Dynamically Modeling Bus Routes for Informed Transit and Electric Vehicle Development

Sebastian Sachet Peck

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Masters of Chemical Engineering

University of Washington

2025

Committee:

Daniel T. Schwartz, Chair

David A. C. Beck

Program Authorized to Offer Degree:

Department of Chemical Engineering

University of Washington

Abstract

Dynamically Modeling Bus Routes for Informed Transit and Electric Vehicle Development

Sebastian Sachet Peck

Chair of the Supervisory Committee:

Daniel T. Schwartz

Department of Chemical Engineering

Metropolitan Transit Agencies incorporating Battery-Electric Buses face significant hurdles when developing these programs due to the uncertainty and variability inherent to mass transit and the substantial energy requirements involved. To expedite the process and support transit electrification, I have developed an open-source python-based modeling tool called reRoute_Dynamics that couples Longitudinal Dynamics Models and GIS data to generate realistic energy storage load profiles. I validated this model through comparing collected GPS data to aggregated Monte Carlo Simulations for several transit routes in King County, Washington. I then demonstrated that this tool is capable of providing information transit agencies pursue in their pilot programs - how vehicle energy mileage depends on different aspects of a trip, route, and geography, as well as provide insights into the cell-level behavior of the energy storage system. Geography, vehicle mass, and driving behavior are all significant driving factors that reduce vehicle mileage from a manufacturer's quote, conclusions that are validated by existing literature. This work provides a tool and framework for electric vehicle transit research, and will enable more transit agencies to pursue electric vehicle pilot programs at lower cost.

Page Intentionally Left Blank

Table Of Contents

Page Intentionally Left Blank

List of Figures

Acknowledgments

I would not have made it to this point without the support of countless people, and I would like to take a moment to provide some specific thanks to everyone who has helped me on my academic journey thus far.

To Dan Schwartz, thank you so much for taking me on as an undergraduate and enabling me to set off on a course I could have never predicted for myself at the time. Your mentorship, advisement, and support has been incalculably valuable over these past 5 years, and I'm extremely grateful for it.

To Dave Beck – your enthusiasm and passion seem to know no bounds and have been a constant inspiration to me over the last two years. Thank you for always being willing to provide advice, support, and insight, and for all the effort you put into helping me grow as a student and data scientist.

To Erica Eggleton, thank you for welcoming me into the world of battery and data science research. Without your hard work laying the foundation both for this project and for my own development as a researcher, this would not have been possible.

To the folks in my lab, in Benson 237, and beyond – Rose, Rebecca, Yuefan, Kristin, Leo, Tata, Lauren, Elena, Maddie, Chris, Julia– Thank you all for the conversations, proofreads, wisdom, and for simply just being there.

Julie, Alex, David, Cole, Stu, Eric, Dave D., Jaque, and everyone in the faculty and staff at UW ChemE - thank you so much for supporting, teaching, and mentoring me over the past 5 years.

To the folks in my Graduate cohort, whom all worked incredibly hard to get through our classes and theses together.

Thanks To Nick, Jay, Eric, and Yujia – for simultaneously inspiring me to push beyond the limits of what I thought myself capable of, and for simply just being your amazing, dedicated, friendly, and supportive selves.

To all the friends (a list, as grateful as I am for all of you, that would be too long for this section) whom have helped me ride the highs and lows of life in countless and immeasurable ways – the world we live in would be a much less joyful place without you all in it.

And finally, to my family. The love and support you've given me throughout my life is second to none in the world. I wouldn't be here without you. August, Dad, Mom, Oma & Opa, Grandma, and to all the rest, I love you, and am forever thankful for you each being part of my life.

Chapter 1 - Introduction

1.1: Transit Electrification in the US

Since 1990, greenhouse gas emissions in the US have been dominated by three sectors: Electric Power, Industry, and Transportation [1] While the widespread commercialization of electric vehicles (EVs) began to reduce the volume of greenhouse gas emissions in 2007, the emission volume has not substantially changed in the years since. This proliferation of electrification technology is only just beginning to spread into the heavier-duty applications, such as trucking and mass transit, spurred on by both federal and state environmental and economic incentives. Slowly, these high-emission subsets of transportation are becoming more economically viable to electrify [2], [3], [4]. King County Metro (KCM) was among the first public organizations to begin developing its fully-electric infrastructure, seeking to be carbon-neutral across all transit modes by 2035 [5] – but groups across the country have followed suit [6], and are beginning pilot programs with similar targets.

1.2: Energy Use and Transit Pilots

Traditionally, electric vehicles have been evaluated for mileage according to a set of standards, whether it be through the EPA's standard protocol [7], or the Federal Transit Administration's "Altoona" protocol [8]. These standard procedures use dynamometers and have the vehicle follow a standard set of drive cycles to run down the energy storage system (ESS) to get a general idea for how far the vehicle is capable of travelling. While these methods account for a variety of velocity-based conditions with a reasonable amount of overhead for auxiliary systems, they can be considered general. Made explicit within the "Altoona" test protocol is the fact that "The result of this test will not represent the actual mileage" [8]. Nonetheless, manufacturers such as New Flyer use the metrics produced by these tests in their marketing materials for transit organizations looking to electrify. This poses a problem for these transit organizations – a lack of clear information on whether this vehicle will serve their purpose will be a significant hurdle in getting funding and getting any EV project off the ground. This lack of clear insight into the energy requirements of these vehicles also impacts the infrastructure development. Without knowing how much energy the vehicles will need to get to their destination and back successfully, there's no telling what extra strain it might put on the grid – let alone what the potential downtime and service costs are if a vehicle fails. When potentially spending $1 Billion on infrastructure alone [9], transit organizations want to be certain their investment is sound. These are the kind of questions pilot programs seek to solve [10]. Unfortunately, pilots are difficult, expensive, and time-intensive.

KCM began their pilot program in earnest in 2016, and even then only began with three vehicles for a single route [10]. As of 2024, this number of vehicles has expanded to 48 [11]. However, with 147 different transit routes in king county each with several permutations based on time, day, and season, this program cannot encompass the true scale in variability of these routes. Compounding the uncertainty generated by this is the fact that ridership, traffic, and weather conditions will all also vary significantly. When pivoting to novel transit that has fewer established maintenance pathways and proven use cases, it is important to consider as wide a breadth of potential circumstances as possible – something that a pilot program can only do at great financial and time cost. At a time where implementing these changes is more critical than ever, these barriers need to be reduced wherever possible.

1.3: Modeling Electric Vehicles

This issue of EV energy use in transit vehicles has been a routine subject of study. To understand the variability in energy requirements and help supplement BEB implementation, previous work [12], [13], [14], [15] has demonstrated the efficacy of using a Longitudinal Dynamics Model to determine EV Energy requirements. A Longitudinal Dynamic Model (LDM) is a simplified physics model of a vehicle that relies on a balance of forces to calculate the energy needed to move that vehicle.

$$F_{traction} = F_{acceleration} + F_{gravitational} + F_{air}$$

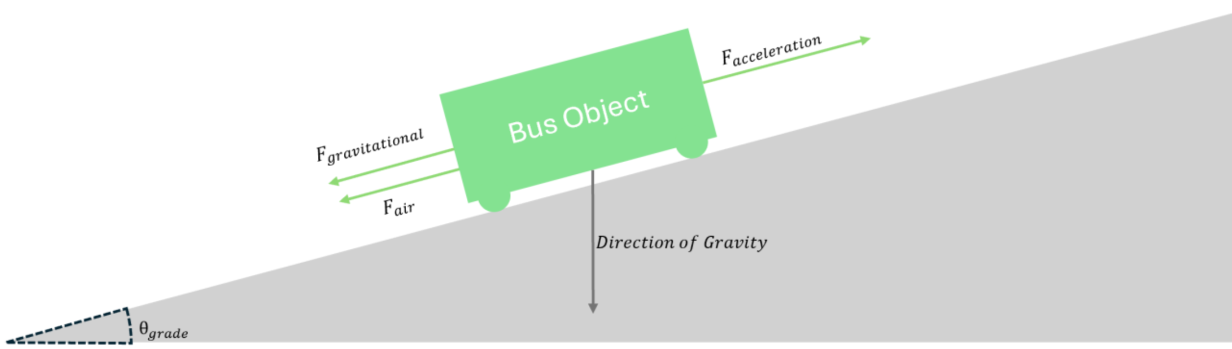


***Figure 1.1**: Diagram of Longitudinal Dynamics Model of a Bus Object on a slope.*

The force of traction, $F_{traction}$, is the total amount of force that the vehicle needs to output at the wheels to meet the external forces $F_{gravitational}(F_g)$ and $F_{air}(F_w)$, and achieve a net force of acceleration, $F_{acceleration}(F_a)$. While methodology for determining these forces vary based on drivetrain mechanisms, efficiencies, and more intricate details, the general force balance principle remains the same. The formulas used to calculate these forces are typically simple. F_w, as an example, is calculated as follows:

$$F_w = \frac{C_d * A_f * \rho}{2} * (v_{bus} - v_w)^2 \text{ (eqn. 1.1)}$$

C_d and A_f are the Coefficient of Drag and Frontal Area of the bus, ρ is the density of air, and v_w and v_{bus} are the velocity of the wind and bus, respectively. F_g is similarly defined based on simple physics:

$$F_g = (\sin(\theta_g) + \cos(\theta_g) * C_f) * g * m \text{ (eqn. 1.2)}$$

Where θ_g is the road angle, determined by grade of the road, C_f is the coefficient of friction between the wheel and the road, g is gravity and m is total vehicle mass. F_a is more difficult and situational to define. Likewise, the actual power used depends on the drivetrain model, which vary.

These studies have found similar conclusions in their search for what impacts the energy efficiency of EV Busses. Road grade, something that the conventional mileage tests do not directly account for, is a significant driving factor in the variability of modeled capacity. Likewise, variable mass has been demonstrated to have an effect, which is yet another variable unaccounted for by the conventional EPA and Altoona test. The case study using Malaysia's transit routes [15] used GIS data to inform their modeling, and other studies have used existing drive cycles and custom tools in Simulink & Python to study this behavior. The capability to reasonably model these routes in specific cases paves the road for easily-accessible tools for transit organizations to do their own studies – as there is no 'one-size-fits-all' approach when it comes to modeling for these EV's.

1.4: Software to Fill the Gaps of EV Transit Knowledge

As Battery-Electric Vehicles (BEVs) are the most intense use cases for batteries, knowing what will impact their use and aging processes is of the utmost importance and serves multiple purposes. First, it can help support the development of modern EV transit infrastructure through providing insights into energy consumption trends. With more ways to assess the impact of installing new technologies and how effective they will be, we can lower the barrier for any transit organization looking to electrify. Second, the modern Lithium-ion battery is both resource and environmentally intensive to produce [16]. It is in the best interest of sustainability, economic, and humanitarian goals that we use these resources to their utmost once mined and manufactured.

Because prior work has demonstrated the efficacy of modeling these drive cycles, the impacts they have, and the factors affecting them, the next step is to model EV drive cycles at a larger scale. Being able to model any route, anywhere, with a reasonable degree of accuracy is a path to alleviate the existing struggles of BEB use and transit planning. Coupling LDMs, open-source data, and user-friendly software tools has the potential to expand upon specific use cases for transit research. Additionally, modeling BEB energy use can create inroads for high-throughput EV battery aging models and studies.

Chapter 2 – Python-Based Modeling of EV Drive Cycles

2.1: Introduction

To enable more easily accessible route modeling and energy requirement projections, Dr. Erica Eggleton began developing a python software package - 'Route_Dynamics' [17]. Route_Dynamics was intended to be an open-source tool that coupled a Longitudinal Dynamics Model and geospatial data to generate realistic drive cycles and their corresponding ESS load. This work was presented on at ECS Dallas 2019, ECS Atlanta 2019, and ECS Vancouver 2022 [18], [19], [20] and the tool itself remains available on GitHub. Route_Dynamics laid a strong foundation of ideas for creating a widely-applicable route modeling tool. However, the toolkit itself needed more thorough validation, as well as left plenty of room for further development. To this end, I began from the original concept, re-wrote and re-organized everything from scratch, and created "reRoute_Dynamics".

reRoute_Dynamics couples geospatial data with a LDM, object-oriented physics, and driving logic to enable the creation of highly customizable BEV Models. It implements careful geospatial coordinate handling, an expansive library of modeling parameters, and the flexibility to expand each object-oriented module as the user desires. This restructuring has opened the possibility for the tool to be used in a wider variety of applications the original was not designed for, including Monte Carlo simulations and sensitivity analysis.

2.2: Modeling

reRoute_Dynamics operates based on the core physics of a LDM, using route and bus conditions at iterative geospatial points to determine the operating energy. The generalized flow to create a model is illustrated below.

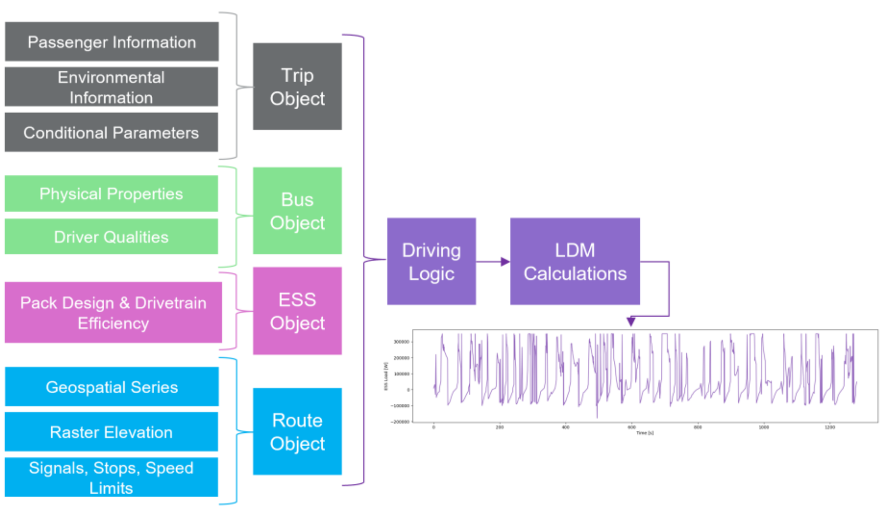


Figure 2 . 1: *Generalized flow diagram of reRoute_Dynamics' process.*
By taking parameters describing a trip, a bus, an ESS, and a route, reRoute_Dynamics can have the described vehicle perform according to driving logic, resulting in load, velocity, and acceleration profiles for that trip, among other possibilities.

The process begins with object generation. There are 4 programmatic objects used in the generation of a profile – the Bus, the Trip, the ESS, and the Route.

2.2.1: Bus Object

The bus object stores a variety of parameters used in the modeling of a bus and the physics involved.

m: Mass – the empty mass of the entire vehicle.
w_f: Frontal Width – the width of the front of the vehicle.
h_f: Frontal Height – the height of the front of the vehicle.
C_d: Drag Coefficient – a coefficient used to quantify the drag of the vehicle.
C_f: Friction Coefficient – a coefficient used to quantify the friction and rolling resistance of the wheels in the total rolling resistance force.
a_{br}: Braking Acceleration – maximum amount of acceleration the bus will experience when braking on a flat road.
$a_{profile}$: Acceleration Profile – The acceleration profile the bus will adhere to a via speed governor.
a_{max}: The final acceleration that the bus will reach after the profile is completed.
f_{br}: Braking Factor – a factor representing what fraction of the braking acceleration is used initially. In other words, how aggressively the driver brakes.
f_a: Acceleration factor – a factor representing how willing a driver is to exceed the speed limit as a set point
f_i: Inertial Factor – a factor representing the amount of resistance the bus has to acceleration.
dx_{max}: The ideal maximum distance it takes for the bus to fully brake.
dt_{max}: The timestep resolution of the final acceleration.
P_{max}: The maximum power able to be output by the electric motors in the vehicle.

The most complicated parameter among these is the Acceleration Profile, $a_{profile}$. Most modern transit vehicles like those produced by New Flyer have their speed and acceleration behavior controlled by a speed governor [21]. This governor regulates the moment-to-moment acceleration of the bus to ensure a safe and sustainable acceleration for the riders. Because these can't be easily defined as a single acceleration value, custom acceleration vs. time data can be provided for the bus to try to adhere to when accelerating.

2.2.2: Trip Object

Each time a bus travels a route, there will be a variety of different external conditions. These external conditions that the model accounts for are among the most variable, and are accounted for within the Trip Object.

m_{pass}: Mass per Passenger
MOE: Margin of error on the speed limit – how far above or below the speed limit the driver is willing to let the vehicle speed vary.
$\%_{signal}$: Chance that a bus will have to stop at any given signal light
t_{sig}: Rest time per signal
t_{stop}: Rest time, per passenger, per stop.
t_{end}: Rest time at the start or end of a route.
ρ_{air}: Density of air
θ_{wind}: Wind heading
v_{wind}: Wind speed
n_{riders}: The mean number of unique riders for a given route.
m_{stop}: Margin for the velocity to be considered fully stopped.
$\%_{traffic}$: How slowed the bus is due to traffic – this artificially reduces the speed limit.

Because the available data about ridership is not always comprehensive and there is inherent variability in how ridership changes at each stop, the mean ridership for a route in reRoute_Dynamics is used to generate a ridership profile. This profile is created by using the number of stops in a route and randomly distributing the mean unique riders across all stops. Subsequently, a random number of the 'current riders' at each stop are removed up until the terminus of the route.

Similarly, since there's no certainty about when a bus will hit a stop signal or for how long it will rest, the values of $\%_{signal}$ and t_{sig} are presumed to be 'typical' constants for a given trip. Wind heading and wind speed are unlikely to vary throughout a trip, and so are also assumed constants. Traffic, being another unknown that highly depends on time of day and location, is a constant for a given trip being simulated.

2.2.3: ESS Object

In reRoute_Dynamics's current design, the ESS of the bus is primarily used in the final step of translating the required motor power for a given route to the demand on the ESS. It has no impact on the actual driving performance of the vehicle itself, as the maximum motor power is stored in the bus object.

η_m: Motor Efficiency
η_i: Inverter Efficiency
η_a: Auxiliary Efficiency
η_r: Regenerative Braking Efficiency
P_{aux}: Auxiliary Load
P_{regen}: Maximum Regenerative Load
V_{cell}: Cell Open Circuit Voltage
R_{cell}: Cell Internal Resistance
SP_{mod}: Battery cell series-parallel configuration
SP_{bus}: Battery Module series-parallel configuration
Q_{cell}: Cell charge capacity

To calculate the load the ESS is experiencing, the following calculations are used:

$$P_{ess} = \begin{cases} P_{bus} \geq 0: & \frac{P_{bus}}{\eta_m * \eta_i} + \frac{P_{aux}}{\eta_a} \\ P_{bus} * \eta_r * \eta_m > P_{regen}: & P_{bus} * \eta_r * \eta_m + \frac{P_{aux}}{\eta_a} \\ otherwise: & P_{regen} + \frac{P_{aux}}{\eta_a} \end{cases} \quad \text{(eqn. 2.1)}$$

This set of formulae does not wholly encompass the mechanistic losses in an electric or hybrid drivetrain. This is a preliminary model that is designed to align with literature [22] , but is modular by nature and allows for future adaptations and more specific adjustments depending on user desire. The ESS object can also be used to calculate a simple estimate of cell voltage, using an R_{int} model, should the user desire. This is done by scaling down the power using the model and pack design parameters.

$$V_{cell} = V_{cell}^0 - \frac{R_{cell} * \left(\frac{P_{ESS}}{R_{bus}}\right)^{\frac{1}{2}}}{SP_{bus}[paralell] * SP_{mod}[paralell]} \quad \text{(eqn. 2.2)}$$

While this model is simple when compared to using a version of the Butler-Volmer equation, the ESS object is written and designed in such a way that future users are able to extend the class and modify it to suit their needs.

