the vehicle using MATLAB from the inputs from the driver and the environment;
- Team with other students to simulate the dynamic response of an automobile and write a technical paper
- Have an introductory understanding of major trends in the automotive industry – in terms of unconventional powertrains and automated driving.

Systems Engineering

- Many definitions abound. For the class:
  - Modeling the dynamics of various (automotive) subsystems at a level of abstraction that captures the impact on vehicle level performance and fuel economy
    - Such subsystems will also include feedback control systems embedded within
  - Integrating the subsystem models and creating simulation models at the vehicle level that can be used to predict "performance and fuel economy"
  - Perform parametric studies on these models to perform:
    - What-If analysis:
      - If I encounter a steady 5% grade and starting from rest, how long will it be before I can catchup this other vehicle that just passed me at 50 mph?
      - If I lose my ability to be in top gear, how does my maximum speed and 0-to-60mph time change?
    - Design Optimization:
      - How should I size my battery so I run max number of loops on a race track before my juice runs low?
- The models that are appropriate for systems engineering are captured as Differential Algebraic Equations (DAEs), which can often be simplified to Ordinary Differential Equations (ODEs)

Simulating a simple mechanical system



- Consider a simple mechanical system as above
  - You are tasked with determining the speed of the mass as a function of time
  - If your car lost its wheels and you start sliding on slushy mud, perhaps the above simple model might have a resemblance to reality!
- The first thing to do is to model the system behavior – i.e. write the equations of motion (EOM) (sometimes also referred to as state equations)
  - $m\frac{d}{dt}v \;:=\; m\,\dot{v} = F \;-F_f := F \;-b\,v$ starting from initial v_0

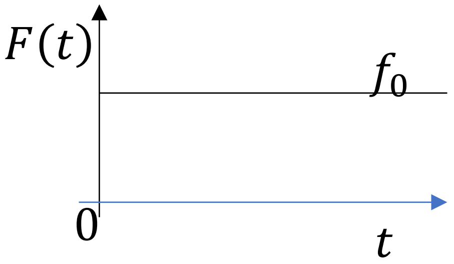
Derivative w.r.t. time

“by definition=“

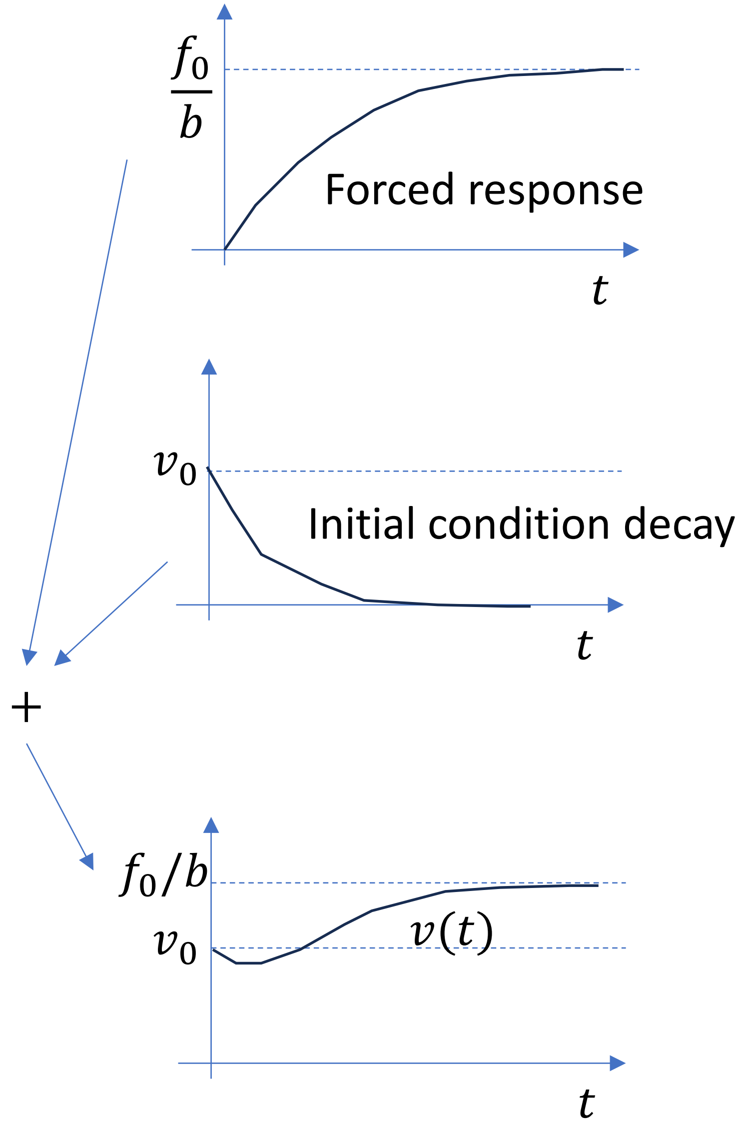
Assuming linear viscous friction force

Analytical solution approach

- Take the Laplace Transform of the ODE (Remember 364)
  - $s\,m\,V(s) - m\,v_0 = F(s) - b\,V(s)$
  - Or, $V(s) = \frac{1}{sm+b}\left(F(s) + m\,v_0\right)$
- Consider the simple case, when the applied force is a "step" function:
  - Then $F(s) = f_0\frac{1}{s}$



