InceptEV: 2024 Case Study Project

Sponsor Mentors: Alex Wadell and Kian Pu
Student Cohort: Justin Wong, Ryan Stephen, Tanush Mittal, Yash Dumpeta

Fall Semester

Table of Contents

Problem Statement and Solution

InceptEV is a startup focused on pioneering the electric revolution, specifically in truck fleets. Through advanced simulation technology, InceptEV addresses critical challenges in commercial fleet electrification. Fleet operators face significant hurdles when transitioning to electric vehicles, including range anxiety, charging infrastructure optimization, and high initial costs. The company's simulation-first approach enables fleet managers to optimize their transition before committing resources, reducing operational risks and total ownership costs.

Project Intent

Our project aims to perform a rigorous evaluation of InceptEV's advanced simulation technology in a thorough case study. The initiative embarked on early iterations of validation — with innovative methods such as leveraging data from social media insights, like a Tesla Semi trip tweet, to gauge the accuracy of simulated projections against real-world performance as shown in Figure 1. This case study will provide InceptEV and its users with information on the accuracy of the simulation software, which simulates a vehicle’s conversion from gas to electric. Since InceptEV utilizes a simulation-first approach, this is incredibly valuable information as users will have more calibrated confidence in the results the simulation returns. InceptEV will also gain insight on what factors affect the accuracy most, and what they should focus on for improvement.

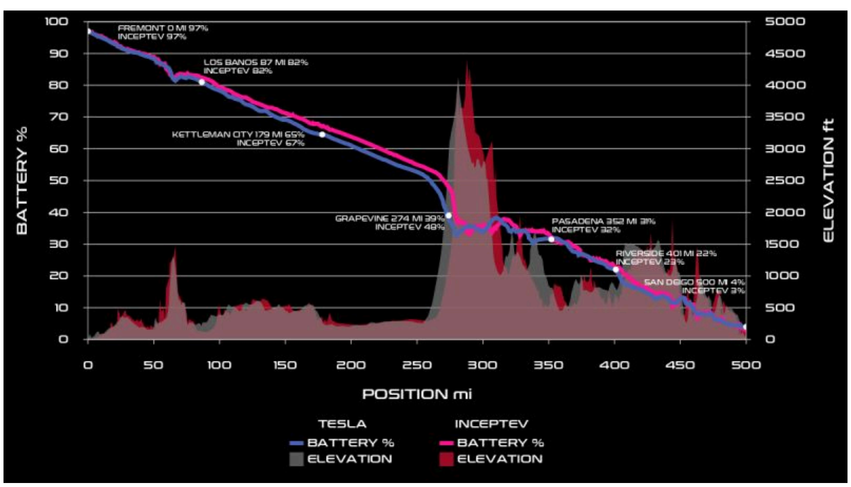

Figure 1: Crude Validation Method of InceptEV Prior to Project

Project Summary

Our project aims to conduct a validation study to determine and validate the accuracy of Incept-EVs simulation software. We want to evaluate its performance in various scenarios, so we can determine the specific factors that affect its accuracy. Our project consists of two main phases: data collection and data analysis. During the data collection phase, we are collecting 4 types of data:

1. The first type is **telematics** data, which is data related to the vehicle such as speed, acceleration, location, and state of charge.
2. The second type is **environmental** data, which is data related to the environment such as elevation, wind speed, and temperature.
3. The third type is **driver** data, which is related to the habits of the driver such as aggressive drivers, or safe drivers.
4. The final type is **HVAC** data, which is the difference between the outdoor temperature and the vehicle's set temperature.

Upon collecting and cleaning our data, we will use our recorded parameters to simulate the same data we collected on InceptEV's simulation software. We will then analyze the error, considering all the various factors listed above, and provide a detailed error analysis to InceptEV, specifically highlighting which variables affect accuracy the most, the overall error, and other important insights.

OKRs

We created a list of key results for the objective we hope to achieve during the two semester-long project. These have been listed below, and include the status of these items.

Objective: Conduct a **validation study** on InceptEV’s simulation software by comparing real-time telematics data with simulated data.