2.2.4: Route Object

Routes are generated using a series of geospatial points and elevations that correspond to GPS positions on a given route. These geospatial points are used to determine heading and distance travelled, and can be used to query elevation, which in turn is used to determine the slope. The distances between each point on the route are calculated using the Vincenty Formula, as provided by the geopy package's geodesic() method. Heading is calculated using the following formulae:

$$x = cos(\lambda_2) * sin(\varphi_2 - \varphi_1) \text{ (eqn. 2.3.1)}$$

$$y = cos(\lambda_1) \cdot sin(\lambda_2) - sin(\lambda_1) \cdot cos(\lambda_2) \cdot cos(\varphi_2 - \varphi_1) \text{ (eqn. 2.3.2)}$$

$$\theta = arctan(\frac{x}{y}) \text{ (eqn. 2.3.3)}$$

λ and φ are the respective latitude and longitude for point 1 or 2. While this heading can hypothetically be used for things like checking against signage, for our purposes it will be only be used in the calculation of external wind force. The slope grade calculation is comparatively simple:

$$\%_g = \frac{z_2 - z_1}{dx} * 100 \text{ (eqn. 2.4)}$$

Where z is the elevation of two points and dx is the geodesic distance between those two points. More importantly, the grade angle is determined by the following:

$$\theta_g = \arctan \frac{\%_g}{100} \text{ (eqn. 2.5)}$$

It is important to note that geospatial data and elevation data can be particularly noisy, especially depending on the type and source. As such, there are also included filtering methods, including point interpolation and grade/elevation smoothing. The gps interpolation is performed by interspersing as many evenly-spaced points between two others with a gap that exceeds a certain size, dx_{gap}. For example, if the dx_{gap} parameter is 10 meters, a gap of 30 meters will have 2 points at even spacing between both ends. The elevation and grade smoothing method uses a Savisky-Golay filter to reduce the noisy data to a series of polynomials that are customizable in degree and points encompassed.

Beyond the path and corresponding elevations, the speed limits, stops, and signals are also important to how a route performs. If the route already has the corresponding data for each point, it can be easily added. reRoute_Dynamics includes a handful of tools that can be used to extract this information from common formats of raster files or GIS shapefiles. However, in the event they cannot, the default speed is set to 25 mph, with 10 evenly spaced stops, 8 evenly spaced signals, and 3 evenly spaced stop signs. This, again, is entirely customizable.

2.2.5: Driving Logic

When combined, this aggregation of Bus, Trip, ESS, and Route can be cumulatively passed through a set of driving logic that helps determine what physics calculations to perform. The bus object is progressively stepped through each point and the corresponding conditions. Those conditions and the trip variables inform the 'driving decisions':

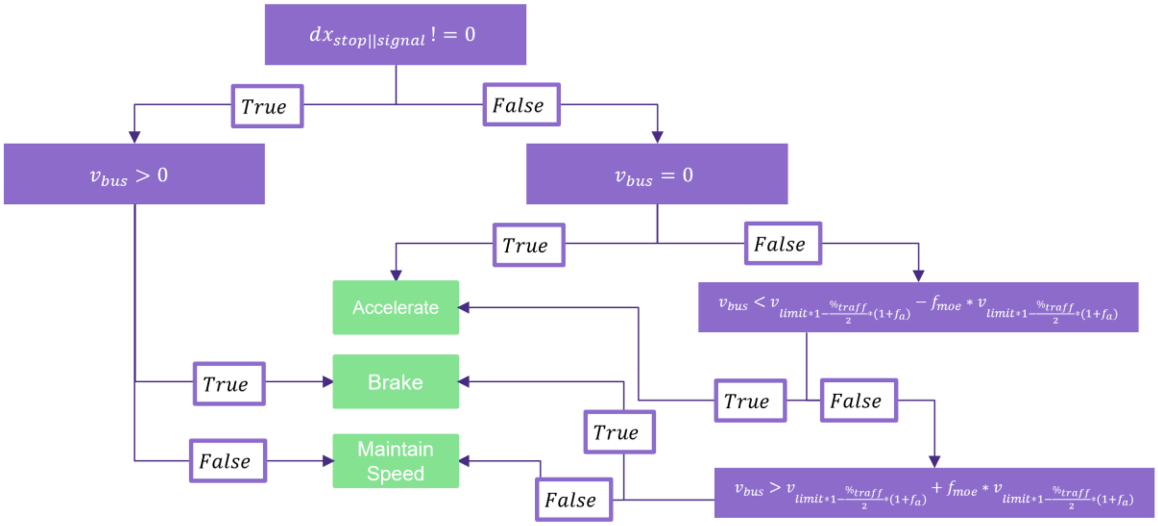


Figure 2 . 2: *Flow Diagram of Driving Logic*

The two more complicated steps below the $v_{bus} = 0$ are for determining what is loosely termed 'speed limit'. The first formula is to check if the vehicle is over the set point for the current location, while the second is to see if it's below the set point. Both are illustrated in eqn. 2.6.2.

$$v_{set} = v_{limit} * \left(1 - \frac{\%traffic}{2}\right) * (1 + f_a) \text{ (eqn. 2.6.1)}$$

$$v_{bus} \overset{>}{\underset{<}{=}} v_{set} \pm f_{moe} * v_{set} \text{ (eqn. 2.6.2)}$$

This set point is determined by the flow of traffic for the trip, as well as the acceleration factor – the willingness for the driver to exceed the speed limit. The margin of error, f_{moe}, is used to help prevent constant oscillation between accelerating and braking, as it is not feasible for the bus velocity to ever exactly match the target set point.

When the vehicle comes to a stop, a period of rest time is added separately to the three functional bus states, as determined by the stop type. If the stop is due to a signal light, the rest will use the concrete value of t_{sig}, whereas if the stop is a bus stop, the rest will use t_{stop}, multiplied by the total change in riders. There are also buffer periods of time, t_{end}, that are the times between when the bus starts or stops, and when the engine is turned on or off.

2.2.6: LDM Physics Calculations

The flow diagram in Figure 2.2 terminates in one of three conditions for each point it's passed – Accelerating, Braking, or Maintaining Speed. Each have slight differences in how they're approached, but all operate on the same fundamental principles of the Longitudinal Dynamic Model outlined in Chapter 1, section 3, which we modify a bit to account for relative wind heading.

$$\theta_{rw} = (\theta_{bus} - 180) - (\theta_{wind} - 180) \text{ (eqn. 2.7)}$$

$$F_w = \frac{(C_d * w_f * h_f * \rho)}{2} * (v_{bus} - v_w * \cos(\theta_{rw}))^2 \text{ (eqn. 1.1.1)}$$

When accelerating, braking, or maintaining speed, each option operates based primarily on the kinematic equations. The most straightforward way to demonstrate this is with braking, particularly the braking distance calculation.

2.2.6.1: Braking

a_{decell} is calculated first. By weighting the braking acceleration a_{br} with the inertial factor f_i and the braking factor f_{br}, it accounts for the momentum and driver braking aggression. This is then added to the total external force the bus is experiencing, divided by the total mass of the bus. This results in the cumulative rate of deceleration the bus is experiencing.

$$a_{decell} = a_{br} * f_{br} * f_i + \frac{F_g + F_w}{m_{total}} \text{ (eqn. 2.8)}$$

Then, in equation 2.9, the braking distance is calculated using the aforementioned kinematic equations ($v_f^2 = v_{bus}^2 + 2 * a_{decell} * dx_{braking,0}$) and a_{decell} to determine how far the vehicle will have to travel to come to a complete stop.

$$dx_{braking,0} = \frac{v_{bus}^2}{2*(a_{decell})} \text{ (eqn. 2.9)}$$

Within the parameters of the bus, there is a set point dx_{max}. If the braking distance of the vehicle is more than that set point at the current f_{br}, f_{br} will be increased until either it reaches a value of 1 (fully braking as hard as possible) or $dx_{braking,0} = dx_{max}$.

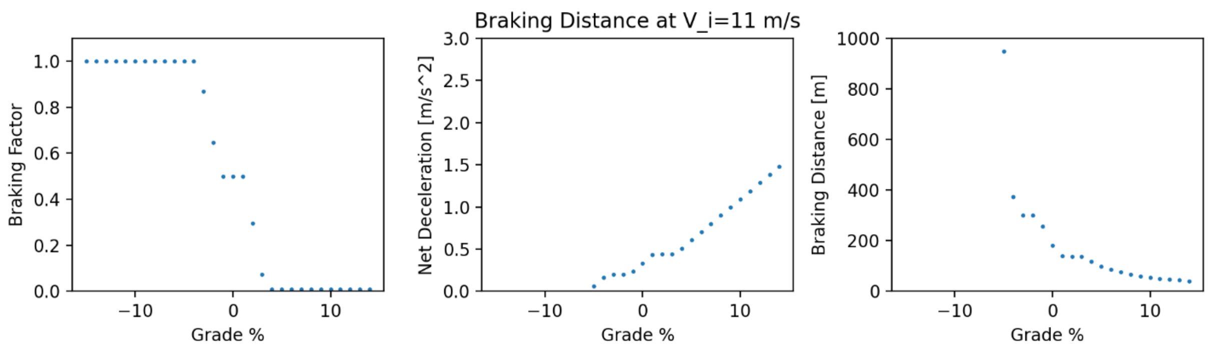


Figure 2 . 3*: Vehicle braking behavior*
Initial velocity of 11 m/s.
Left: Braking factor set point as a function of road grade,
Center: Net Deceleration as function of road grade,
Right: Braking distance as a function of road grade.

We can see this set point behavior in Figure 2.3. At the steeper downhills, the vehicle is unable to brake enough to slow down, despite the braking factor being raised to the maximum. At around a grade of -7.5, we can start to see the bus is able to fully brake, but this still exceeds the braking distance set point. We see the dx_{max} set point of 350 meters when the grade is slightly downhill. The initial set point of f_{br} was .5, and so we can see a plateau of that up until the slope becomes steep enough that the vehicle doesn't need to brake to slow.

When slowing the vehicle, the braking calculations use the same a_{decell} formula, and then calculate velocity and time change, as well as power use.

$$v_f = \sqrt{-a_{decell} * 2 * dx_{braking} + (v_{bus}^2)} \quad \text{(eqn. 2.10)}$$

$$dt = \frac{(v_{bus} - v_f)}{a_{decell}} \quad \text{(eqn. 2.11)}$$

$$P = \frac{m_{total} * -(a_{br} * f_i * f_{br}) * dx_{braking}}{dt} \quad \text{(eqn. 2.12)}$$

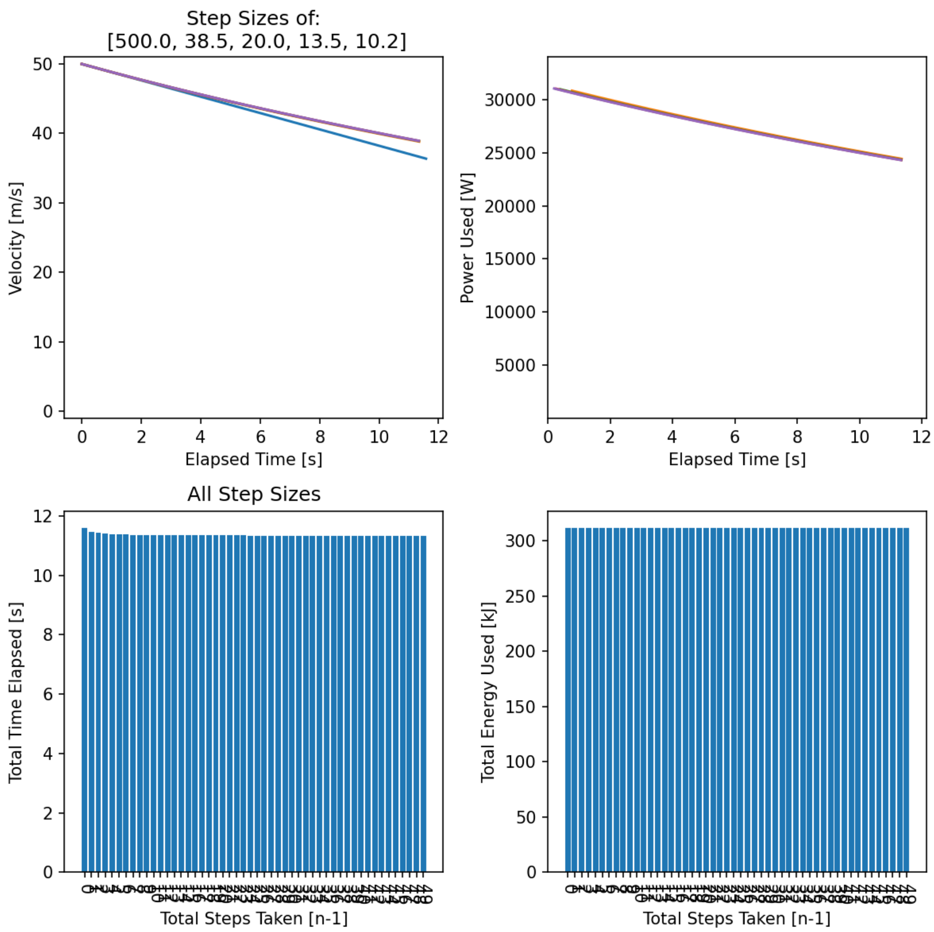


Figure 2 . 4: *How braking distance step size impacts braking performance.*
Top Left: Velocity over time with 5 different step sizes, braking over a distance of 500 meters.
Top Right: Power used 5 different step sizes, braking over a distance of 500 meters.
Bottom Left: Elapsed time as a function of individual braking steps.
Bottom Right: Total energy used as a function of total steps taken.

When braking, the vehicle is designed to brake for a set 'distance step'. Given there is variation of distance between points in a route, it is important to know that regardless of step size, the resulting conditions will be identical. As can be seen in figure 2.4, save for the step of 500 meters being marginally different, the velocity, elapsed time, and energy use are functionally identical regardless of how many subdivisions of 500 meters the vehicle brakes over. To quantify this:

	Velocity [m/s]	Time [s]	Energy [kJ]
Mean	38.7898	11.350	311.1269
Standard Deviation	0.39784	.040828	5.96731*10^-14

Table 2.1: Variation in velocity, time, and energy while braking over 500 meters with step sizes ranging from 500 meters to 10 meters.

2.2.6.2: Maintaining Speed

Maintaining velocity is again relatively straightforward. To maintain speed, the bus must balance the $F_{traction}$ and the external force such that $F_{acceleration}$ is zero. So, if the external forces would slow the vehicle down, the vehicle must accelerate just enough so as to counteract them. To do this, first the power needed to counteract the external forces is calculated, as well as the maximum possible braking power.

$$P_{current} = m_{total} * \frac{F_g+F_w}{m} * v_{bus} \text{ (eqn. 2.13)}$$

$$P_{br\_max} = -m_{total} * v_{bus} * (a_{br} * f_i) \text{ (eqn. 2.14)}$$

If $P_{current}$ exceeds the maximum motor power P_{max}, the bus will end up slowing with the new velocity v_f.

$$v_f = \sqrt{v_{bus}^2 - 2 * dx_{maintain} * \frac{P_{current}-P_{max}}{m_{total}*v_{bus}}} \text{ (eqn. 2.15.1)}$$

The power difference here is being used to determine the level of deceleration of the vehicle. In the circumstance where there isn't enough braking power P_{br_max} to overcome the $P_{current}$, the formula changes slightly. The time change, dt, remains calculated according to equation 2.11.

$$v_f = \sqrt{v_{bus}^2 + 2 * dx_{maintain} * \frac{P_{current}-P_{max}}{m_{total}*v_{bus}}} \text{ (eqn. 2.15.2)}$$

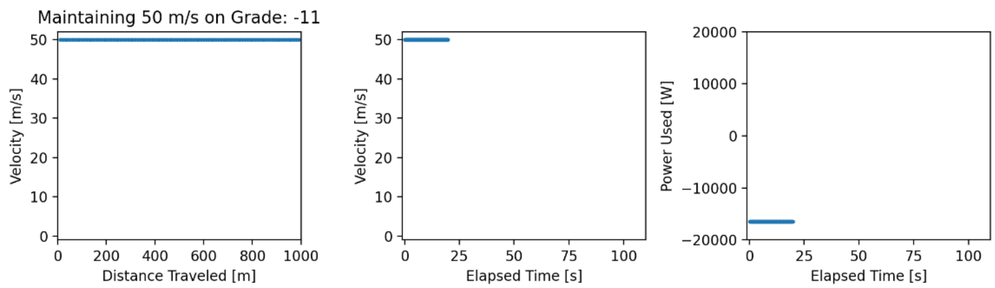


Figure 2 . 5*: Steep Downhill Maintenance Behavior*
Initial speed of 50 m/s on an extremely steep downhill over a distance of 1 kilometer
Left: Velocity vs. Distance traveled
Center: Velocity vs. Elapsed Time
Right: Power used vs. Elapsed time

While trying to maintain an unreasonably high speed on an unreasonably steep incline, the vehicle is perfectly capable, demonstrating a sustained velocity and a reasonable amount of power used to brake the vehicle.

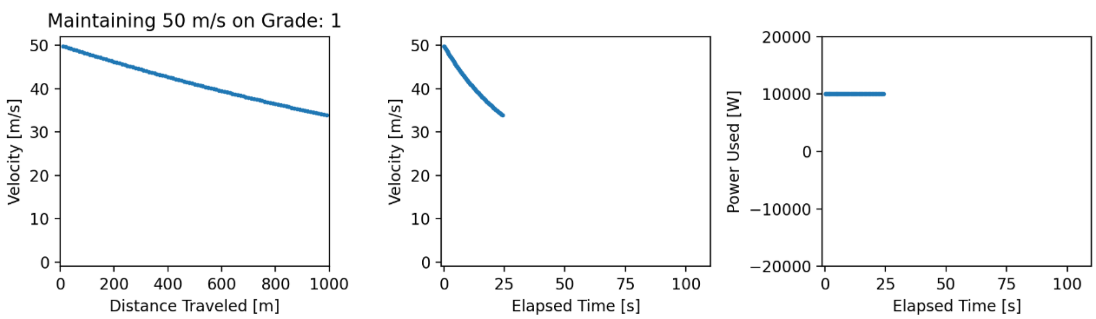


Figure 2 . 6*: Mild Incline Maintenance Behavior*
Initial speed of 50 m/s on a very mild incline for 1 kilometer.
Left: Velocity vs. Distance traveled
Center: Velocity vs. Elapsed Time
Right: Power used vs. Elapsed time.

Comparing Figure 2.5 to Figure 2.6, we can see that the vehicle is no longer able to maintain it's target velocity and is slowing down due to the external forces. As the motor is progressively more able to meet the external demands, the velocity drop-off declines – but the test vehicle's motor power remains constant at it's simulated set point of 10,000 Watts. Further figures can be found in Section 2.6.1.

