

Team Contributions: Rev 0 CVT Simulator

Team #17, Baja Dynamics
Grace McKenna
Travis Wing
Cameron Dunn
Kai Arseneau

This document summarizes the contributions of each team member for the Rev 0 Demo. The time period of interest is the time between the POC demo and the Rev 0 demo.

1 Demo Plans

For the Rev 0 demo we will have a Unity application that can simulate a CVT. It is comprised of the following components:

Unity FrontEnd

- An input page that will take in 8 input parameters from the user.
- The input page will have default values in each input field based on the cars current settings.
- A results page that allows the user to play/pause the simulation.
- On the results page there will also be the following features:
 - Gauges that will display the current speed and RPM of the car.
 - A 3D model of the CVT that will animate based on the input parameters.
 - A mini car to show the position change and incline of the road.
- A download button that will allow the user to download the results of the simulation in the form of a CSV.
- An input table that displays the user there input parameters.

Python Backend

Our demonstration will center around the simulation of the CVT and how the users inputs affect the results. The user will first use the default values provided, run the simulation and observe the visual results. Then the user will go back and change the inputs and then see how the results of the simulation change.

This aims to demonstrate how our simulation models the CVT based on the inputs provided by the user. The backend intends to function as follows:

- The python backend will receive the input from the front end and send back a response.
- It will then simulate the drivetrain and dynamics of our custom baja vehicle. This includes the engine, transmission, gearbox, wheels, air resistance, gravity and rolling resistance.
- Includes an ODE solver to solve a 2nd order ODE given the inputs from the front end and writes all datapoints to a CSV file.

2 Team Meeting Attendance

Student	Meetings
Total	5
Grace McKenna	5
Travis Wing	5
Cameron Dunn	5
Kai Arseneau	5

3 Supervisor/Stakeholder Meeting Attendance

Student	Meetings
Total	0
Grace McKenna	0
Travis Wing	0
Cameron Dunn	0
Kai Arseneau	0

In between the POC and Rev 0 (as of now) there has not been a supervisor meeting. This is mainly due to the fact that we have been working on the software implementation of the project and have not had any questions or concerns

that we needed to ask our supervisor. However we do plan on meeting with our supervisor before we present Rev 0.

4 Lecture Attendance

Student	Lectures
Total	1
Grace McKenna	1
Travis Wing	0
Cameron Dunn	0
Kai Arseneau	0

5 TA Document Discussion Attendance

Student	Lectures
Total	1
Grace McKenna	1
Travis Wing	1
Cameron Dunn	1
Kai Arseneau	1

6 Commits

Student	Commits	Percent
Total	40	100%
Grace McKenna	8	20%
Travis Wing	10	25%
Cameron Dunn	8	20%
Kai Arseneau	15	35%

7 Issue Tracker

Student	Authored (O+C)	Assigned (C only)
Grace McKenna	1	2
Travis Wing	5	3
Cameron Dunn	0	2
Kai Arseneau	18	5

The issue tracker is a bit skewed because of how our team meetings were organized.

Kai Arseneau has the most issue authors because during the meeting where we scoped out the project and made issues he was the one who was typing and creating the issues while the rest of us were discussing each issue and what needed to be done. This is also why Travis Wing has some authors because after a while he took over the typing and creating of issues from Kai.

As far as the assigned issue tracking that is much more even across the board as we have been evenly distributing the work amongst ourselves.

8 CICD

We will utilize CICD to automatically lint, format and test our code.

For the python backend we will be using the following tools:

flake8 - A Python linter that checks for PEP8 compliance.

black - A Python code formatter that will ensure consistent code style.

unittest - Python's built-in testing framework that will be used for unit testing.

coverage - A testing framework for Python that will be used for code coverage.

For the Unity C# frontend we will be using the following tools:

SonarLint - C# linter that checks for code quality and security vulnerabilities.

StyleCop - C# linter that checks for code style and formatting.

UTF - A testing framework for C# that will be used for unit testing.

UTR - A testing framework for Unity that will be used for unit testing and code coverage.