Key Results:

1. **Complete** the setup for data collection (Completed)
2. **Perform** the first test drive to validate the data collection setup (Completed)
3. **Establish** the pipeline for data preparation and post-processing (Completed)
4. **Design** of our experiments (Completed)
5. **Accumulate** 3 hours of driving data from Ford Mach-E (Completed)
6. **Finalize** error formulas and metrics for analysis (Completed)

Deliverables

<u>Phase 1 Deliverables (Winter Semester)</u>:

- **Establish** data collection sources (completed Mach-E setup) for testing and validation of simulation
- **Develop a** comprehensive database of 3 hours of driving data for error analysis on simulation

<u>Phase 2 Deliverables (Fall Semester)</u>:

- **Design** of experiments for 2 different routes testing 5 factors: Urban/Suburban/Mixed, Load, Driving Behavior, Temperature Difference, and Time of Day
- **Documented** methodology of 5 experiments, detailed in the final case study report

Project Plan

At the time of writing this report, Figure 2 shows our most updated project plan for both semesters. As shown, we have split it into 4 major phases: Planning, Data Collection, Error Analysis, and Report. Our biggest difference from the original plan, shown in Figure 3, was our method for collection of State of Charge (SoC) and redistribution of phases.

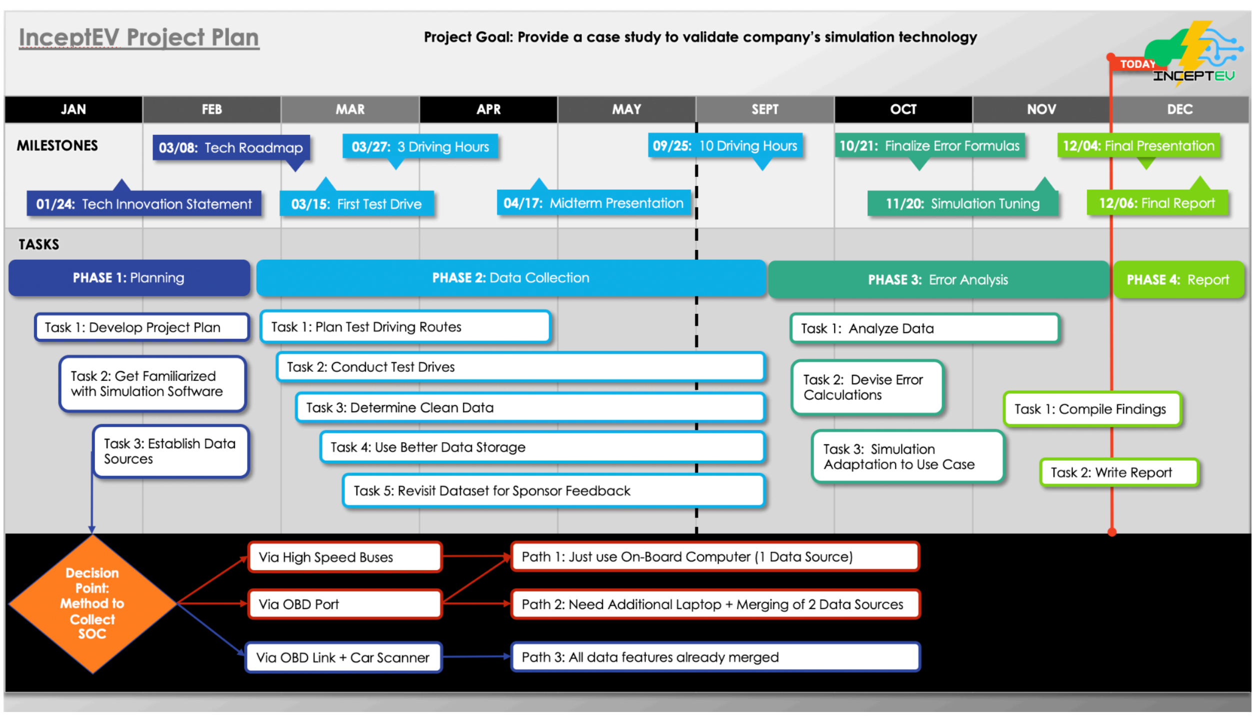

Figure 2: Up-To-Date Project Plan (Updated 12/01)