2.2.6.3: Accelerating

Acceleration is the most complex of the three states. Because of the aforementioned speed governors built into transit buses, the velocity is limited to adhere closely to a set profile of acceleration to reduce potential impact on passengers. These 'acceleration profiles' are used by reRoute_Dynamics to determine how a bus should be accelerating when at a given velocity. At any point where the bus must accelerate, the initial velocity v_{bus} is checked against an integrated velocity profile to find the closest index. The profile's achievable acceleration is adjusted such that the power at any given step would be at or under P_{max}. Then, the profile is then stepped through, using the corresponding internal accelerations until the second integration of the profile (the travel distance) meets the $dx_{acceleration}$ that's passed. The core of the calculations adhere to the same equations used for braking and maintaining velocity, with some adjustments for edge cases. In the event the remaining travel distance to meet $dx_{acceleration}$ is lower than what using the next step of the profile would yield, dt for that step is back-calculated.

$$dt_{step} = \frac{-v_i + \left(v_i^2 + 2 * (dx_{remaining} * a_{profile\ step})\right)^{\frac{1}{2}}}{a_{profile\ step}} \quad \text{(eqn. 2.16)}$$

This is then used to determine the final velocity for the final step, as well as the energy consumed. Because acceleration can have multiple steps, power is instead calculated at the end. The cumulative energy for each step in the acceleration is averaged across the total time spent accelerating. Despite this, power is still limited by P_{max} – the output just won't be nearly as accurate to a minutia of time.

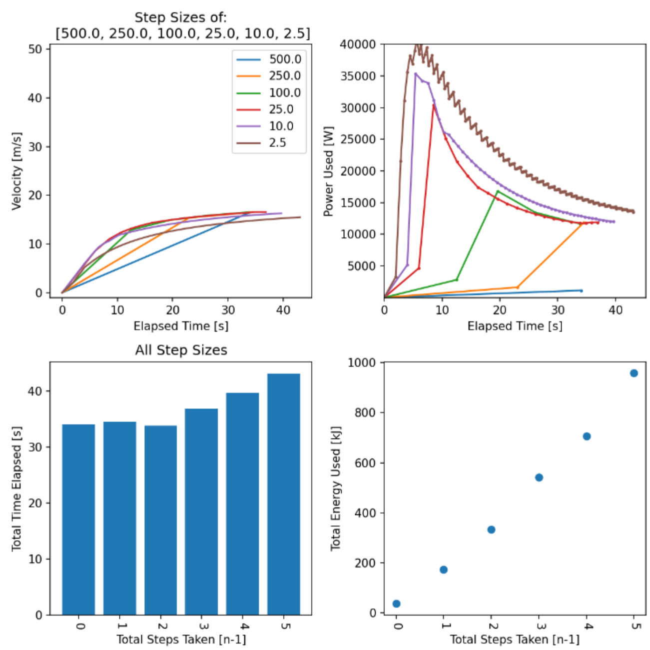


Figure 2 . *7: Accelerating a vehicle using different dx step sizes.*
Distance of 500 meters.
Top Left: Velocity vs. Elapsed time for different step sizes.
Top Right: Power used vs. Elapsed Time for different step sizes.
Bottom Left: Total calculated elapsed time vs. Steps taken.
Bottom Right: Total calculated energy used vs. Steps taken.

From Figure 2.7, it is clear there's a difference in the outcome depending on the passed distance the vehicle is intended to travel. The closer the dx is to the resolution of the profile, the more accurate the resulting time and energy become. This is because the load is not averaged across as much time or distance as it would be at higher step sizes.

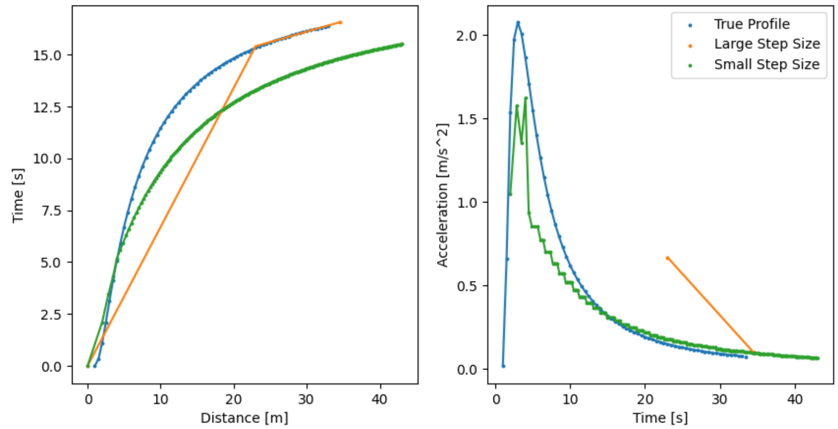


Figure 2 . 8 *Acceleration profile comparison for step size.*
Left: Time Vs. Distance comparison. Right: Time vs. Acceleration comparison.

As we can continue to see in figure 2.8, the smaller the step size, the more accurately the vehicle's acceleration will fit the profile. Contrasting this, the final elapsed time would appear more accurate for a larger step size – but this is deceptive. With each acceleration step, the external conditions are updated. The smaller step size, the more frequent the conditional updates. Thus, the external forces like wind drag and road frictional losses are more accurate to reality when the step size is smaller.

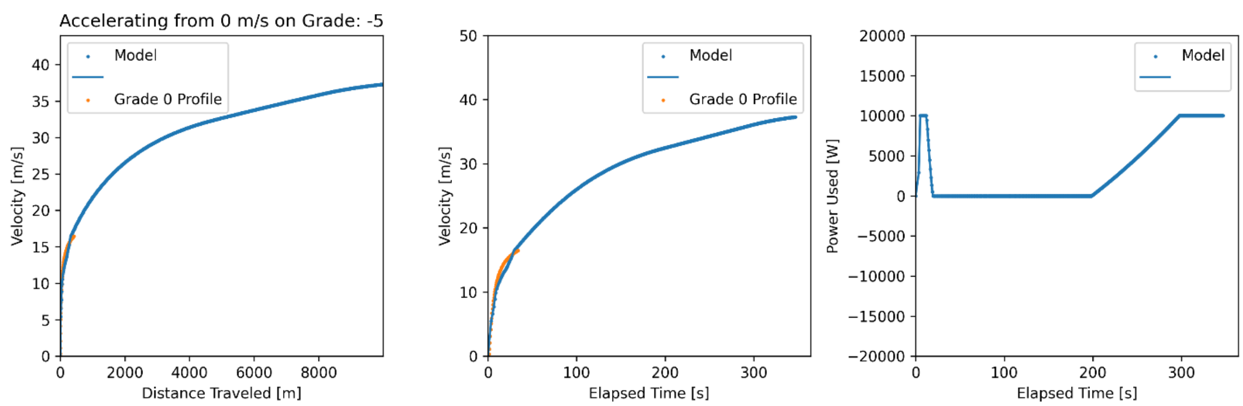


Figure 2 . 9*: Accelerating from 0 m/s on a downhill grade.*
Left: Velocity vs. Distance.
Center: Velocity Vs. Time.
Right: Power vs. Time.

Figure 2.9 further illustrates this with a small step size – on a downhill grade, the bus is almost perfectly able to match the theoretical acceleration profile, as those external conditions are being met effectively by the motor – which, after the initial push, is not needed to accelerate up until the wind drag becomes too much.

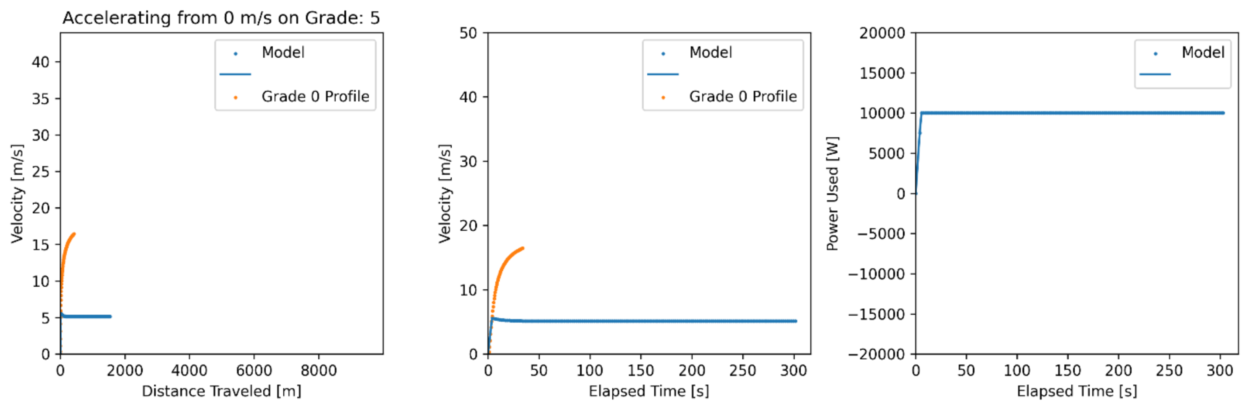


***Figure 2 . 10**: Accelerating on an uphill grade.*
Initial velocity of 0 m/s
Left: Velocity vs. Distance.
Center: Velocity Vs. Time.
Right: Power vs. Time.

In this example, the maximum motor power is set to be 10,000 watts. We can see that at an incline, the vehicle is not able to meet the acceleration conditions as outlined by the acceleration profile. Further illustrations of this behavior can be found in section 2.6.2.

2.3: Model Validation

2.3.1: GPS Data Collection

To validate the model, I collected real-world GPS data from a selection of routes in King County. While King County provides resources to track its vehicles, it proved difficult to determine persistently identify vehicles with a clear level of confidence. Of the 147 different routes in King County, I selected routes 45, 70, 8, and 49 to ride 3 separate times each. Route 45 is long and suburban, covering a wide breadth of ridership conditions – many at the beginning, few at the end. Route 70 is on the shorter side with less hills, and travels through downtown Seattle on some less-trafficked roads. Route 8 is yet another long route – travelling from Mt. Baker to Seattle Center through some of the most extreme uphills and downhills in Seattle. Finally, route 49 drives through some of the heaviest traffic in downtown Seattle and Capitol Hill, and includes some steep inclines and long declines.

In order to track the locations of these vehicles as I rode them, I used a MTK3333 GPS module [23] attached to a Raspberry Pi Zero 2 W to store the data. As a backup, I used the GPS contained within an IPhone 12 Mini, logged via the Phyphox application [24]. The positional accuracy for both of these tools is approximately between 3 to 5 meters, varying depending on satellite tracking [25]. For each trip, as seating allowed, I sat as close to the front of the bus as possible and placed the antenna next to the window I was seated next to. I boarded each bus at the starting location indicated by King County Metro's maps, and began logging as soon as the vehicle began moving. Likewise, I ceased logging the data as soon as the vehicle ended at its terminus, and disembarked. Unfortunately, due to some quirks of both GPS modules losing tracking in certain areas or failing outright, there are some gaps in the data.

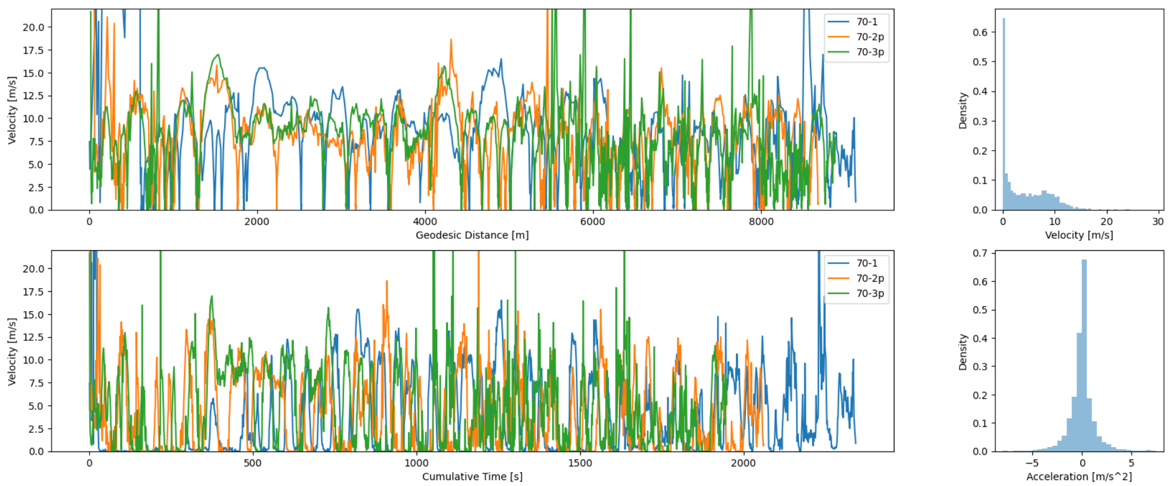


Figure 2 . 11*: Collected data for King County Metro route 70.*
Top Left: Velocity vs. Geodesic Distance.
Bottom Left: Velocity Vs. Cumulative Time.
Top Right: Cumulative Time Density vs. Velocity.
Top Left: Cumulative Time Density vs. Acceleration. -p suffix indicates data collected Via iPhone.

In Figure 2.11, I've illustrated the velocity profile of King County Metro's route 70, travelling from North to South across the three different data logging runs. As can be seen by the large spikes in velocity, there are a handful of points in the data that are not accurate to reality. Compounding this, the traffic and ridership conditions mean that it is nigh-impossible to reasonably compare the three data collection runs to a time-series profile generated by reRoute_Dynamics in this format. However, by aggregating and collapsing these into cumulative time-weighted histograms of velocity and acceleration, we can get a better picture of the vehicle behavior. These histograms, especially that of the acceleration, is the most important to validate the accuracy of – as the acceleration is what is mechanistically going to determine the power needed, and likewise, what the ESS experiences.

2.3.2: Modeling Data Collection

In order to generate histogram profiles using the model, it requires the requisite data to model the routes. As illustrated in section 2.2, to set up a reasonable model requires the geographic series of points the vehicle travels, the corresponding elevations, speed limits, stoplights, and bus stops. King county provides their GTFS feed [26], which has the geospatial data for each route and the corresponding stops. This was supplemented by publicly available GIS data from King County and Seattle, which contained the speed limits [27] and stoplights [28]. Finally, the data used to determine elevation was pulled from the Washington Lidar Portal [29], and was benchmarked using the NOAA Geodetic Survey Database [30].

2.3.3: Generating the Routes

GTFS Data is stored in a manner such that each 'Route' (i.e. 45, 70) has several associated 'shapes'. Each route that had GPS data collected for it had been for only one shape. To determine which GTFS geodata shape corresponded with the ridden routes, a visual comparison was performed between the two's plotted points.

Route	Associated GTFS Shape ID
45	31045010
70	20070015
8	31008006
49	11049009

Table 2.2: Route "shortname" used and associated with King County GTFS Shape ID for the collected data.

Because the geospatial shapes provided by King County contain the bare minimum number of points to define a route, large gaps between points existed. To avoid potential losses and inaccurate positional stop and signal associations, these points were interpolated such that no gap larger than 10 meters existed – slightly smaller than the length of the shortest 35 foot bus. To determine the elevation along the routes, each point had to have its elevation queried against King County Lidar data. The survey raster files contain high resolution imaging of elevation throughout King County, both as Digital Terrain Models (DTMs) and Digital Surface Models (DSMs). First, the raster files had to be reprojected to 'EPSG:4326' to match the projection system of the bus route data. This was done using the 'rasterio' [31] package transform and reproject methods. To verify the reprojection was accurate, the elevations of a handful of geodetic benchmarks were compared to their associated DTM and DSM elevations.

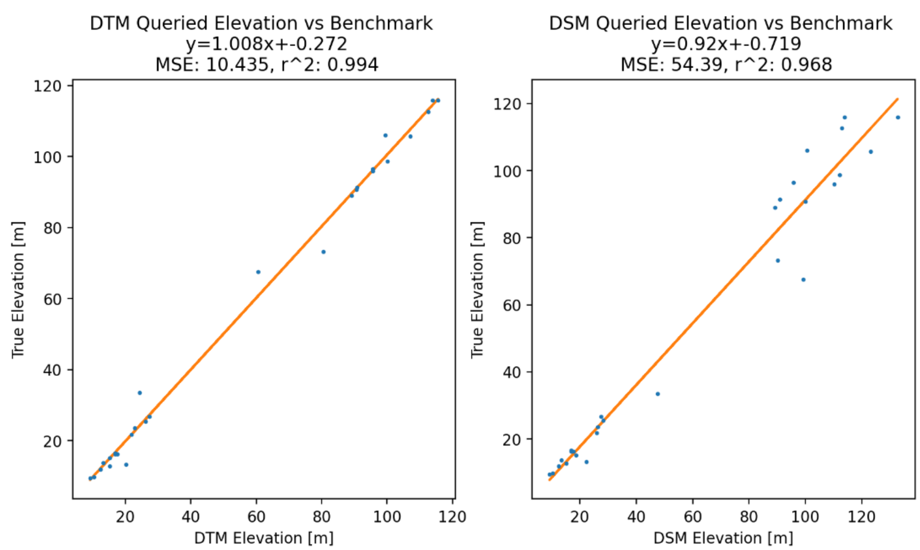


***Figure 2 . 12**: Comparison of elevation benchmarks.*
Benchmarks in Redmond, WA, compared to the DTM and DSM elevations of those positions.

As seen in Figure 2.12, both the Digital Terrain Model and Digital Surface model are quite accurate to the benchmarks, with the DTM being the better of the two by virtue of its Mean Squared Error and r^2 value of a linear fit being higher. This discrepancy in accuracy is likely due to the differences between a DTM and DSM – where the DTM is the ground-level elevation, the DSM is the topmost elevation at a given point. Since the benchmarks are placed in the ground, it is reasonable to expect the DTM is marginally more accurate – especially since the DSM can also include noise from trees and cars. However, in this lies an issue – the DTM can completely neglect bridges and overpasses, given those are not at ground-level. To illustrate this, compare this section of highway in Redmond, Washington:

Figure 2 . 13: *DSM (top) versus DTM (bottom).*

As visible in Figure 2.13, there are distinct chunks missing from the DTM that would be otherwise critical in determining driving conditions for a vehicle. As such, despite the DSM being marginally less accurate to the benchmarks, it is what must be used.

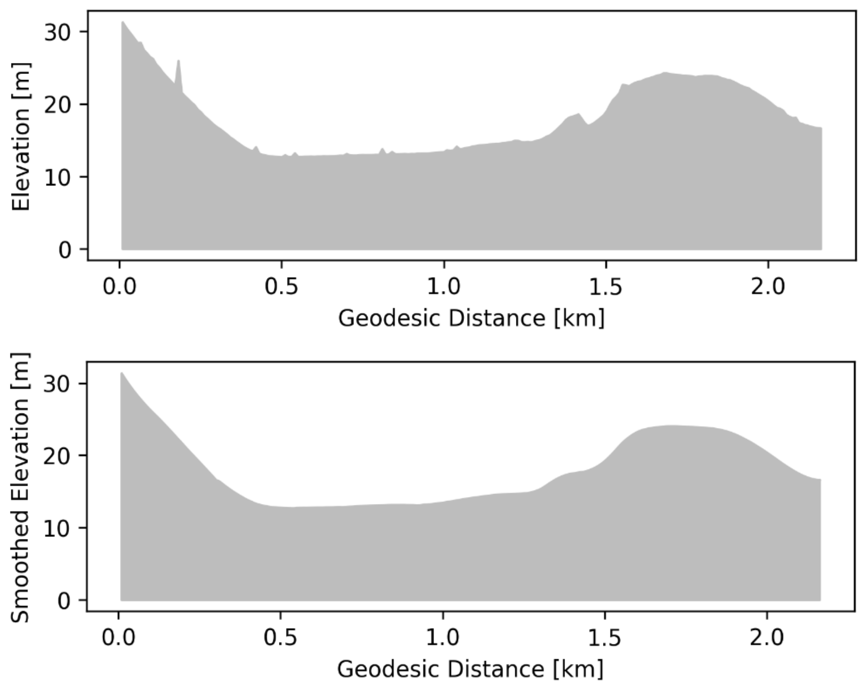


Figure 2 . 14: *Comparison between Unfiltered (Top) and Filtered (Bottom) DSM elevation for the stretch of highway in Figure 2.13.*

The noise in the DTM is then reduced using a Savisky-Golay filter, with a 3rd degree polynomial and a sample size of 25. This does not significantly affect the elevation profile, as can be seen in figure 2.14. Likewise, the slope grade as calculated by $\frac{dy}{dx} * 100$ is smoothed using a Savisky-Golay filter with a 3rd degree polynomial and sample size of 43 - approximately 430 meters. While road grades for arterials in Seattle are permitted to be up to a 9% [32], an upper limit of 7.5% was set to help filter out any more potential noise from cars or trees.

