

ENGR 010 Group Project - Mechanical Engineering Option

Vehicle Dynamics Analysis Project

Overview

In this group project, you will develop a Python-based application to analyze vehicle performance data and simulate basic vehicle dynamics. Your team will create an interactive dashboard that mechanical engineers could use to evaluate vehicle performance metrics, analyze efficiency, and visualize motion characteristics.

One important part of this project is the use of Generative AI tools like ChatGPT to complete the assignment. AI tools are going to become increasingly useful and more widely used in the job market you're headed into, and we want you to be prepared for it. So, while you need to understand everything the AI creates on your behalf, for this assignment we are not only allowing you to use AI, we are requiring that you make use of it. Let's see what amazing work you can create when you use AI as your personal assistant.

Project Description

Background

Understanding vehicle dynamics is crucial for automotive design and performance optimization. Mechanical engineers regularly analyze data from vehicle tests to assess performance, efficiency, and safety characteristics. This project simulates a simplified version of the tools used in automotive engineering practice.

Project Specifications

Your team will create a Python application that:

1. Data Management
 - Imports vehicle test data from CSV files
 - Stores measurements including:
 - Speed
 - Acceleration
 - Engine RPM

- Fuel consumption
- Distance traveled
- Handles multiple test runs and driving conditions

2. Analysis Features

- Calculate basic statistics (mean, median, standard deviation)
- Determine acceleration characteristics
- Calculate fuel efficiency metrics
- Analyze braking performance
- Perform basic motion analysis

3. Visualization Features

- Time series plots of vehicle parameters
- Speed-acceleration curves
- Fuel efficiency maps
- Interactive dashboard for performance monitoring

Sample Data

Sample data will be provided from three vehicle test runs, with measurements taken at 0.1-second intervals during various driving conditions.

Learning Objectives & Requirements

- Apply fundamental Python programming concepts in a real-world context
- Gain experience with scientific computing libraries
- Develop skills in data visualization and analysis
- Practice collaborative software development
- Experiment with using Generative AI to assist with writing code, developing concepts, etc.

Technical Requirements

Your project must implement the following Python concepts:

1. Variables and Data Types

- Appropriate use of numerical data types for measurements
- Clear and consistent variable naming
- Use of constants for physiological parameters

2. Control Structures

- Multiple if/elif/else decision structures
- At least two different types of loops (for, while)
- Error handling using try/except

3. Functions and Organization

- Minimum of 5 custom functions
- Proper use of parameters and return values
- Documentation using docstrings
- Code organization into multiple Python files

4. Data Structures

- Lists for storing time-series data
- Dictionaries for storing patient parameters
- NumPy arrays for signal processing

5. Scientific Computing

- Basic signal processing using NumPy
- Data manipulation using Pandas
- At least 3 different types of plots using Matplotlib
- Statistical analysis of the data

6. Creativity

- Please add some fun or interesting features to your application to make it your own. Lean on AI to do this if you aren't sure what to do beyond what we have learned in class. For example, you could make your project into a web app instead of a notebook, try using a different library for visualizations, or add some own sample data of your own to do another type of visualization / analysis.
- Whatever you decide to do, be creative and have fun with it!

Submission Requirements

1. Jupyter Notebook File Submission

- Code used to complete project
- Written descriptions that complement code and explain thought process (similar to Engineering Assignments)

- If you want to turn in your code in a different format - i.e. a bunch of python files - that is absolutely allowed but not required.

2. AI Prompts with Annotations

- You should also submit any and all prompts to ChatGPT or other Gen-AI systems (e.g. Claude, Gemini, etc.) that you used for this project
- Please annotate each prompt with a short description that describes your thought process for that inquiry. Some potential things to write about in those annotations:
 - Why did you ask AI that question?
 - Did AI answer the question you asked well? How do you know? (e.g. the answer matches what we learned in the class on loops)
 - If you updated a prompt after getting an unsatisfactory answer, what was your thought process for the updated prompt?

3. Reflection on using AI for programming

- Did you enjoy using AI to help with this project?
- What type of questions/prompts did AI work well for? What about things it didn't do well?
- Did using AI save you time?
- Were the concepts used in the AI-generated code things that you already had seen and understood? If not, why did you decide to use them in your final project?

4. Final Presentation

- 5-minute demonstration
- Short Q&A
- All team members must participate

Grading Breakdown

150 points for successful project - no partial credit - based on the following:

- Functionality and Implementation
- AI prompts, annotations, and reflection
- Creativity & innovation
- Documentation
- Final Presentation