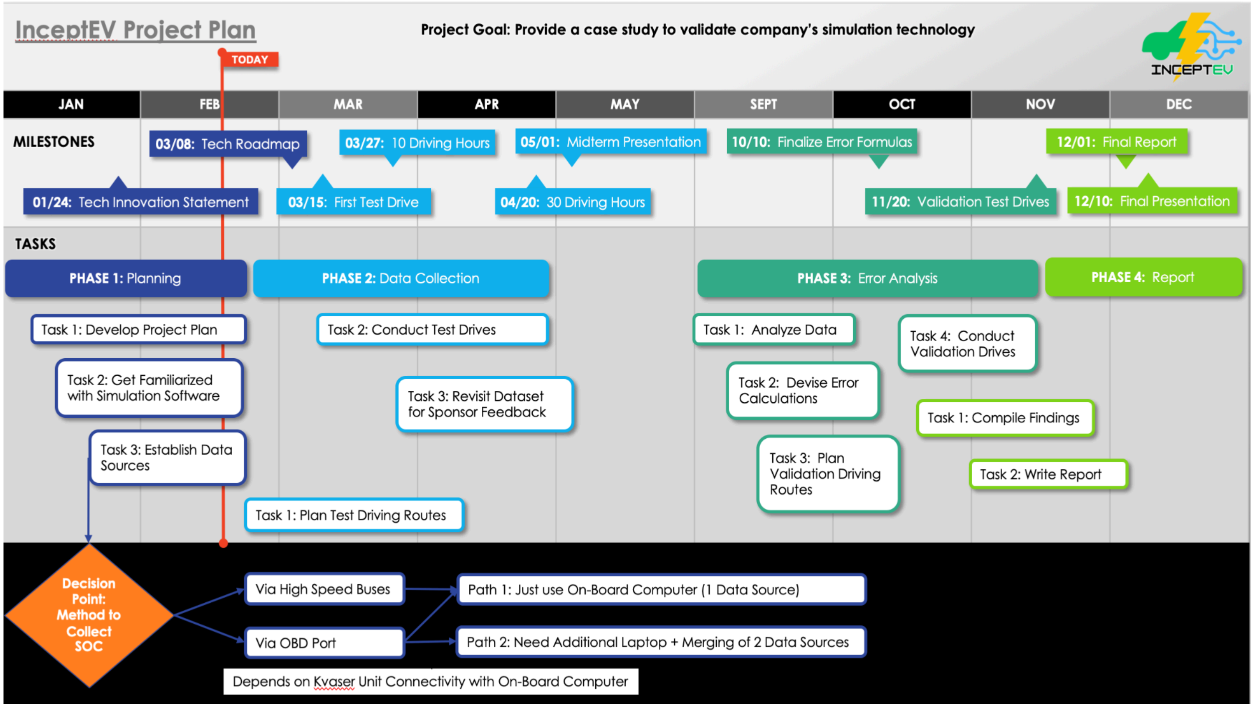

Figure 3: Initial Project Plan (Updated 02/20)

At the start of the project, our team deliberated over two primary methodologies for gathering State of Charge (SoC) and additional vehicle diagnostics data: via the high-speed Controller Area Network (CAN) bus or the On-Board Diagnostics (OBD) CAN bus. Utilizing the high-speed CAN bus emerged as the ideal option initially since it would seamlessly integrate with the acquisition of other crucial data points, such as vehicle speed, accelerator pedal position, braking pressure, and location. Nonetheless, extracting data directly from Ford's proprietary vehicle systems presented significant challenges, necessitating the consideration of third-party facilitators for access. Consequently, our team decided on an alternative route—employing the OBD CAN bus in conjunction with a Kvaser Leaf Light interface to procure the SoC data. This approach, as depicted in Figure 4, involves aggregating data from disparate sources across multiple computers and subsequently synchronizing these datasets along a joint timeline to maintain data coherence.

Advancing further into the project timeline, we identified and implemented a more streamlined data collection strategy. As outlined in the revised Project Plan section, this strategy uses an OBDLink LX tool paired with a Car Scanner application to interface directly through the vehicle's OBD port. This setup, which we currently employ, has allowed for more efficient data extraction and has been vital to maintaining project momentum. With several tasks in Phase 2 still underway, we anticipate their progression into the next semester. Despite these ongoing elements, their anticipated resolution aligns with our project schedule, keeping us on course for timely completion.