After querying the filtered DTM elevation for each point along the selected routes, these points had to be associated with its requisite stops, speed limits, and signals. To determine a route's speed limits, each point on the route is queried against the king county road data [27] . If a point is within $3 * 10^{-4}$ latitude/longitude degrees of a road, the point gets assigned that limit. If not, the point uses the previously assigned speed limit. The same methodology is used for assigning stops and signals, with the margin instead being 10 meters. As an additional step for the stops, the bus stop data is filtered to match the specific shape.

2.3.4: Generating the Monte Carlo Simulation

As can be seen in section 2.3.1, comparing individual velocity profiles for accuracy is nigh-impossible. So instead, I opted to run a Monte Carlo simulation to compare the aggregated velocity and acceleration histograms. To generate this simulation, a handful of the many possible modeling parameters had to be selected to vary. Since the reRoute_Dynamics Trip module is designed for that, the bulk of the variation takes place in modifying the external conditions. The Bus and ESS will not vary significantly between runs, save for the auxiliary power use and driver behavior, so those modules see the least variation in parameters.

Module	Parameter	Minimum	Maximum
Trip	$\%_{traffic}$	.1	1
Trip	$\%_{signal}$	.2	.8
Trip	v_{wind}	0	4
Trip	θ_{wind}	45	315
Trip	t_{sig}	5	150
Trip	t_{stop}	20	100
Trip	n_{riders}	15	60
Bus	f_{br}	.1	.5
Bus	f_a	0	.4
Bus	dx_{max}	150	400

Table 2.3: Validation Monte Carlo Parameters.

For the traffic, I selected the range of .1 to .9 due to there not being a clear way to correlate the traffic I experienced while riding the buses to a given value. Because full traffic signal cycles are expected to last approximately 240 seconds [33], the signal rest time range was set between 5 and 150 seconds, to allow for a little overhead. And because of that total cycle time of 240 seconds, the time-based chance that a vehicle would hit any one signal should be approximately 50%. Given that expected value, a window of 20%-80% hitting any given signal is a reasonable range. The predominant wind direction in Seattle during March and April is towards the south[34], so I chose a range of 45 degrees to 315 degrees from North. The range of wind speeds in Seattle that are ~2 standards of deviation away from the mean are between 0 and 4 m/s [34], so that was used for the simulation. To help determine ridership ranges, I attempted three tallies of riders while riding the buses.

Route	Riders
70	~47
49	~23
8	~51

Table 2.4: Tally of approximate unique riders for route 70, 8, and 49.

These tallies are imperfect measurements, but can inform a reasonable expectation. With a margin for error, a range of 15 to 60 was selected. For the nominal braking factor, 0.1-0.5 was used, as like with traffic there's no easy way to correlate braking factor to what was experienced on the bus rides. No driver of a bus I rode immediately jammed on the brake at full pressure, so having the maximum be in the validation model range was unreasonable. Likewise, there's no easy way to account for the risk tolerance of the drivers, so the dx_{max} and f_a ranges were set to be low-risk as that is what king county would expect. To help determine dx_{max}, the set point distances selected were the approximate lengths of offramps on Interstate 5.

Complicating the validation are the bus models of each route. Routes 45, 8, and 70, each use the XDE60 model, while the 49 uses the XT40. Because none of these vehicles are fully-electric, it is hard to account for what fraction of the time or parts of a route the electric motor would kick in. So, the motor power was set to be the respective power of the closest fully electric model.

	XDE60	XT40
$m\ [kg]$	19096 [35]	13835 (assumed) [35]
$w_f\ [m]$	2.6 [35]	2.6 [36]
$h_f\ [m]$	3.3 [35]	3.3 [36]
C_d	.7 (assumed from aggregate)	.7 (assumed from aggregate)
C_f	.01 [37]	.01 [37]
$a_{br}\ [\frac{m}{s^2}]$	1.5 [38]	1.5 [38]
f_{br}	-	-
f_a	-	-
f_i	1.1 (assumed)	1.1 (assumed)
$dx_{max}\ [m]$	-	-
$a_{profile}$	Braunschweig	Braunschweig
$a_{max}[\frac{m}{s^2}]$	.4 (assumed)	.4 (assumed)
$dt_{max}[s]$	.5	.5
$P_{max}[W]$	30000 (assumed) [39]	246000 [36]

Table 2.5: Bus Model Parameters for the XDE60 and XT40 as used in the Monte Carlo Simulation.

The Coefficient of Drag for these specific models is not readily available. Studies on similar buses have demonstrated the drag coefficient ranging anywhere from .9 to .3, with the typical value being ~.6 [12], [40]. Previous works with Longitudinal Dynamics Models have used a value of .7. For our validation purposes, we will also use .7.

The Acceleration Profile used was based on a mathematical fitting of the accelerations experienced in the Braunschweig City Driving Cycle [41]:

$$-\frac{m}{t^2} * e^{\frac{m}{t}+b} \quad \text{(eqn. 2.17)}$$

Where t is time from zero, m and b are the slope and y-intercept of the linearized acceleration data. m and b are -4.9661 and 2.9465, respectively.

Of these parameters that could vary and have an impact, the acceleration profile is one of the most important and hard to quantify the impact of on accuracy. To attempt to better understand this, I attempted the Monte Carlo for route 45 with three different acceleration profiles.

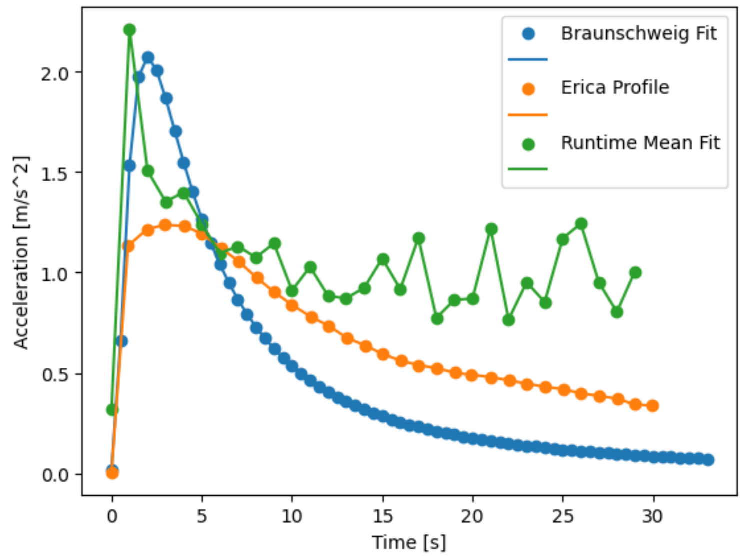


Figure 2 . 15: *Comparison of different possible acceleration profiles.*

The Braunschweig fit was the acceleration profile used for the initial results. The profile labeled 'Erica Profile' was an acceleration profile previously used by Erica Eggleton, the source of which is unknown. Finally, the 'Runtime Mean Fit' profile is the mean acceleration calculated by averaging the collected data's first 30 seconds of every acceleration.

For validation purposes, there is no need to use the ESS, as there is no way to use it to validate against the collected GPS data, and likewise has no impact on the actual driving behavior of the bus. The Monte Carlo Simulation was then run 300 times each, using a different random seed from python's in-built 'random' package.

2.4: Results & Discussion

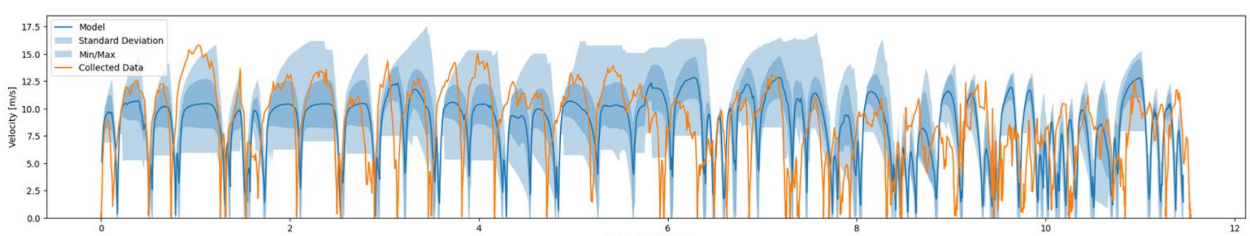


Figure 2 . 16: *geodesic comparison of route 70's velocity profile and the Monte Carlo Model*

While it is possible to visualize the fit of the Monte Carlo model for a given route to any given run of the collected data, it is hard to quantify how accurate the model is to reality. So, to better assess these fits, I chose three different aggregate metrics by which to compare the Monte Carlo simulations to the real-world data. The first, and simplest, is cumulative time. Using the mean and standard deviation total runtime for a route, we can perform a two-tailed student's t-test, which yields the following results.

Route	GPS Runtime x̄/σ [s]	MC Runtime x̄/σ [s]	t-score	p-value
45*	3077.78 / 134.07	3150.88 / 456.89	-0.8938	0.37211
70	2117.95 / 165.64	20170.49 / 273.00	-0.4389	0.66098
8	2781.06 / 174.32	2696.23 / 372.94	0.824185	0.410486
49**	1742.45 / 9.54	1887.59 / 258.76	-8.84783	$7.85 * 10^{-17}$

Table 2.6: Runtime means and standard deviations, t-scores, and p-values.

We can also benchmark the Monte Carlo Runtime and GPS Runtime against King County Metro's expected runtimes for each route [42].

Route	Expected Runtime x̄/σ [s]	t-score vs GPS	t-score vs. MC	p-value vs. GPS	p-value vs. MC
45*[1]	3244 / 484	1.71558	1.45587	0.09066	0.14628
70	2123 / 179	0.05186	-1.55404	0.9587	0.12097
8	2113 / 225	-6.4188	-16.489344	$2.729 * 10^{-8}$	$3.45 * 10^{-46}$
49*[2]	1821 / 223	2.656	-2.050528	0.01010	0.0410405

Table 2.7: Runtime mean and standard deviations as expected by king county, with the T-score and p-values against both the Monte Carlo simulation and the GPS data.

The Monte Carlo simulation and the GPS data for each route aside from route 49 has a p value greater than .05, and thus fails to reject the null hypothesis. The expected runtime for each route is not nearly as accurate to the GPS data, with the only value that doesn't reject the null hypothesis concretely being route 70. The Monte Carlo Simulation and the expected runtime fail to reject the null hypothesis in route 70 and 45, but the margin on those is worse than that of the simulation against the GPS.

In other words, it seems clear that the Monte Carlo simulation is more accurate to the reality of the bus runtimes than the quoted king county metro runtimes – with the expectation value being, on the whole, slightly longer than reality, while the Monte Carlo simulation is slightly quicker than the projected times. Runtime alone cannot perfectly represent the accuracy of the Monte Carlo model, however. Because the primary driving factor behind running a Monte-Carlo simulation is to get a distribution of expected values rather than a perfect alignment, we can compare the proportional time spent at certain velocities or accelerations, like what was seen in figure 2.11.

---

*[1] Route 45's first run has a slightly undermeasured runtime due to sampling issues.
*[2] Route 49 only has 2 GPS samples.

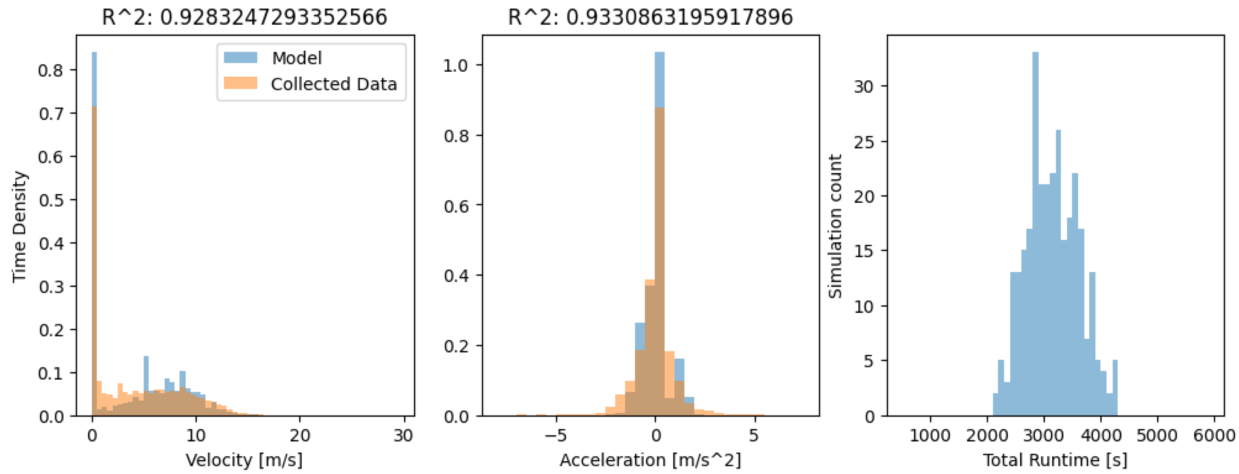


Figure 2 . 17*: Monte Carlo Histograms for KCM Route 45.*
(left & center, bin size 0.5) Proportional time spent at velocity and acceleration,
(right, bin size 100) distribution of runtimes

From Figure 2.17, the coefficient of determination for velocity and acceleration are 0.928 and 0.933, respectively, as calculated from the counts within each bin of the histograms. The total runtime appears to follow a normal distribution, centered squarely on the GPS mean runtime of 3150.88 seconds. From examining the velocity histogram, it appears as if the model is at rest slightly more than reality, and the real bus spends a bit more time creeping at lower velocity than the model produces. The acceleration histogram produced by the model is more conservative in braking and decelerating than the collected data. There is a second peak within what the model produced that implies the model's bus is dwelling less at the lower accelerations, and higher at the middling ones. Further examples for the other routes can be seen in Section 2.6.3.

Route	Velocity r^2	Acceleration r^2
45	0.928324	0.933086
70	0.836055	0.838288
8	0.90480	0.94644
49	0.72225	0.922406

Table 2.8: Velocity and Acceleration coefficients of determination from Monte-Carlo simulation.

These trends hold across the other three routes, with the model being able to better capture the variation in acceleration rather than velocity. As can be seen in the additional figures in 2.6.3, while the simulations capture the primary trends of both distributions well enough, the model does not account for creeping motion and slower velocity. Ultimately, acceleration is the more important value of the two to be accurate, as it is more participatory in the calculation of power.

	Erica's Profile			Runtime Mean		
Route	Velocity r^2	Acceleration r^2	p-value	Velocity r^2	Acceleration r^2	p-value
45	0.849946	0.93979	$6.7 * 10^{-11}$	0.413166	0.818168	$6.2 * 10^{-32}$
70	0.723159	0.91264	$7.27 * 10^{-6}$	0.427310	0.617509	$1.5 * 10^{-5}$
8	0.840846	0.93006	0.000559	0.429987	0.847375	$3.5 * 10^{-13}$
49	0.612433	0.95508	$3.4 * 10^{-51}$	0.2270265	0.826001	$2 * 10^{-83}$

Table 2.9: Comparison of velocity and acceleration coefficients of determination for Erica's Profile and the Runtime Mean fit.

From this, we can see an illustration that the acceleration profile can have an outsized impact. Using the 'Erica Profile', the velocity r^2 sees a slight loss in accuracy, while the acceleration sees an increase. The P-values for the runtime against the GPS also see a dramatic shift, with route 45 becoming almost spot-on. The Runtime Mean, however, is measurably worse in every category. Out of these three options, the Braunschweig fit appears to be the best given its relatively consistent accuracy for velocity, acceleration, and time.

2.5: Conclusions

While there are some minor discrepancies in accuracy, reRoute_Dynamics is able to reasonably produce re-creations of the velocity and acceleration behavior of real-world routes. This demonstrates the efficacy of reRoute_Dynamics as a modeling tool, and demonstrates that it is capable of simulating the real-world bus behaviors to a reasonable degree. The acceleration behavioral accuracy, being core to calculating the velocity, time, and power, is particularly important for the model to capture, which it did – capturing between 83% to 94% of the acceleration variance. Likewise, the run time of the model being generally more accurate to reality than the quoted King County times is yet another confirmation that this toolkit capable of accurately simulating EV routes.

With that said, there are certain aspects of the modeling that could be more accurate, and the methods more precise. There are three primary areas of focus for further development of this tool. First, while the tool is friendly enough for someone who has experience with python, it requires a heavy amount of data handling and sorting in order to run. While there are some common formats used for the storage of this information and the tool provides useful ways of interpreting those formats, there are ways the tool could require less up-front work. Possible solutions include providing a script to help sort through GTFS-sorted routes. The second area of focus is in how signal behavior is assessed. While a user could choose to provide a specific series of which signals they want the vehicle to hit and have the probability of hitting a stoplight be 100%, the current method makes it easier to randomly select a stop. Likewise, the method of determining rest times at signals is simplistic and thereby cannot fully encompass the variability that having each stop have its own set time could. Traffic modeling could be improved, as having a single constant uniformly affect the speed limits of a route is not fully accurate to the experience of a vehicle. Third and finally, the ESS modeling and the simulation output could be improved to have more heuristics, like a more easily accessible change in state of charge. These could also be improved by adding an aging model, where an ESS could have theoretical decay be observed.

Despite these areas of improvement, the toolkit is already highly customizable and has been shown to, even with a number of assumptions and distributions, accurately model trends in EV Bus data, and can enable further study of these routes, trends, and impacts.