- Thus, $V(s) = \frac{f_0}{s(sm+b)} + \frac{mv_0}{sm+b}$
- Partial fraction expansion gives:
  - $V(s) = \frac{f_0}{m}\left(\frac{\left(\frac{m}{b}\right)}{s} - \frac{\left(\frac{m}{b}\right)}{s+\frac{b}{m}}\right) + \frac{1}{s+b/m}v_0$
- Then we can take the inverse Laplace to get
  - $v(t) = \frac{f_0}{b}\left(1 - e^{-\frac{b}{m}t}\right) + v_0 e^{-\frac{b}{m}t}$

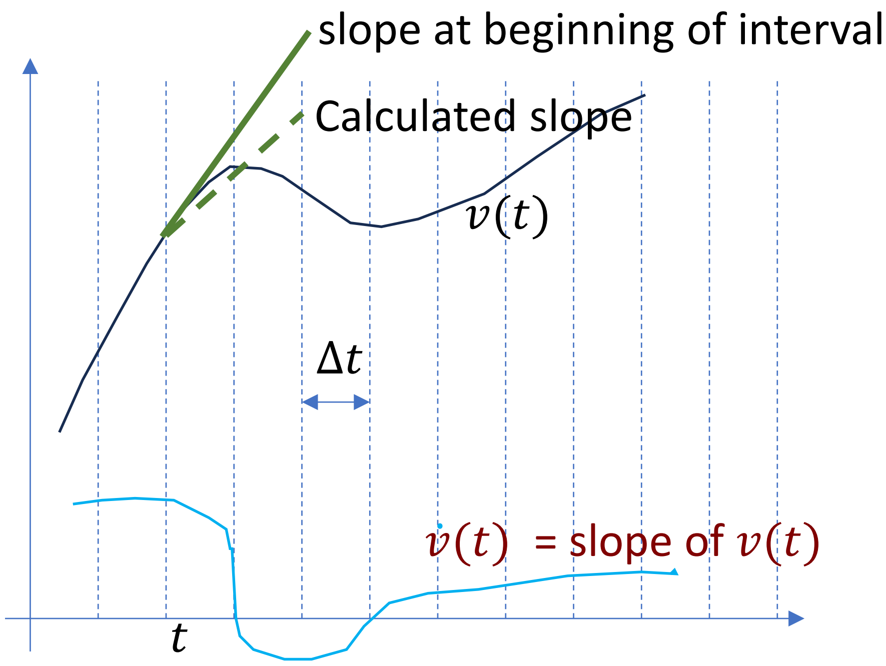


Limitations of the analytical approach

- We made a lot of assumptions in order to be able to solve the problem and get such a nice analytical solution:
  - System is Linear
    - Damping could be nonlinear; we could have columbic friction, …
  - Forcing function was a step
    - It could be an arbitrarily changing nonlinear, even discontinuous function … then?
- System was simple!
- The analytical method of solving is very useful in getting very good insights to your system
  - However, real-life systems – in particular, automotive systems, are <u>complex</u>, <u>nonlinear</u>, and constantly have to deal with <u>time-varying inputs</u>
- So, what is the solution approach:
  - Numerical Analysis

Numerical Integration

- By the definition of a "derivative":
  - $\dot{v}(t) = \lim_{\Delta t \to 0} (v(t + \Delta t) - v(t))/\Delta t$
- A simple approximation, with Δt sufficiently small gives:
  - $\dot{v}(t) \approx \frac{v(t+\Delta t)-v(t)}{\Delta t}$
- This will allow us to calculate (recursively, starting at time $t = t_0$, where $v(t_0) = v_0$ is known), the value of $v(t)$ for all time $t \geq t_0$
- For our specific problem we write:
  - $v(t + \Delta t) \approx v(t) + \Delta t\, \dot{v}(t) = v(t) + \Delta t \frac{1}{m} (F(t) - b\, v(t))$



Simulation Based Analysis

- Using Numerical Integration, we can now calculate the velocity of our system, given a starting velocity:
  - Algorithm:
    - Set initial value: $v(0) = v_0$
    - Repeat until end of time: $v(t + \Delta t) = v(t) + \Delta t \frac{1}{m} \left(F(t) - b\, v(t)\right)$
- The approximation that we used is one of the simplest approximations possible – called the "Euler Approximation" or "Euler Integration" method
  - There are other more sophisticated numerical approximations, and correspondingly different numerical "integration" algorithms
- Some key advantages:
  - The forcing function can be any function (not just a step)
  - The damping can be a complex function $F_f = \phi\left(v(t)\right)$ for some nonlinear function $\phi()$
- There are many COTS software tools that can perform this numerical integration very efficiently
  - In this class we will use MATLAB/Simulink as the simulation software environment

Project "0" (individual)

- All projects will be simulation-based and will use MATLAB/Simulink.
  - Make sure you get access to MATLAB/Simulink. You will be using version R2025
  - Work through tutorials so you know how to build simulation models and perform simulations using Simulink
    - Basic MATLAB tutorial: https://matlabacademy.mathworks.com/details/matlab-onramp/gettingstarted
    - Approximately 2 to 3 hours
      - Complete the projects, send individual completion certificate
    - Basic Simulink Onramp tutorial: https://matlabacademy.mathworks.com/details/simulink-onramp/simulink
      - Approximately 2 to 3 hours
      - Complete the projects, send individual completion certificate
- All submissions will be through CANVAS
- All Tutorial completions should be submitted to Project 0 assignment in CANVAS
- Strongly recommended:
  - https://matlabacademy.mathworks.com/details/simulink-fundamentals/slbe
  - https://matlabacademy.mathworks.com/details/matlab-fundamentals/mlbe