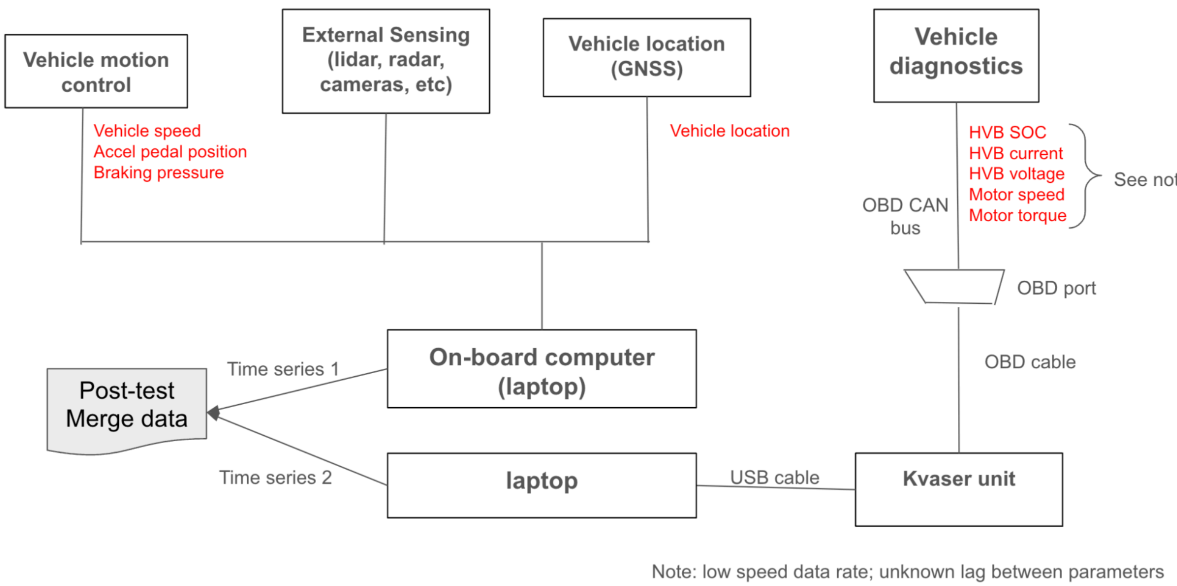

Figure 4: Initial Data Collection Architecture

Our project timeline required strategic redistribution of resources and time allocation to address several technical challenges. The most significant adjustment involved extending the data collection and error analysis phases while condensing the reporting phase.

The extended data collection and analysis period proved essential for adapting to our specific testing requirements at MCity and accommodating the transition from Tesla Model 3 to Ford Mach-E vehicles. This extension allowed us to gather more diverse datasets and properly account for the differences between Boston and Ann Arbor driving conditions. The additional time was crucial for implementing necessary code modifications and enhancing our simulation capabilities to match these new parameters.

The decision to compress the reporting phase, while introducing certain risks such as reduced review time and potential communication constraints, aligns with the company's current development stage. We prioritized comprehensive data collection and thorough analysis to ensure the long-term value of our findings. This approach provides more robust validation of the simulation technology while maintaining the essential quality of our deliverables.

Our timeline redistribution demonstrates a strategic balance between thorough testing and efficient project completion. The emphasis on extended data collection and analysis ensures that our conclusions are built on comprehensive, well-validated data, even with a condensed reporting timeline. This methodological adjustment better serves the company's needs by providing more reliable and adaptable testing results while addressing the complex technical challenges we encountered.

Project Progress

Data Collection Methodology

During the Winter semester, significant headway has been made in establishing our data collection framework. In collaboration with MCity staff, specifically Allen Dobryden and Vince Belanger, we have finalized our data collection methodology using the Ford Mach-E. Our process now incorporates the use of an OBDLink device shown in Figure 7 that interfaces with the vehicle's OBD port. Data retrieval is facilitated through the Car Scanner application on a smartphone, which also serves to store the downloaded data files. The app's simple and intuitive interface is shown in Figure 8.

Figure 7: OBDLink Device

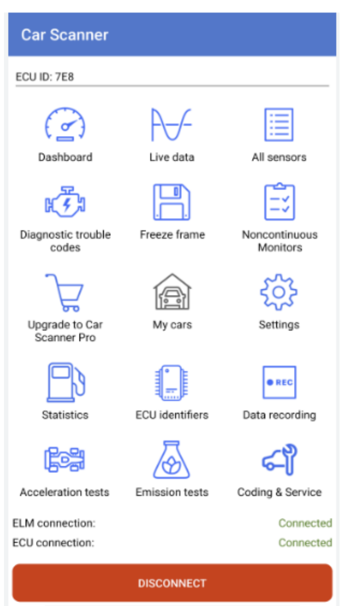

Figure 8: Interface of Car Scanner App

Figure 9 provides a snapshot of the initial data gleaned from a 25-minute test drive carried out by our team. Presented in the form of a Pandas DataFrame, the raw data exemplifies what we collect during a typical session. Utilizing Python within the Google Colab Integrated Development Environment (IDE) will be our standard procedure for the subsequent cleaning and analysis of this data. This streamlined process is a pivotal step forward in our project, ensuring we can analyze the high-quality data necessary for comprehensive testing and validation of InceptEV's simulation software.