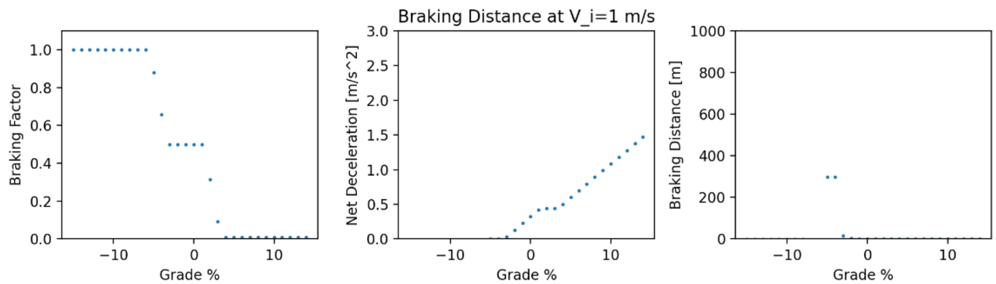
2.6: Appendix

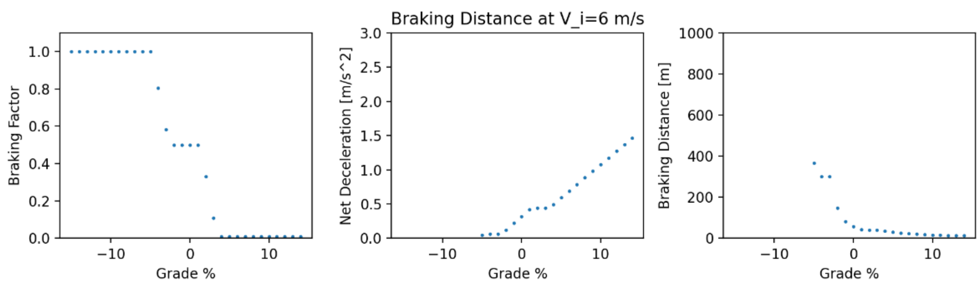
2.6.1: Braking Behavior Supplemental Figures

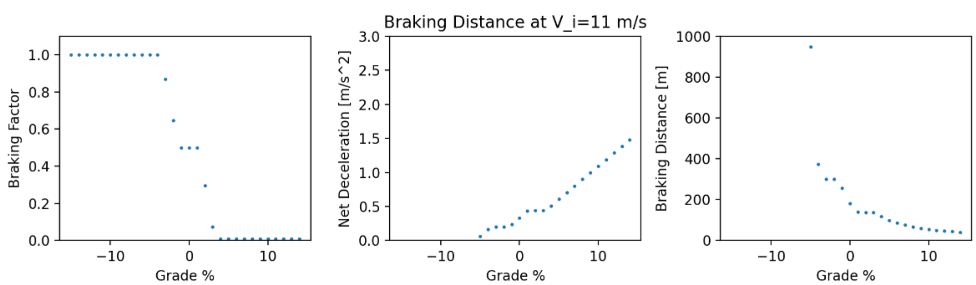
For the purposes of this behavior analysis, the following parameters were used:

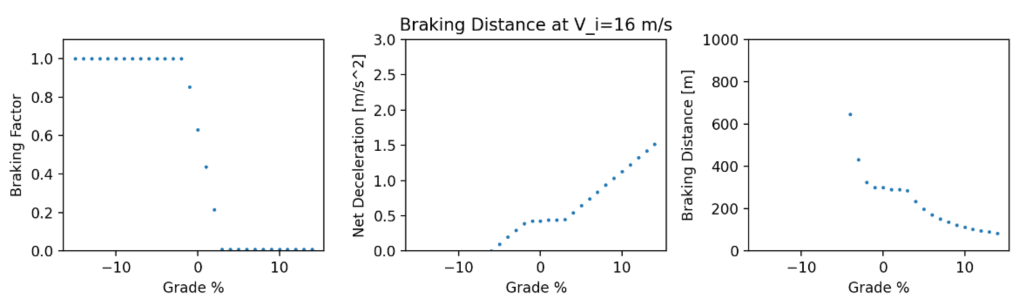
$$\theta_{bus} = 0, \qquad \theta_w = 180, \qquad v_w = 5\frac{m}{s}, \qquad \rho = \frac{1.2kg}{m^3}, \qquad C_d = 1, \qquad A_f = 2\,m^2,$$

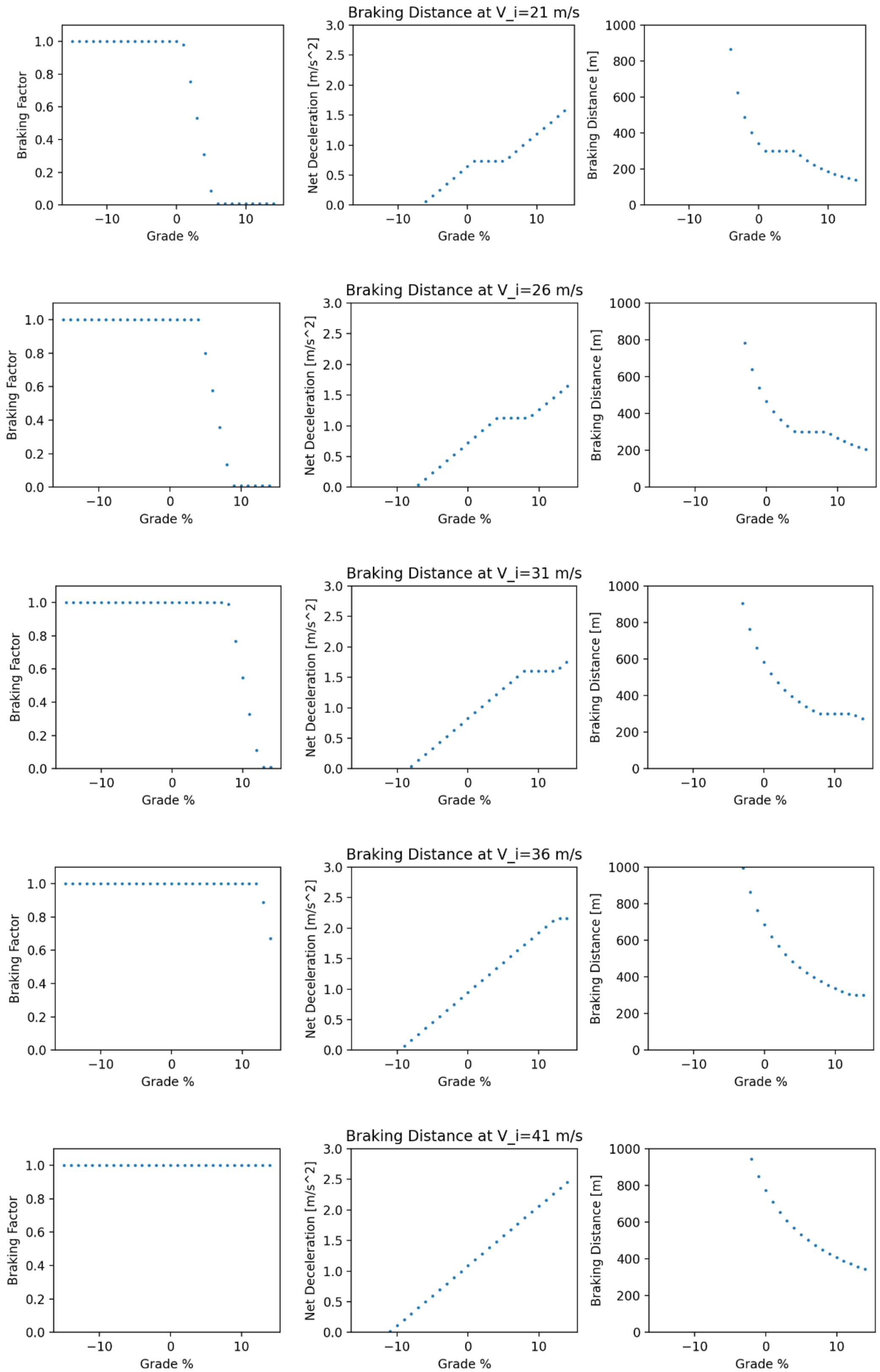
$$m = \left(A_f^3\right)^{\frac{1}{2}} * 1000\;kg$$

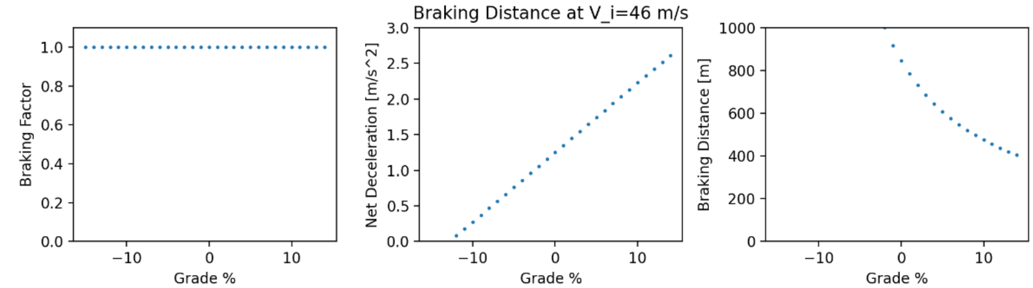








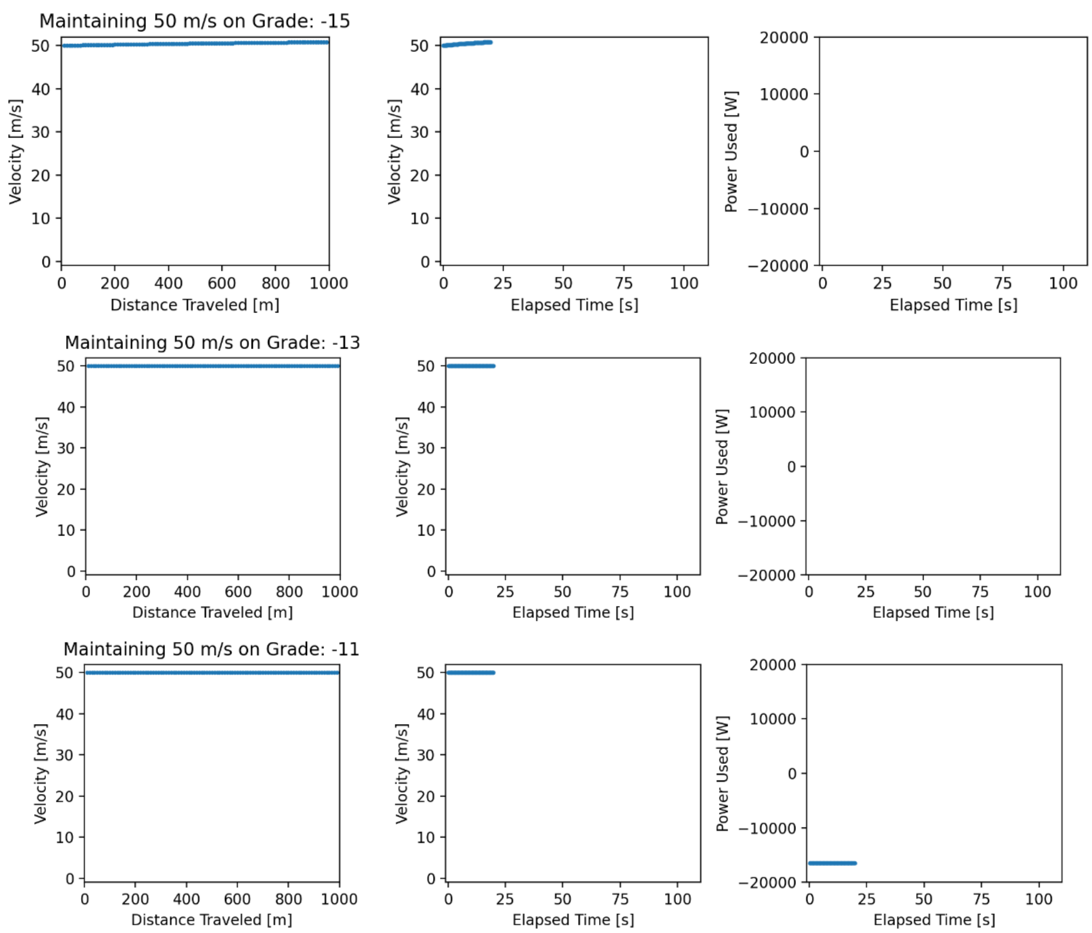

Braking Distance at V_i=21 m/s
Braking Factor
Net Deceleration [m/s^2]
Braking Distance [m]
Grade %
Braking Distance at V_i=26 m/s
Braking Factor
Net Deceleration [m/s^2]
Braking Distance [m]
Grade %
Braking Distance at V_i=31 m/s
Braking Factor
Net Deceleration [m/s^2]
Braking Distance [m]
Grade %
Braking Distance at V_i=36 m/s
Braking Factor
Net Deceleration [m/s^2]
Braking Distance [m]
Grade %
Braking Distance at V_i=41 m/s
Braking Factor
Net Deceleration [m/s^2]
Braking Distance [m]
Grade %

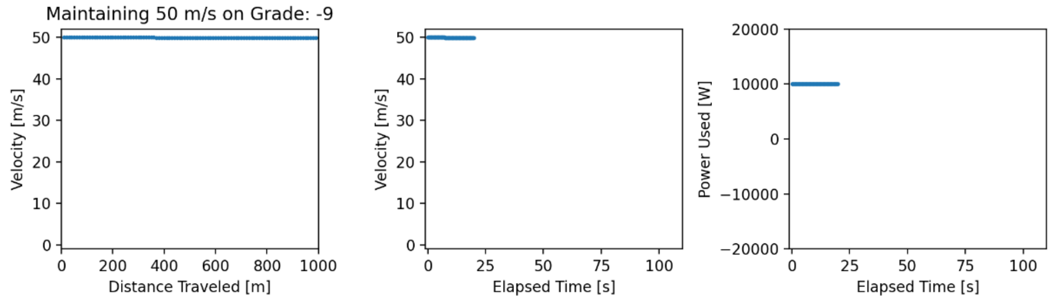


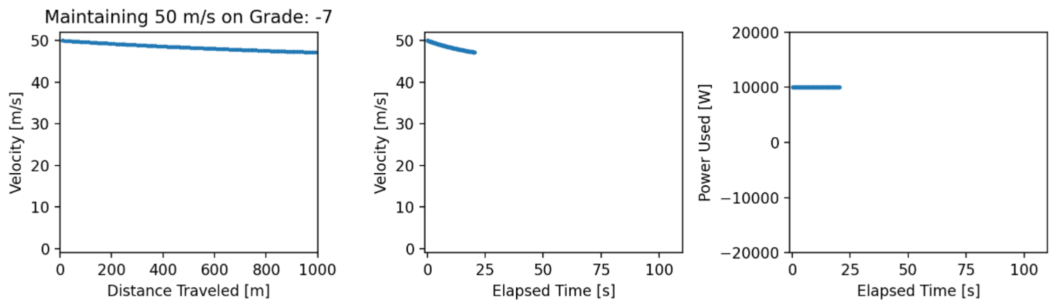
2.6.2: Maintaining Speed Behavior Supplemental Figures

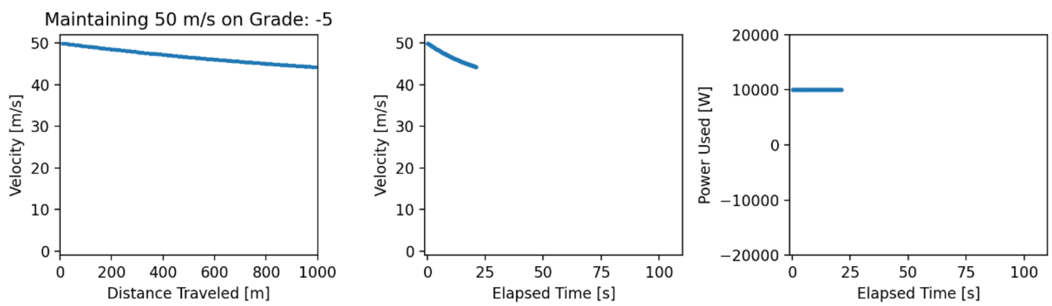
For the purposes of this behavioral analysis, the same parameters were used as in section 2.6.1, with the following additions:

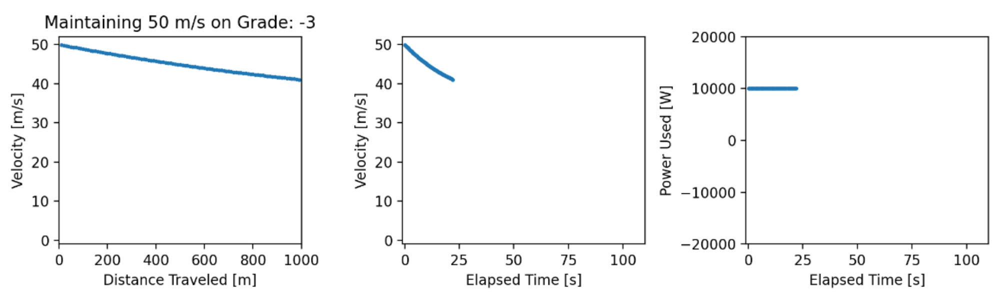
$$P_{max} = 10000\, Watts$$

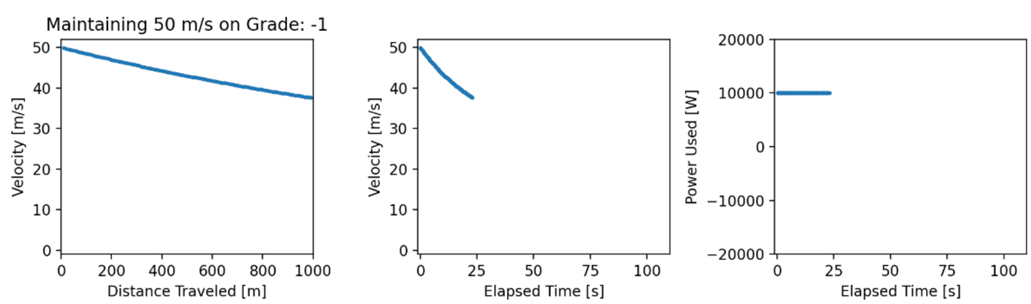


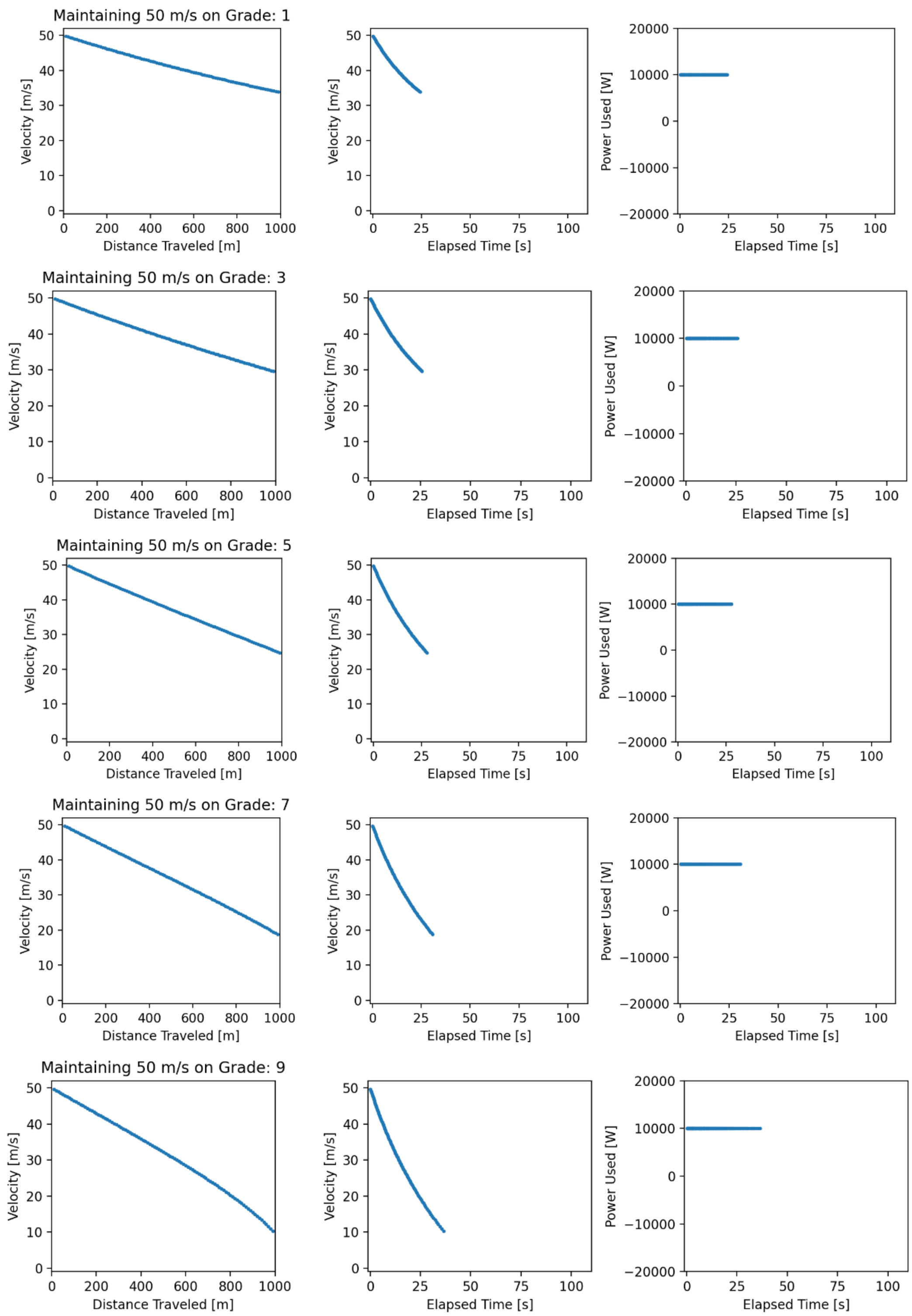

Maintaining 50 m/s on Grade: -9
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]

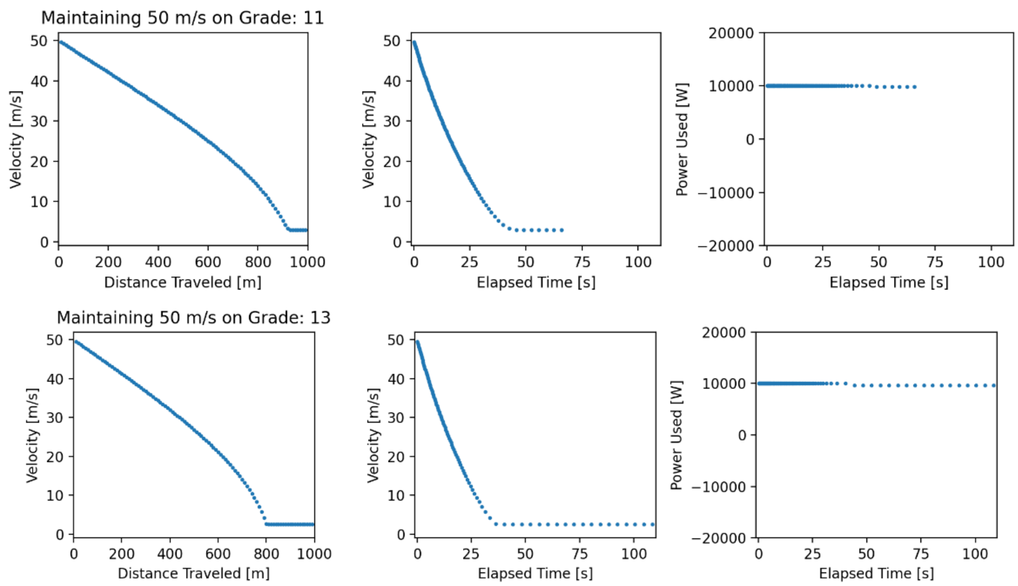

Maintaining 50 m/s on Grade: -7
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]


Maintaining 50 m/s on Grade: -5
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]


Maintaining 50 m/s on Grade: -3
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]

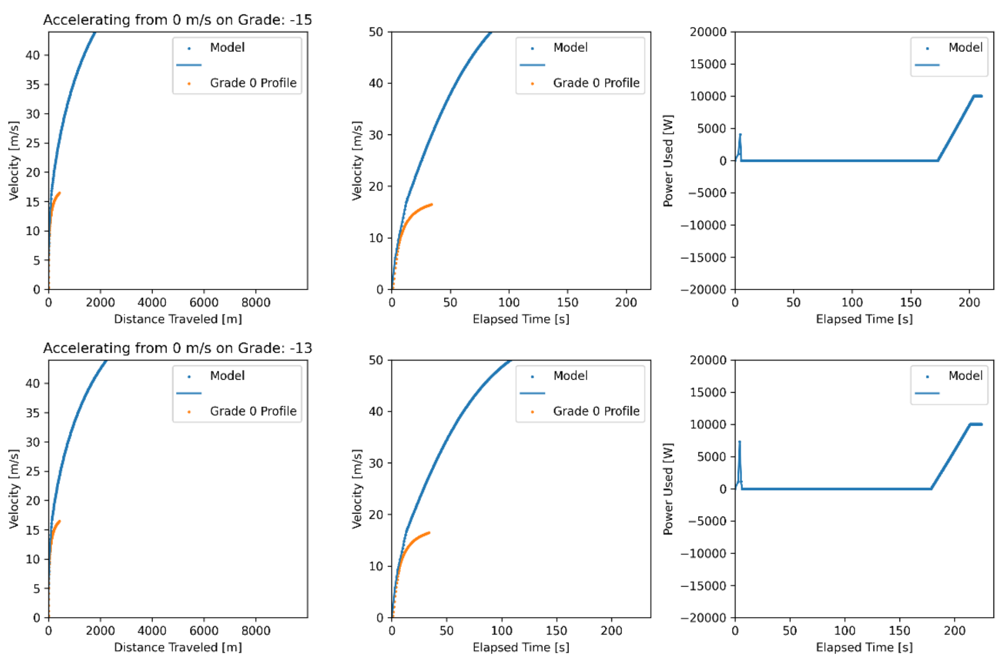

Maintaining 50 m/s on Grade: -1
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]

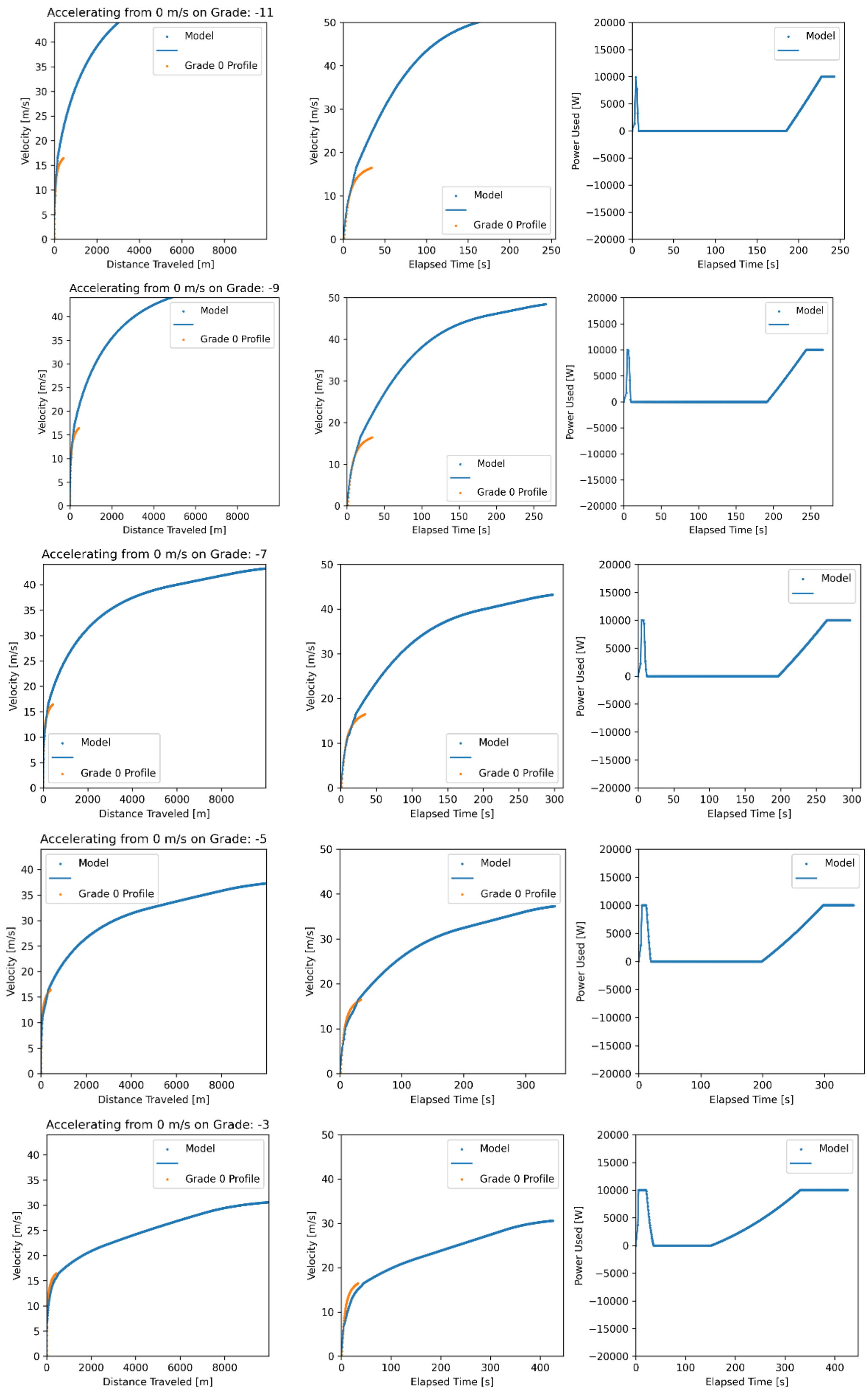

Maintaining 50 m/s on Grade: 1
Maintaining 50 m/s on Grade: 3
Maintaining 50 m/s on Grade: 5
Maintaining 50 m/s on Grade: 7
Maintaining 50 m/s on Grade: 9
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]

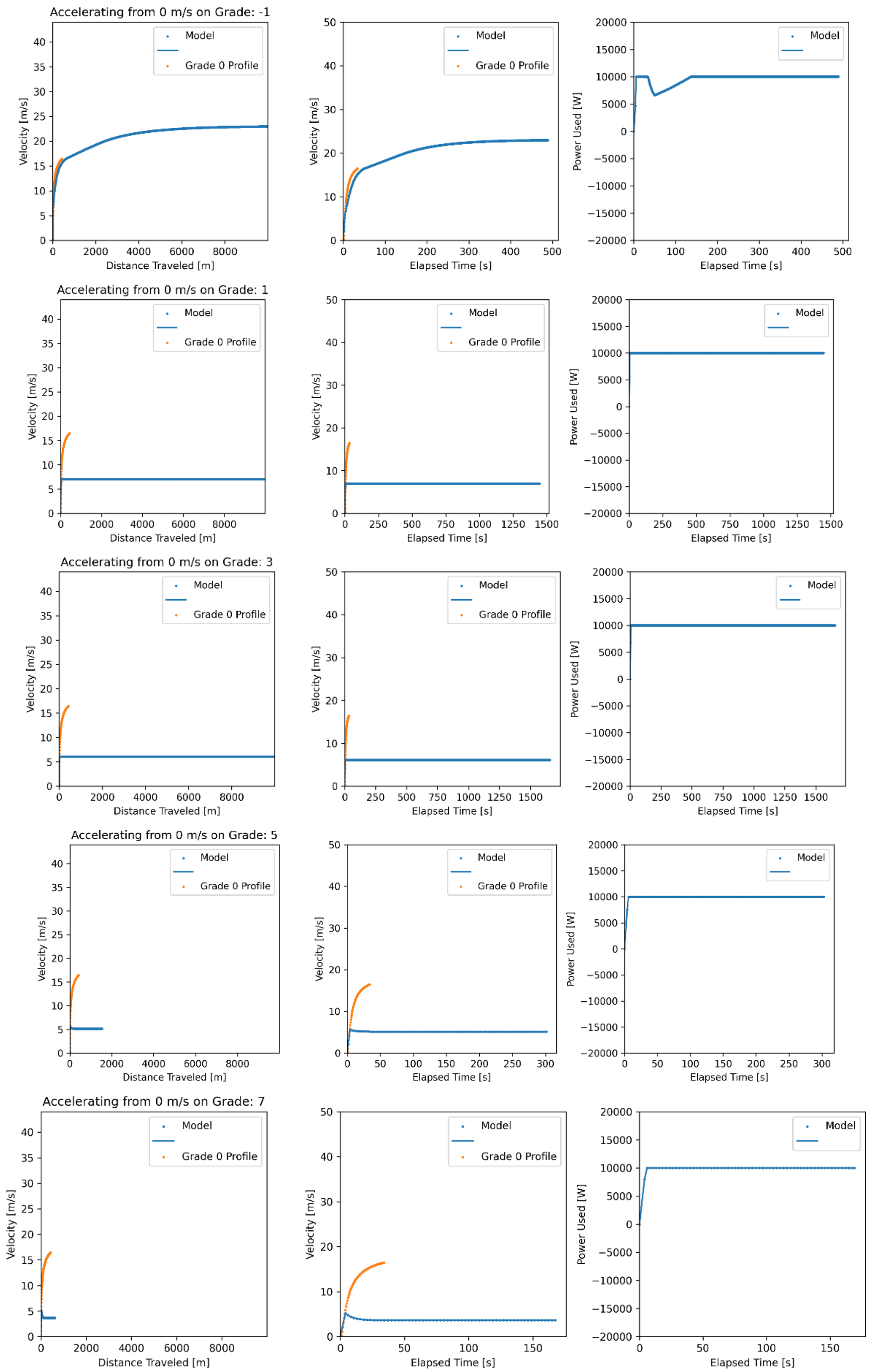


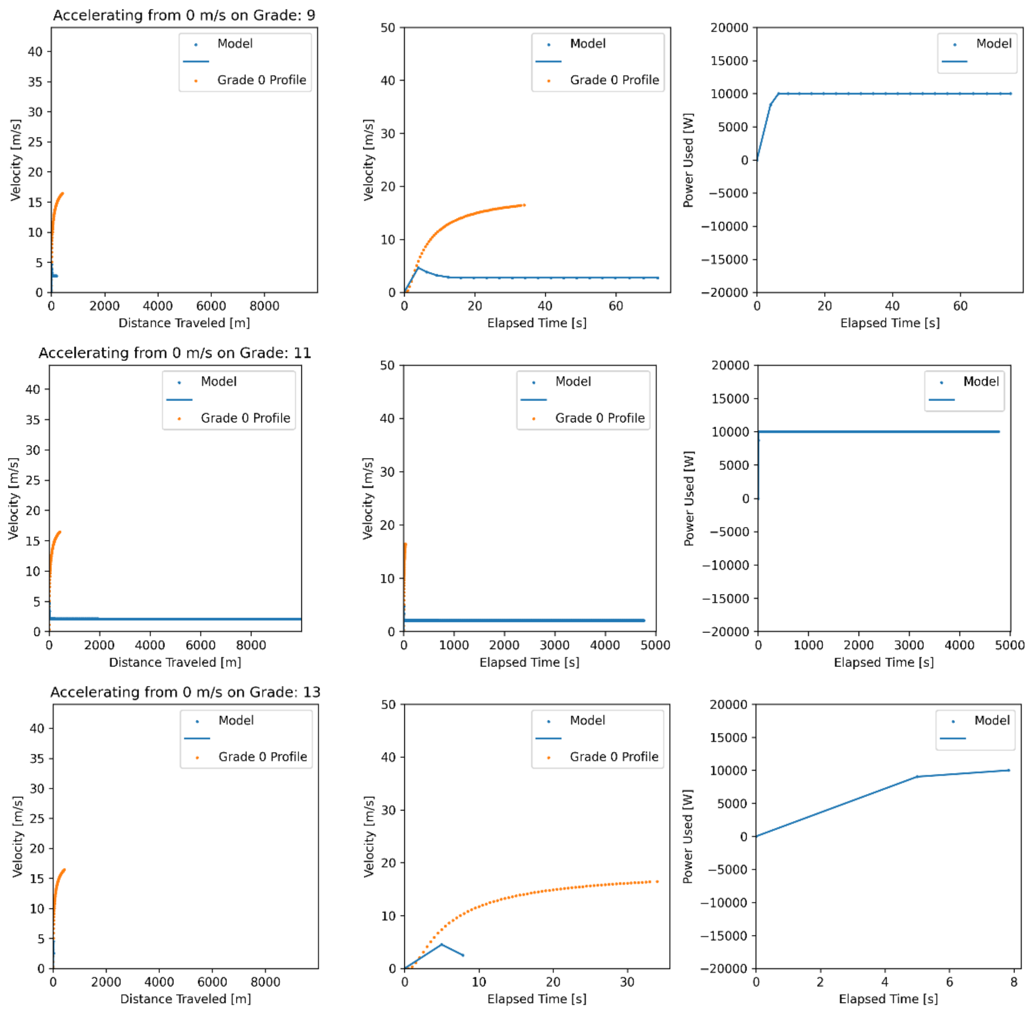
2.6.3: Acceleration Behavior Supplemental Figures

For the purposes of this behavioral analysis, the same parameters were used as in section 2.6.1 & 2.6.2




Accelerating from 0 m/s on Grade: -11
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]
Model
Grade 0 Profile
Accelerating from 0 m/s on Grade: -9
Accelerating from 0 m/s on Grade: -7
Accelerating from 0 m/s on Grade: -5
Accelerating from 0 m/s on Grade: -3


Accelerating from 0 m/s on Grade: -1
Accelerating from 0 m/s on Grade: 1
Accelerating from 0 m/s on Grade: 3
Accelerating from 0 m/s on Grade: 5
Accelerating from 0 m/s on Grade: 7
Model
Grade 0 Profile
Velocity [m/s]
Power Used [W]
Distance Traveled [m]
Elapsed Time [s]


Accelerating from 0 m/s on Grade: 9
Accelerating from 0 m/s on Grade: 11
Accelerating from 0 m/s on Grade: 13
Model
Grade 0 Profile
Velocity [m/s]
Distance Traveled [m]
Elapsed Time [s]
Power Used [W]

Chapter 3 – Modeling and Analyzing the needs of Transit Routes in King County

3.1: Introduction

In Chapter 2, reRoute_Dynamics was demonstrated to reasonably model the trends in a handful of routes from King County, Washington. As outlined in Chapter 1, previous work has been done to identify the variables that most impact the mileage and energy use of these vehicles through Longitudinal Dynamics Models. These models have used collected velocity data and bus parameters to determine impact on load, and often investigate variables such as dwell time, average speed, passenger numbers, and bus model. These studies have typically found that mass, elevation change, and auxiliary load have the most impact on the energy consumption. However, these models create LDMs by beginning with a velocity profile, rather than reRoute_Dynamics's approach of generating the cycles whole-cloth. This increases the total number of possible variables and routes that can be modeled and investigated, and could possibly demonstrate new trends in what impacts the energy consumption – and more intricate information on the way that energy is being consumed.

3.2: Methodology

As in Chapter 2, the same routes (45, 70, 8, and 49) will be used in order to remain assured of the reasonability of the model. However, to get a more comprehensive analysis of the needs of the routes, the Monte Carlo simulation parameters were expanded, and the vehicles used were changed to fully-electric models rather than their hybrid or trolley siblings. The Monte Carlo simulation ran 300 iterations per shape, using an entirely new random seed for each run.

Module	Parameter	Minimum	Maximum
Trip	$\%_{traffic}$	.1	1
Trip	$\%_{signal}$	.1	1
Trip	v_{wind}	0	15
Trip	θ_{wind}	0	360
Trip	t_{sig}	1	150
Trip	t_{stop}	20	150
Trip	n_{riders} *3	-	-
Bus	P_{aux}	3000	10000
Bus	f_{br}	.1	1
Bus	f_a	0	1
Bus	dx_{max}	150	400

Table 3.1: Comprehensive Monte Carlo Simulation parameters.