	time	Altitude (GPS) (feet)	Latitude	Longtitude	HVB State of Charge (%)	Vehicle Speed High Resolution (mph)	Absolute pedal position D (%)	A/C Compressor Current (A)	Ambient air temperature (℉)
11000	13:32:57.490	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11001	13:32:57.571	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11002	13:32:57.611	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11003	13:32:57.723	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11004	13:32:57.773	NaN	42.299279	-83.700174	NaN	NaN	16.0	NaN	NaN
11005	13:32:57.814	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	66.2
11006	13:32:57.885	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11007	13:32:57.926	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11008	13:32:58.028	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11009	13:32:58.109	NaN	42.299279	-83.700174	NaN	NaN	NaN	0.000000	NaN
11010	13:32:58.190	NaN	42.299279	-83.700174	NaN	0.961184	NaN	NaN	NaN
11011	13:32:58.292	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11012	13:32:58.332	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11013	13:32:58.444	NaN	42.299279	-83.700174	96.326	NaN	NaN	NaN	NaN
11014	13:32:58.484	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11015	13:32:58.585	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11016	13:32:58.626	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11017	13:32:58.687	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	NaN
11018	13:32:58.738	NaN	42.299279	-83.700174	NaN	NaN	16.0	NaN	NaN
11019	13:32:58.778	NaN	42.299279	-83.700174	NaN	NaN	NaN	NaN	66.2

Figure 9: Example of Raw Data

Data Collection Challenges and Solutions

One of our primary challenges with the current data collection methodology is the asynchrony of the time measurements among different data points, which results in numerous empty values, denoted as NaN, within our data set. To overcome this issue, we have implemented data-cleaning procedures designed to rectify these gaps.

The core principle of our data cleaning method involves populating the empty cells with the most recent non-zero value that was recorded. This is accomplished using a specific function available within the pandas library, effectively ensuring our data set remains comprehensive and usable for analysis. The functionality and its application are detailed in Figure 10.

Subsequent discussions with the MCity team and Matt, our project sponsor from InceptEV, have led us to confirm that the data cleaning approach we have employed is satisfactory for our project needs. This consensus was reinforced upon reviewing the resulting cleaned data, which is depicted in Figure 11, demonstrating a structured and uninterrupted data flow suitable for the next stages of our analysis. The code that accomplished data cleaning can be found here: https://colab.research.google.com/drive/196dbMw1nUdtcSOd_IIWgNc7F3shxuvRn

```
cleaned_df.fillna(method = 'pad', inplace = True)
```

Figure 10: Data Cleaning Command

	time	Altitude (GPS) (feet)	Latitude	Longtitude	HVB State of Charge (%)	Vehicle Speed High Resolution (mph)	Absolute pedal position D (%)	A/C Compressor Current (A)	Ambient air temperature (℉)
11000	13:32:57.490	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11001	13:32:57.571	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11002	13:32:57.611	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11003	13:32:57.723	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11004	13:32:57.773	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11005	13:32:57.814	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11006	13:32:57.885	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11007	13:32:57.926	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11008	13:32:58.028	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11009	13:32:58.109	902.018038	42.299279	-83.700174	96.326	1.368958	16.0	0.000000	66.2
11010	13:32:58.190	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11011	13:32:58.292	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11012	13:32:58.332	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11013	13:32:58.444	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11014	13:32:58.484	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11015	13:32:58.585	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11016	13:32:58.626	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11017	13:32:58.687	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11018	13:32:58.738	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2
11019	13:32:58.778	902.018038	42.299279	-83.700174	96.326	0.961184	16.0	0.000000	66.2

Figure 11: Example of Cleaned Data

We also learned that the Car Scanner app only tracks the values the user sets, and the speed at which each entry is collected is inversely proportional to the total number of parameters. Thus we are working on optimizing the parameters we collect. This is particularly difficult because the Car Scanner has hundreds of parameter options to choose from, many of which have similar names but different functions. Thus we must test a variety of these variables to determine which ones are the best for our purpose.

Route Planning and Experiment Design

We have also made significant progress in route planning and experiment design. To make sure we are gathering as much diverse and reliable data, we are designing routes in various environments and conditions.

Route planning is critical, as it aims to capture a diverse range of driving conditions such as variations in speed due to different road types (residential or highway), stopping instances (at street lights, turning, and stop signs), and changes in elevation. As illustrated in Figure 12, our proposed route – formulated using Google Maps – has been carefully plotted to encompass these elements. The outlined route includes sections of highway driving and maneuvers through residential areas near Gallup Park, taking advantage of the stop signs and traffic lights on the University of Michigan's campus. It also spans a range of elevation from 965 to 768 feet.

We acknowledge the challenges in accounting for the rate of change in elevation. Ann Arbor's predominantly flat topography presents limitations in assessing the impact of steeper inclines. Although the elevation map shown in Figure 13 offers some insight, it falls short in providing specifics on gradient steepness, which is more relevant to our data collection objectives. To mitigate these challenges, we plan to explore alternative methods and tools that could provide us with the necessary level of detail regarding elevation profiles. InceptEV's simulation samples elevation using latitude and longitude which it is able to use to roughly calculate gradient steepness. Approval for this route strategy is pending confirmation from MCity's Alison Smith, and we remain open to making requisite adjustments based on MCity's response.

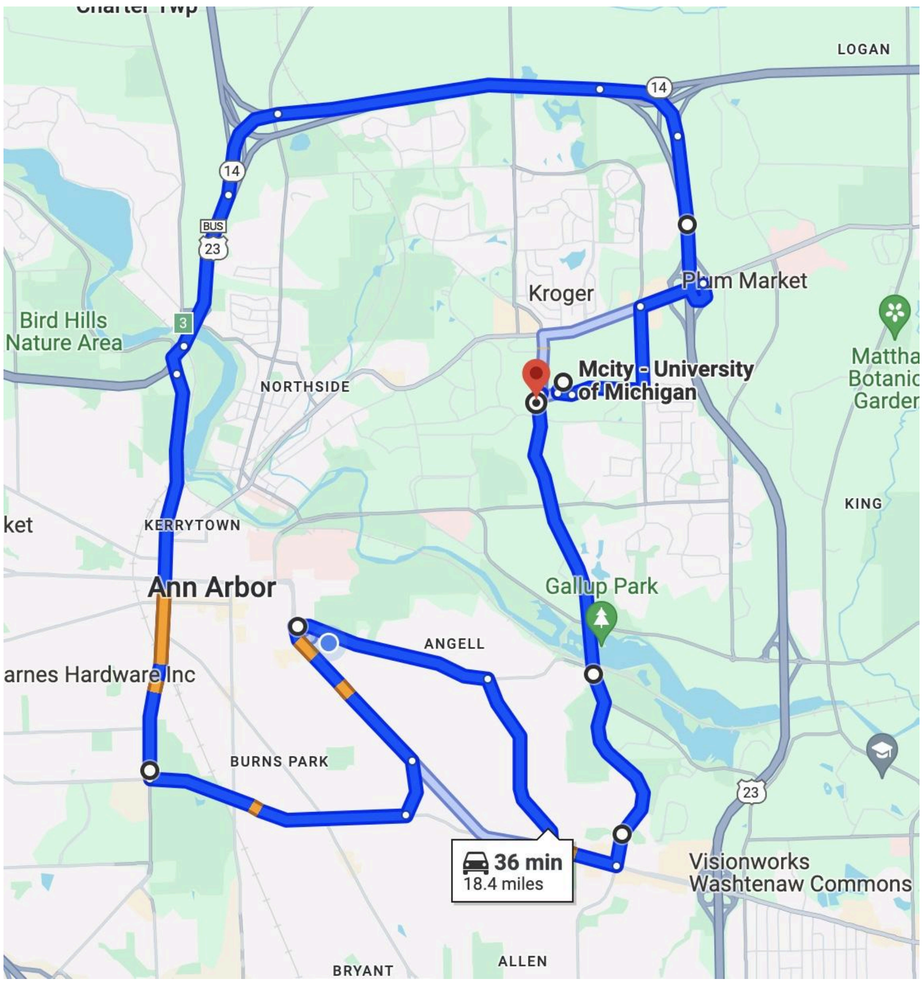

Figure 12: Example Test Drive Route

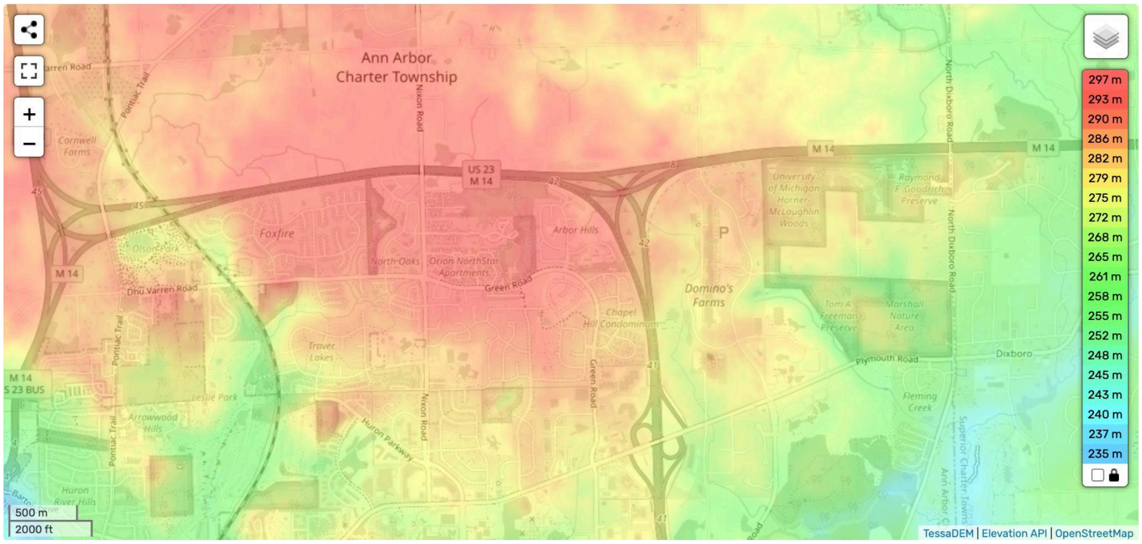

Figure 13: Elevation Map of Ann Arbor

Experimental Variables

The careful planning of our experimental design is crucial for obtaining reliable and insightful data. We have pinpointed several variables that can be controlled, including the initial State of Charge (SoC), the internal temperature setting, the type of road, the time of day, and the payload weight. A comprehensive table – depicted in Figure 14 – outlines the specific conditions that each of these variables will be subjected to during our test drives.

Variable	Description	Levels
State of Charge (SOC)	Battery charge level to start tests	25%, 50%, 75%, 100%
Internal Set Temperature	Temperature during tests	32°F (0°C), 59°F (15°C), 86°F (30°C)
Road Type	Type of test drive route	Urban, Highway, Mixed
Time of Day	Time when tests are conducted	Morning, Afternoon
Payload Weight	Weight of cargo and passengers	300 lbs, 450 lbs

Figure 14: Experimental Design for Controlled Variables

Given the impracticality of testing all possible variable permutations, we have developed a targeted test plan that prescribes specific values for each variable per test drive. This approach allows us to systematically assess the impact of each controlled variable, ensuring a comprehensive understanding of their effects on vehicle performance. As we progressed into the