*3 n_{riders} depends on the ridership of the individual route – taking the minimum and maximum for each direction of travel – and as such is variable.

Bus	XE60	XE40	XE35
$m\ [kg]$	20638 [35]	13835 [35]	13920 [35]
$w_f\ [m]$	2.6 [35]	2.6 [35]	2.6 [35]
$h_f\ [m]$	3.38 [35]	3.38 [35]	3.38 [35]
C_d	.7 (assumed from aggregate)	.7 (assumed from aggregate)	.7 (assumed from aggregate)
C_f	.01 [37]	.01 [37]	.01 [37]
$a_{br}\ [\frac{m}{s^2}]$	1.5 [38]	1.5 [38]	1.5 [38]
f_{br}	-	-	-
f_a	-	-	-
f_i	1.1 (assumed)	1.1 (assumed)	1.1 (assumed)
$dx_{max}\ [m]$	-	-	-
$a_{profile}$	Braunschweig	Braunschweig	Braunschweig
$a_{max}[\frac{m}{s^2}]$	.4 (assumed)	.4 (assumed)	.4 (assumed)
$dt_{max}[s]$	.5	.5	.5
$P_{max}[W]$	30000 [39]	30000 [39]	30000 [39]

Table 3.2: Model Parameters for the fully-electric vehicles used in the Monte Carlo Simulations.

For the ESS design, the vehicles are assumed to be identical in everything except pack structure.

Bus	XE60	XE40	XE35
η_m*4	.93	.93	.93
η_i*4	.88	.88	.88
η_a*4	.89	.89	.89
η_r*4	.54	.54	.54
$P_{aux}[w]$	-	-	-
$P_{regen}[w]$	20000 [21]	20000 [21]	20000 [21]
$V_{cell}[V]$* 5	3.3	3.3	3.3
$R_{cell}[ohms]$* 5	.008	.008	.008
SP_{mod}*6	12, 8	12, 8	12, 8
SP_{bus}*6	16, 40	16, 32	16, 32
$Q_{cell}[Ah]$* 6	2.3	2.3	2.3

Table 3.3: ESS design parameters for the fully-electric vehicles used in the Monte Carlo simulation.

Then, in order to determine which of the selected model parameters have the most hypothetical impact on mileage, I performed a sensitivity analysis across the same space as the Monte Carlo simulation. In this case, I chose to only use the XE60 Model, as its higher initial mass will likely result in more dramatic changes. Using the python package SALib, I sobol-sampled the

*4 Assumed based off of [https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/121450/2539-07.pdf?sequence=1]
*5 Assumed based off of the A123 LFP 26650 cells as used in the XDE models
*6 Assumed based off of the smallest ESS for available for each BEV [https://www.newflyer.com/site-content/uploads/2023/01/Xcelsior-CHARGE-NG_Brochure.pdf], and the module structure for XDE models (with a target of 633 Volts).

parameter space for each route (40, 70, 8, 49) for a total of 6144 permutations each. Finally, using the results of the initial Monte Carlo simulation, we can observe modeled cell-level behavior.

3.3: Results & Discussion

Run	Mileage Mean/STD [mi/kwh]	99.67% Success Capacity [kWh]	ΔSOC [%]	Velocity Mean/STD [m/s]	Mean/STD Grade Mean/STD [%]	Expected mileage [mi/kwh]
45 – XE35	0.30/0.05	38.89	11.27	8.73/3.85	0.72/3.69	0.527
45 – XE40	0.31/0.05	38.74	11.23	8.73/3.85	-	0.510
45 – XE60	0.22/0.03	52.13	10.03	8.69/3.82	-	0.275
70 – XE35	0.34/0.06	22.10	6.41	7.45/4.07	-0.27/3.76	0.527
70 – XE40	0.34/0.06	22.02	6.38	7.45/4.07	-	0.510
70 – XE60	0.26/0.04	28.51	5.48	7.39/4.05	-	0.275
8 – XE35	0.30/0.04	34.38	9.96	8.47/3.85	0.59/4.43	0.527
8 – XE40	0.31/0.04	34.24	9.93	8.47/3.85	-	0.510
8 – XE60	0.22/0.03	43.8	8.42	8.41/3.81	-	0.275
49 – XE35	0.30/0.05	21.18	6.14	7.87/4.09	0.41/4.15	0.527
49 – XE40	0.30/0.05	21.09	6.11	7.87/4.09	-	0.510
49 – XE60	0.22/0.03	27.83	5.35	7.84/4.08	-	0.275

Table 3.4: Monte Carlo Simulation distilled results.

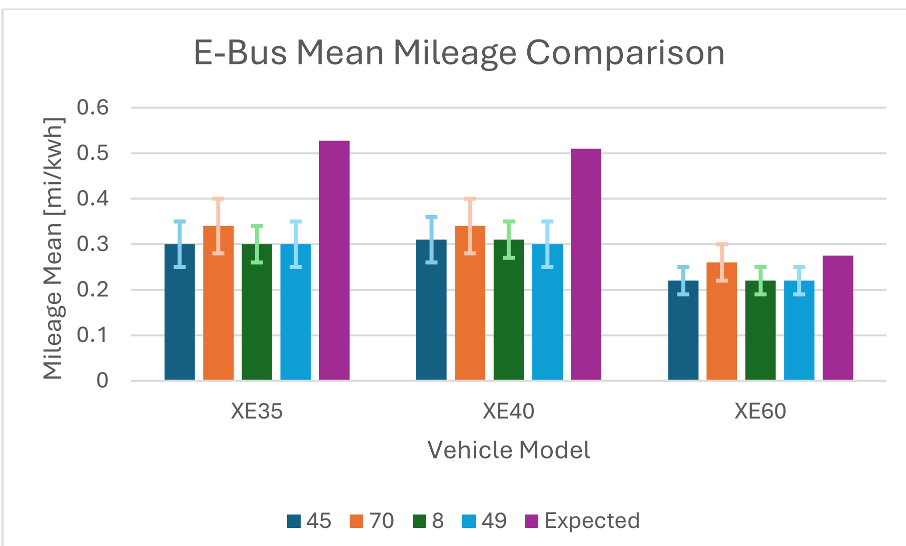


***Figure 3 . 1**: E-Bus Mean Mileage Comparison*

While the XE60 has a mileage that is close to the expected value across all four routes, the XE35 and XE40 are routinely underperforming their quoted manufacturer mileage. However, in reviewing King County Metro's pilot datasheet from 2018 [10] , the mileage for their battery-electric busses was 0.42 mi/kWh. While the bus manufacturer was different, this value is representative of the reality of a 40-foot bus. Per the measurement standards of New Flyer mentioned in Chapter 1, their quoted capacity is not reflective of real-world mileage, and thus we would expect the results to be lower. The pilot mileage demonstrates this to be the case.

Critical to the development of any transit route is the knowledge of how much energy would be needed to get the vehicle successfully from the beginning to the end of its trip. During the initial 2016 pilot, the vehicles being used consumed ~75% of their total capacity, thus requiring them to be charged after each trip. However, with the advancements in ESS technology since 2016, this is no longer necessarily the case. For each route, the third column in table 3.4 indicates the total amount of energy needed for the vehicle to complete that segment, 299 times out of 300. The fourth column demonstrates what the change in state of charge that represents for a fresh ESS, and it is clear that regardless of the model, the vehicles would be entirely capable of traversing these trips several times. And, since these trips were in single directions, it is likely that for those with a mild incline (45, 8, 49), the ΔSOC would be lower on the return leg.

For the XE35 and XE40 models, 99.67% of the times this trip was run, the consumed capacity was approximately identical, stemming from the similar masses and identical ESS capacity. The XE60's 99.67% quantile value for consumed energy is understandably the highest, regardless of route. As for the influence of the grade, the overall gradual decline for route 70 is reflected in an increase in mileage. Interestingly, the progressively increasing slope across routes 49, 8, and 45 does not seem to impact the mileage much at all.

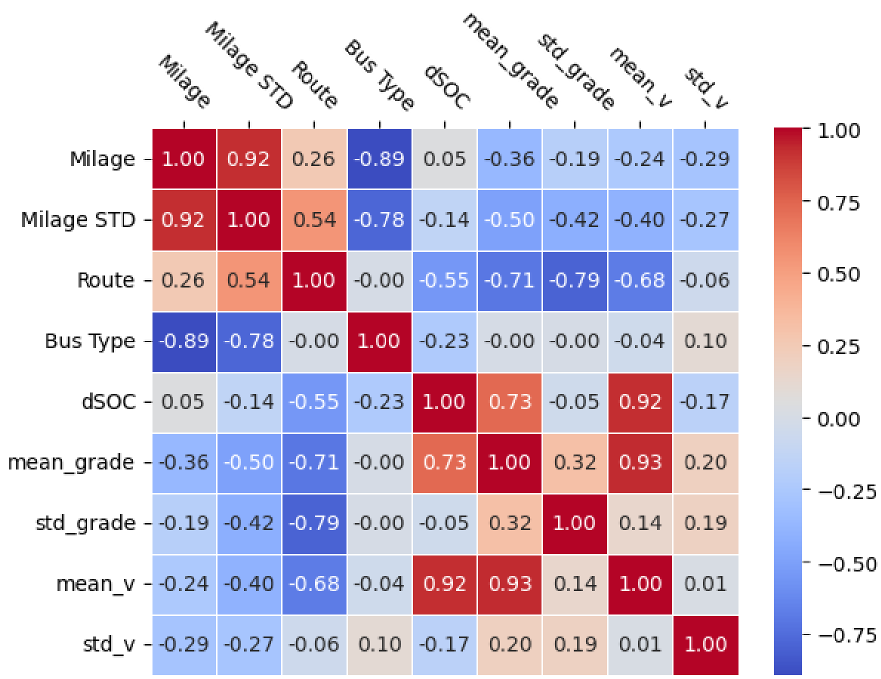


Figure 3 .2: *Pairwise correlation matrix of summarized results*

Through the use of pairwise correlations, we can create a correlation matrix between the simulation results.

$$pairwise\ correlation(X, Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \text{ (eqn. 3.1)}$$

This takes the covariance of the two variable's populations, and divides it by the standard deviations of each population, σ_X and σ_Y. From this, it allows us to identify how close to a linear trend two variables have. By no means is this an absolute representation of how the variables are actually related, but it can help us understand general trends in these results, and serves to re-validate the model further. For example, the bus model size has a significant inverse relationship with mileage. In other words, as the vehicle size increases, the mileage decreases. This aligns with expectations of larger vehicles requiring more energy to move. Likewise, the mean grade of a given route is inversely correlated with mileage - as is the mean and standard deviation of velocity. The route traveled on has a not-insignificant correlation with the mileage, though the proportion and value is not based in any numerical reality given the route number is not a linear scale.

Examining the variation in mileage for a given trip simulation: as the bus model scales up in size, the mileage becomes less varied. The higher mass of these models may indicate that the external forces have less of an impact on how much energy the vehicle needs to move. We see this reflected in grade variation – The XE60's mileage variation does not change significantly with grade variation within a route, whereas the lighter XE40 and XE35 do see some increased variation. Interestingly, the mileage variation has stronger inverse correlations with speed than speed variation, running counter to the overall mileage's trend. In other words, as the overall velocity of a trip increases, the mileage variability will decrease. We can also see the impact that grade has – as the grade increases, there's less variation in the vehicle mileage. This could, hypothetically, be due to the motor reaching its maximum power more often to meet the required external forces of the higher grade.

The change in state of charge for a given trip ultimately depends on the size of the ESS and the distance of the trip, but it is worth noting that the loss in state of charge has strong correlations with increasing overall grade and overall velocity.

While we can observe trends via correlation in the results, the sensitivity analysis can provide insights into what parameters of the model and trip conditions are influencing the mileage the most.

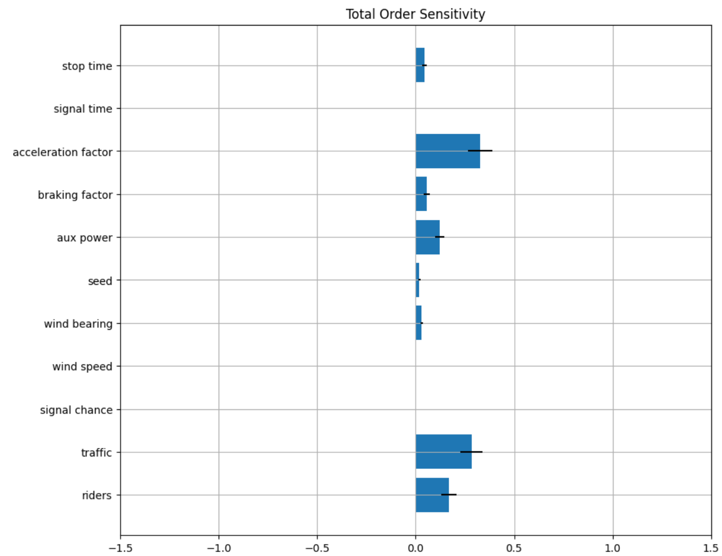


Figure 3 . 3*: Total order sensitivity for Route 45.*

Figure 3.3 illustrates that the variables with the most impact on mileage are those that directly affect top speed of the bus. Acceleration factor f_a and traffic volume, $\%_{traffic}$ are different sides of the same coin, with the acceleration factor determining how much higher over the speed limit the bus is willing to go, and the traffic artificially reducing that speed limit. It follows, then, that when the bus is accelerating 'more', the amount of energy used will be more. This is also reflected in n_{riders} being particularly impactful – the number of riders will both affect number of stops and cumulative mass, which means that with higher numbers of riders the bus will be accelerating and decelerating more, with higher required forces.

Route	1st Most Sensitive	2nd Most Sensitive	3rd Most Sensitive
40	f_a	$\%_{traffic}$	n_{riders}
70	f_a	f_{br}	$\%_{traffic}$
8	f_a	$\%_{traffic}$	P_{aux}
49	f_a	$\%_{traffic}$	f_{br}

Table 3.5: A summary of the most impactful variables on ESS mileage for the XE60 on each of the selected routes.

From looking at the other 3 routes, braking factor stands out as the highest impact – the more aggressive a driver brakes, the less time there is to regenerate power as that maximum regeneration load is hit quicker. P_{aux}, being the auxiliary power load from things like HVAC and subsystems, will have a direct impact on the bus' ESS load. Granted, this is not particularly relevant to driving conditions, and rather serves to reinforce the importance of environmental conditions and temperature on EV efficacy.

Figures for Total Order, First Order, and Second Order Sensitivity can be found in section 3.5.1. By using a Rint model of a fresh $LiFePO_4$ (LFP) cell, the model is also capable of creating cumulative distributions of proportional time spent at a cell voltage. While the implementation may have changed in the New Flyer full EV's, the cell and module potential is used to inform maintenance and module replacements. On the hybrid systems, the modules were intended to be replaced should their potential spend 10 progressive seconds above 3.96 or below 0.5 volts.

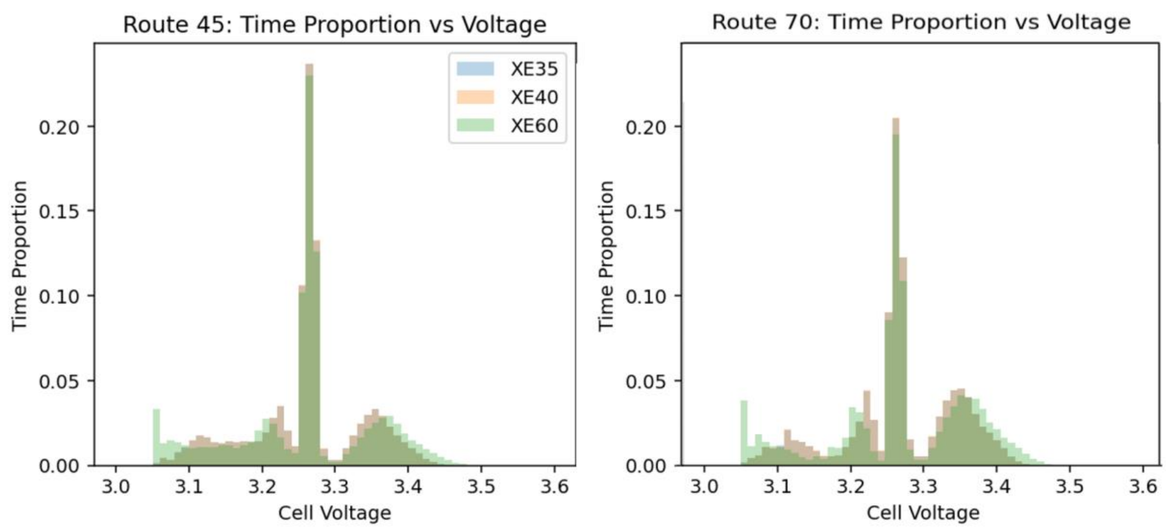


Figure 3 . 4: *Comparison of Voltage Distributions for Route 45 & 70*

Considering the fact that the cell model is not aged, neither route 45 nor route 70 have voltages that approach the cutoff points. For route 45, we can see that there's very little difference between the XE35 and XE40, likely stemming from their identical ESS designs. The XE60 follows the same general trends as the other two models, but the potential has shifted further away from the nominal voltage of 3.3, instead spending more time in both higher and lower voltages. While using regenerative braking, the cell potential is much more normally distributed around ~3.35 V. This normal distribution lies in stark contrast to the cell voltage during discharge, which is more evenly distributed on a plateau between 3.2 and 3.05 volts. The charging and discharging overpotential behavior differences may be caused by the less-than-smooth acceleration distribution, as was seen in figure 2.17.

Route 70, on the other hand, is not as evenly distributed when discharging. The higher voltage peak at around 3.35 V again corresponds to overpotential from regenerative braking, and has a distinct separation from the nominal cell voltage. It's clear that more time is spent braking in route 70, and less time idling due to the lower peak. The plateau of discharge voltage seen in route 45 is not present for route 70. Route 70 is predominantly a gradual downhill route, with some variability along the way. Route 45, by comparison, has less grade variability but has nearly triple the overall grade change, however slight. So, it may be the case that these peaks visible for route 70 are artefacts of the acceleration profile used, or, these peaks may be more pronounced as the uphills and downhills of route 70 are slightly steeper than 45.

3.4: Conclusions

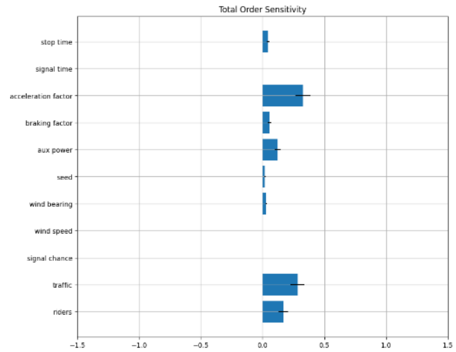
Even with the rudimentary approximations for driver 'personality', this behavior has significant impact on the amount of energy needed for the vehicle to perform a given route. The more 'aggressive', the more energy is going to be used while starting and less energy will be regained when stopping. For the high level of start-stop variability inherent to city bus driving and the influence of traffic, this would mean that the capacity needed for any route is liable to be significantly larger than manufacturer testing would indicate. Likewise, this set of simulations and analysis have shown that grade and vehicle mass likewise have outsized impacts on the route's mileage – conclusions that align with the existing literature. We can also clearly see that the landscape has impacts on cell-level behavior, with the potential distribution when discharging being markedly disparate depending on which route the vehicle was driven on.