Fall semester, we were not able to thoroughly test certain variables due to practical constraints and safety considerations. The State of Charge (SoC) was kept consistently high at MCity's request to prevent the risk of the vehicle becoming stranded. We also refrained from manipulating the internal temperature settings to avoid interfering with the Ford Mach-E's systems. As a result, we were unable to test the wide range of internal set temperatures we had initially planned, instead conducting our drives under relatively consistent ambient temperature conditions. Instead of isolating specific road types, we opted for a more holistic approach, incorporating a mix of urban, highway, and residential driving conditions in each test drive while also making sure to include elevation changes. Payload weight was simplified to be based on the number of occupants, assuming 150 lbs per person. These adjustments, while limiting some aspects of our original variable control, allowed us to collect more realistic and operationally feasible data. The resulting dataset, now available in a dedicated Google folder, provides a comprehensive view of the vehicle's performance under various real-world conditions in Ann Arbor, offering valuable insights for our simulation and analysis efforts. Our test drives, with the removal of columns we were unable to test for, are summarized in Figure 15.

Test ID	SOC	Temp	Route Type	Time of Day	Payload	Date
1	96	66.2°F	Mixed	Afternoon	300 lbs	03-05-2024
2	100	80.6°F	Mixed	Afternoon	450 lbs	03-07-2024
3	96	73.4°F	Mixed	Afternoon	300 lbs	09-06-2024
4	96	71.6°F	Mixed	Afternoon	450 lbs	09-17-2024
5	94	71.6°F	Mixed	Morning	300 lbs	09-25-2024

Figure 15: Test Matrix Showing Variable Combinations

Accompanying the telematics data gathered from the Ford Mach-E, there were 2 main weather variables we also needed to account for wind speed and wind direction. We collected this data using MCity's weather station, which provided new wind speed and direction data every 5 minutes.

To accommodate our specific use case with the Ford Mach-E in Ann Arbor (as opposed to a Tesla Model 3 in Boston in their paper), we made several code adjustments to the simulation. We first copied their project and put our new changes in a new project which can be found in examples > pjtl-student-study.

1. Updated the vehicles.yaml file to include Ford Mach-E specifications (mass, drag area, drag coefficient).
2. Implemented a new GPSRoute object in route_sampler.jl to use our collected GPS data directly.
3. Created a load_data.py script to format our driving and weather data for simulation input.

Running the simulation:

1. Activate a Python virtual environment. Note: make sure it is in Python 3.11 to avoid version issues.
2. Ensure you have your simulation input by running “python3 load_data.py”
3. If you do not already have the necessary map file (geofabrik/north-america/us/<state>.osm):
    a. Navigate to the tools directory
    b. Run “rm osmosis”
    c. Run “ln -s osmosis-0.48.3/bin/osmosis osmosis”
    d. Navigate back to the main directory
    e. Run “geofabrik/north-america/us/<state>.osm”
4. Navigate to the pjtl-student-study directory
5. Open the Julia interpreter with “julia --project”
6. Hit “]” to go into Pkg mode
7. Use “instantiate” to install the dependencies
    a. When running “instantiate” may initially fail because Incepts is not registered in this case:
        i. Run “dev ../../” to tell Julia where Incepts is located
        ii. Then you can run “instantiate”

```
julia --project
juila> # At the REPL hit ']' to go to Pkg mode
pkg> instantiate # Install deps
pkg> Hit backspace to go back to the repl
```

(edited)

For the incept_paper.jl examples it’s the same but hit:

```
pkg> instantiate # Will fail cause Incepts isn't registered
pkg> dev ../../ # Tell julia where incepts is located
pkg> instantiate # should be a no-op but yolo
```

Figure 16: Instructions to install dependencies (Steps 4-6)

8. In Julia, run “include(“student_study.jl”)”
    a. In the case of any PyBAMM-related errors:
        i. Run “using PyCall” before running student_study.jl
        ii. pyimport(“pybamm”)

These updates reflect our progress in data collection, experimental design, and simulation adaptation to meet the specific requirements of our project using the Ford Mach-E in Ann Arbor conditions.

Results

Our ultimate goal was to calculate error on state of charge and determine how accurately the simulation tracked that metric on various environmental conditions. Due to bugs in simulation and previously mentioned trouble designing experiments, we were unable to calculate errors for specific situations. Instead, we calculated root mean square error (RMSE) on all state of charge values.

- We calculated an RMSE of **0.088** over a scale of 100

This low error rate demonstrates that, even without isolating individual factors, InceptEV's simulation provides accurate SoC projections, which indicates its value as a predictive tool for electric fleets.

Next Steps

Electric vehicles (EVs) are highly sensitive to various environmental and operational factors, which can significantly impact energy consumption and range. While the current study has provided valuable insights, future case studies could delve deeper into key variables that influence EV performance, particularly in more challenging scenarios. By focusing on factors such as elevation changes, temperature extremes, and starting state of charge (SOC), we can uncover critical insights that enhance our understanding of vehicle efficiency, battery behavior, and overall performance under stress. The following recommendations outline areas for further exploration:

1. Test Steeper Elevation Changes

   Elevation changes directly influence energy consumption in EVs due to the increased demand during uphill climbs and the benefits of regenerative braking downhill. Future studies should test routes with steep inclines and declines over shorter distances, such as those found in mountainous regions. This would provide a more accurate understanding of how elevation variability impacts energy consumption, thermal management, and overall battery efficiency. Insights from these tests could also inform SOC planning and strategic charging station placement in areas with challenging terrains.

2. Test Larger Temperature Differences

   The HVAC (heating, ventilation, and air conditioning) system is a significant energy drain in extreme temperatures, especially in EVs. Testing under substantial temperature differences, such as below-freezing conditions or scorching heat, would stress the HVAC system and reveal how it affects battery performance. These tests could highlight the importance of features like preconditioning and thermal management systems in mitigating energy loss. Moreover, the data collected could inform strategies for improving range and energy efficiency in climates with extreme weather conditions.

3. Test Starting SOC

   Battery consumption in EVs follows a nonlinear curve, with performance differing based on the starting state of charge (SOC). By varying the starting SOC in test scenarios—such as high (90%), medium (50-60%), and low (20-30%)—we can better understand the efficiency and performance trade-offs at different charge levels. These insights could guide optimal charging practices, particularly for long-distance or energy-intensive trips, and improve range predictions under different SOC conditions.

By incorporating these recommendations, future studies can better understand how EVs perform under diverse and challenging conditions, ultimately improving vehicle design, energy management systems, and user experience.

Important Links

InceptEV PJTL Drive - Google Drive for Presentations, Data, Documented Progress and etc.

Github Repo - Github Repo with changes for the project