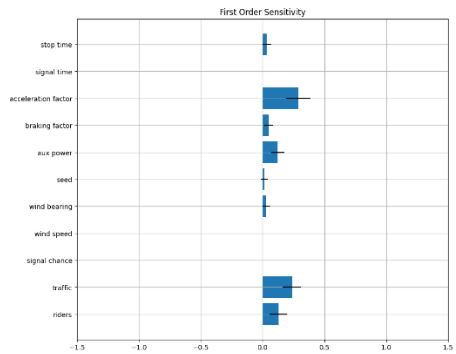
All of the conclusions included here, whether backed by previous work or offering new insights are inherently limited by the scope of the analysis. This was limited to 4 outbound routes, with the purpose of ensuring that these four specific trips were understood and their limitations clearly defined. However, with more time, using reRoute_Dynamics has the potential to offer further insight into transit impacts. Running this simulation across each permutation of the 147 King County Metro routes, with an increased parameter space and number of iterations would allow for even deeper, more confident insights into what affects the vehicle mileage. This could also extend to enabling predictions of how often a route would be able to be driven before a battery needed to be recharged. Additionally, collaboration with King County Metro would enable the comparison of predicted loads, capacity use, and velocity profiles to more real-world data. Finally, in the realm of more direct energy-storage research: Testing the predicted loads on market-available cells would allow for increased insights into the dis-aggregation of inherent cell aging dynamics and what is simply caused by a route's particularities.

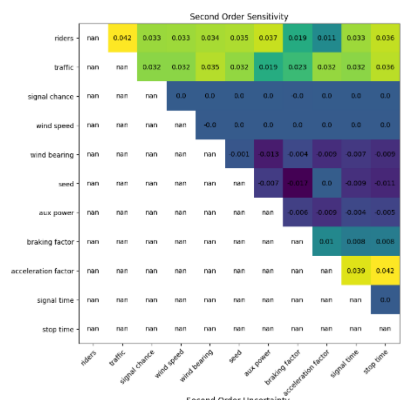
3.5: Appendix

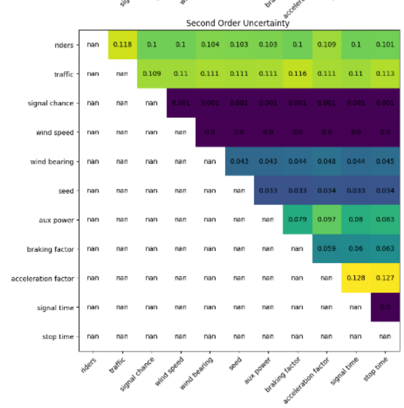
3.5.1: Sensitivity Analysis Supplementary Figures

Route 45

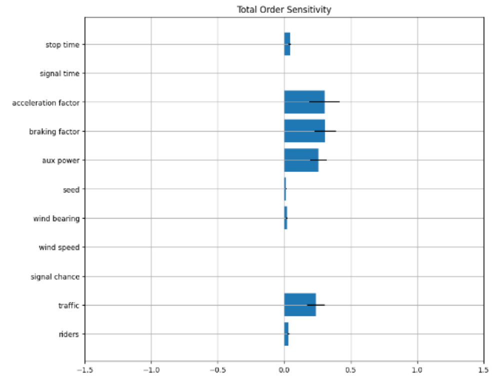


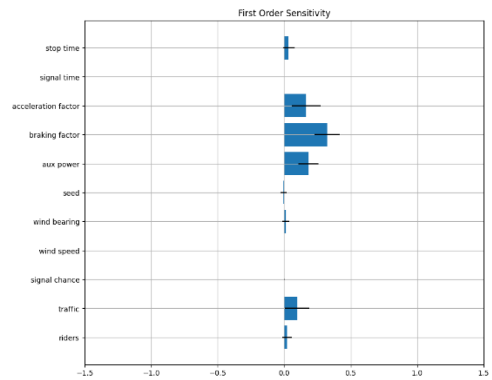


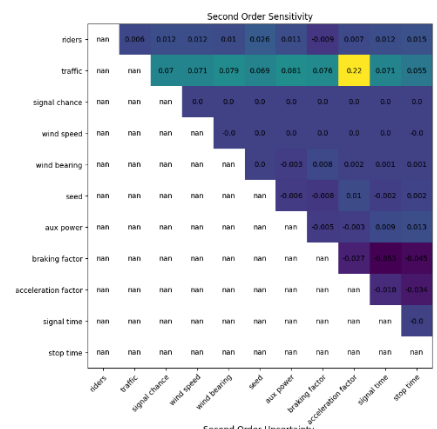


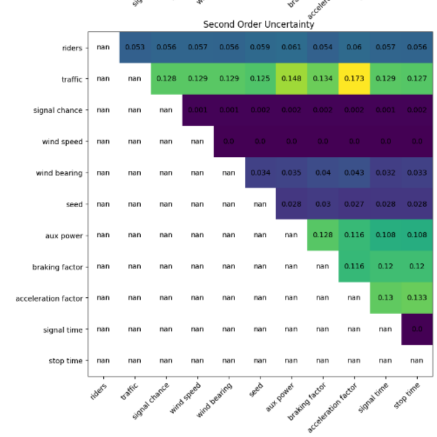


Route 70

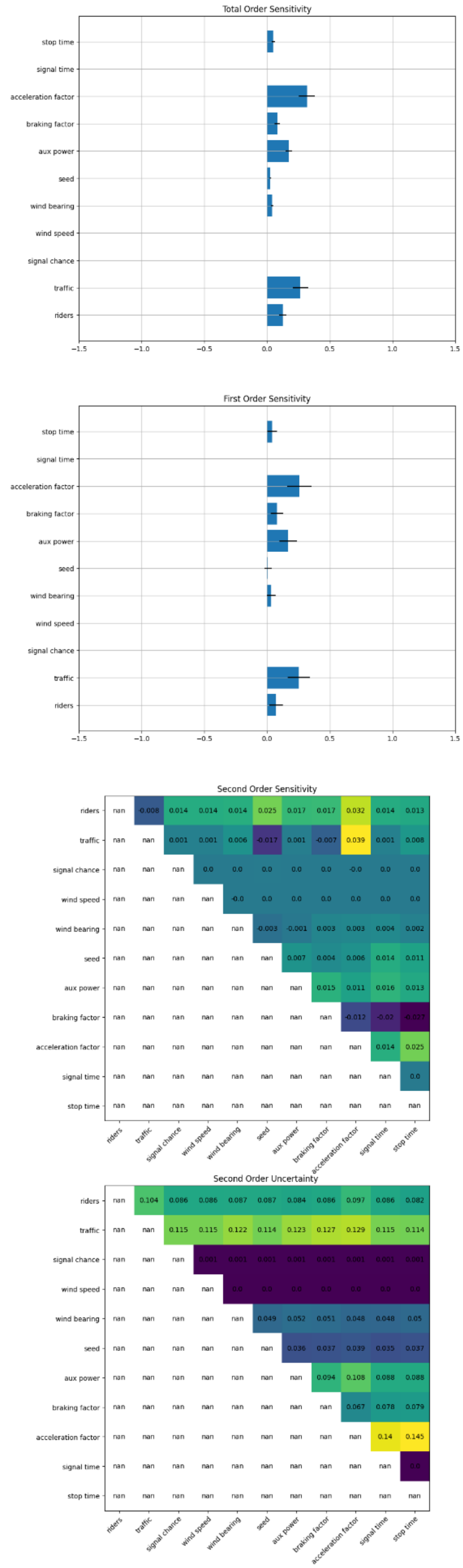

Total Order Sensitivity
stop time
signal time
acceleration factor
braking factor
aux power
seed
wind bearing
wind speed
signal chance
traffic
riders
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


First Order Sensitivity
stop time
signal time
acceleration factor
braking factor
aux power
seed
wind bearing
wind speed
signal chance
traffic
riders
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

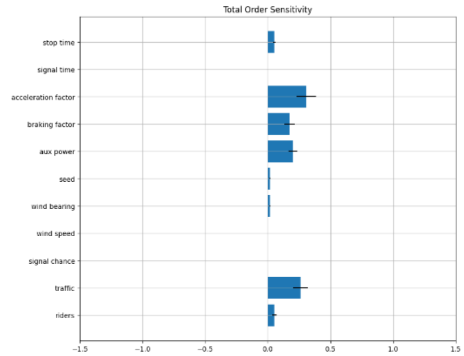

Second Order Sensitivity
riders
traffic
signal chance
wind speed
wind bearing
seed
aux power
braking factor
acceleration factor
signal time
stop time

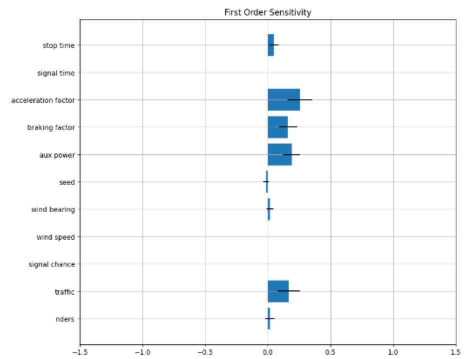

Second Order Uncertainty
riders
traffic
signal chance
wind speed
wind bearing
seed
aux power
braking factor
acceleration factor
signal time
stop time

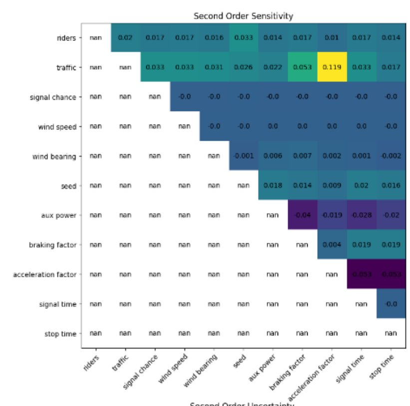
Route 8

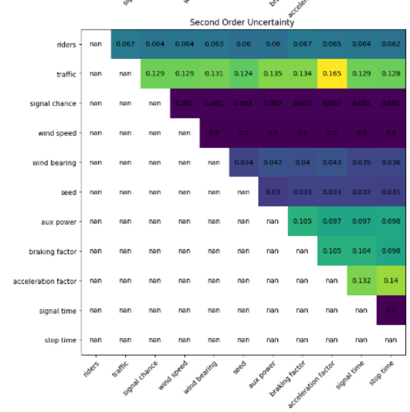

Total Order Sensitivity
stop time
signal time
acceleration factor
braking factor
aux power
seed
wind bearing
wind speed
signal chance
traffic
riders
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5
First Order Sensitivity
stop time
signal time
acceleration factor
braking factor
aux power
seed
wind bearing
wind speed
signal chance
traffic
riders
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5
Second Order Sensitivity
riders
traffic
signal chance
wind speed
wind bearing
seed
aux power
braking factor
acceleration factor
signal time
stop time
Second Order Uncertainty
riders
traffic
signal chance
wind speed
wind bearing
seed
aux power
braking factor
acceleration factor
signal time
stop time

Route 49


Total Order Sensitivity
stop time
signal time
acceleration factor
braking factor
aux power
seed
wind bearing
wind speed
signal chance
traffic
riders
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


First Order Sensitivity
stop time
signal time
acceleration factor
braking factor
aux power
seed
wind bearing
wind speed
signal chance
traffic
riders
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Second Order Sensitivity
riders
traffic
signal chance
wind speed
wind bearing
seed
aux power
braking factor
acceleration factor
signal time
stop time


Second Order Uncertainty
riders
traffic
signal chance
wind speed
wind bearing
seed
aux power
braking factor
acceleration factor
signal time
stop time

Chapter 4 - Conclusion

4.1: Summary

This work is focused on the generation of realistic models of Battery Electric Vehicles from core physical dynamics. Being able to generate these quickly, accurately, and using publicly available data will lower the barrier for development and research of transit EV infrastructure. To this end, the work had two primary goals. The first was to develop a toolkit that enables users to create realistic and highly customizable models of EV drive cycles. The second was to simultaneously demonstrate possible use cases and their value as well as provide preliminary insights into EV electrification.

Chapter 2 illustrated that, through the use of Longitudinal Dynamic Models and modular python scripting, it is possible to generate realistic bus drive cycles. These drive cycles are demonstrably accurate to the true runtimes of existing bus routes in King County after which they were modeled – and, more importantly, are able to capture up to 94% of the variation in a trip's acceleration. While an individual model produced by reRoute_Dynamics may not inherently be accurate to reality, the software is designed in such a way that users are easily able to modify and customize models to capture trends for a given route. The entirety of reRoute_Dynamics is available on GitHub [43] via the MIT License. The repository also provides a number of tutorials and guides on its use that allow potential users to explore the ways they would be able to use the tools.

Chapter 3 extended the use of this tool on the measured routes' models to encompass a wider range of possible conditions, and found that the factors that affect grade, velocity, mass, and stopping frequency have outsized impact the mileage of Battery Electric Busses. These conclusions were validated by prior work in literature and further serve to demonstrate the efficacy of the tool. Additionally, this chapter also illustrated the importance of considering the route's needs more carefully rather than relying solely on manufacturer quoted mileage, as well as provided examples of ways transit organizations would be able to get preliminary answers to their driving questions. Through an expansion of the modeling done in this chapter, it is reasonable to conclude that further modeling will enable deeper, more accurate insights into how vehicles and their energy storage systems are affected by the sheer variety in a route that cannot be captured by one-size-fits-all tests.

4.2: Outlook

With the current landscape for federal public infrastructure funding grants becoming more pessimistic, there are increasingly more barriers to entry for transit organizations to begin doing the research they need to support sustainability goals. While EV technology has consistently improved over the last several decades, and especially over the previous few years, transit adoption has remained a consistently laborious and time-intensive process. While pilot programs are always going to be a critical step in the development of these new technologies at any site, the ability to easily model a route's needs can significantly augment the development process at any stage. Through providing preliminary estimates of energy needs to inform vehicle purchasing decisions, supporting the planning process for infrastructure and maintenance, and beyond, reRoute_Dynamics is capable of reducing the amount of unknowns for agencies looking to electrify. And, in being entirely open-source, it can help reduce the barrier to entry for smaller counties or transit organizations. This toolkit enables augmented decision-making processes that can better inform and forward the sustainability, economic, and humanitarian goals of EV transit programs not just in Washington, but around the world.

References

[1] "Inventory of U.S. Greenhouse Gas Emissions and Sinks." Accessed: Jun. 01, 2025. [Online]. Available: https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks

[2] "2025 Incremental Purchase Cost Methodology and Results for Clean Vehicles," *US Dep. Energy*, Jan. 2025, Accessed: Jun. 01, 2025. [Online]. Available: https://www.energy.gov/sites/default/files/2025-01/2025.01.13_DOE_Incremental_Cost_Report_for_publication.pdf

[3] C. Johnson, E. Nobler, L. Eudy, and M. Jeffers, "Financial Analysis of Battery Electric Transit Buses." National Renewable Energy Lab, Jun. 2020. Accessed: Jun. 01, 2025. [Online]. Available: https://afdc.energy.gov/files/u/publication/financial_analysis_be_transit_buses.pdf

[4] "Global EV Outlook 2019." IEA, May 2019. Accessed: Jun. 01, 2025. [Online]. Available: https://afdc.energy.gov/files/u/publication/financial_analysis_be_transit_buses.pdf

[5] "Zero Emissions by 2035." King County Metro, Jun. 2025. Accessed: Jun. 01, 2025. [Online]. Available: https://kingcounty.gov/en/dept/metro/programs-and-projects/zero-emissions

[6] "Battery and Fuel Cell Electric Buses At California Transit Agencies with Over 100 Buses." California Air Resources Board, Apr. 2018. Accessed: Jun. 01, 2025. [Online]. Available: https://ww2.arb.ca.gov/sites/default/files/2020-06/1805toptransitzebs.pdf

[7] "40 CFR 600.116-12 -- Special procedures related to electric vehicles and hybrid electric vehicles." Code of Federal Regulations, May 29, 2025. Accessed: Jun. 01, 2025. [Online]. Available: https://www.ecfr.gov/current/title-40/part-600/section-600.116-12

[8] "Federal Transit Bus Test Performed for the Federal Transit Administration U.S. DOT In accordance with 49 CFR Part 665." Pennsylvania Transportation Institute, Oct. 2017. Accessed: Jun. 01, 2025. [Online]. Available: https://www.altoonabustest.psu.edu/assets/docs/bus%20docs/updated%20procedures/Energy-Economy1.pdf

[9] H. W. Lee, L. Osburn, and B. Treece, "Charging Forward - Evaluating Public-Private Partnerships for Electric Bus Base Conversion to Support a Zero Emission Fleet." Mobility Innovation Center, University of Washington, 2023. Accessed: Jun. 01, 2025. [Online]. Available: https://mic.comotion.uw.edu/wp-content/uploads/2023/11/Charging-Forward_FINAL-Nov-2023_Bart-Treece.pdf

[10] L. Eudy and M. Jeffers, "Zero-Emission Bus Evaluation Results: King County Metro Battery Electric Buses." National Renewable Energy Lab, Feb. 2018. Accessed: Jun. 01, 2025. [Online]. Available: https://www.transit.dot.gov/sites/fta.dot.gov/files/docs/research-innovation/115086/zero-emission-bus-evaluation-results-king-county-metro-battery-electric-buses-fta-report-no-0118.pdf

[11] "King County Metro Electric Bus Summary." King County Metro, 2024. Accessed: Jun. 01, 2025. [Online]. Available: https://cdn.kingcounty.gov/-/media/king-county/depts/metro/documents/projects/zero-emissions-fleet/king-county-metro-bebs-summary.pdf?rev=6e1118f465924dc789460aee1f33b680&hash=A75D37F52A1F100EFB4678571003EABF

[12] M. Gallet, T. Massier, and T. Hamacher, "Estimation of the energy demand of electric buses based on real-world data for large-scale public transport networks," *Appl. Energy*, vol. 230, pp. 344–356, Nov. 2018, doi: 10.1016/j.apenergy.2018.08.086.

[13] J. Asamer, A. Graser, B. Heilmann, and M. Ruthmair, "Sensitivity analysis for energy demand estimation of electric vehicles," *Transp. Res. Part Transp. Environ.*, vol. 46, pp. 182–199, Jul. 2016, doi: 10.1016/j.trd.2016.03.017.

[14] A. Franca, "Electricity consumption and battery lifespan estimation for transit electric buses: drivetrain simulations and electrochemical modeling," M.S., University of Victoria, 2015. Accessed: Jun. 01, 2025. [Online]. Available: https://dspace.library.uvic.ca/server/api/core/bitstreams/2ef38798-5d53-4416-a30a-81b8b7456da4/content

[15] A. Saadon Al-Ogaili *et al.*, "Estimation of the energy consumption of battery driven electric buses by integrating digital elevation and longitudinal dynamic models: Malaysia as a case study," *Appl. Energy*, vol. 280, p. 115873, Dec. 2020, doi: 10.1016/j.apenergy.2020.115873.

[16] I. Crawford, Y. Shao-Horn, and D. Keith, "How much CO2 is emitted by manufacturing batteries?," MIT Climate Portal. Accessed: Jun. 01, 2025. [Online]. Available: https://dspace.library.uvic.ca/server/api/core/bitstreams/2ef38798-5d53-4416-a30a-81b8b7456da4/conten

[17] E. Eggleton, *EricaEgg/Route_Dynamics*. (Apr. 12, 2024). Jupyter Notebook. Accessed: Jun. 01, 2025. [Online]. Available: https://github.com/EricaEgg/Route_Dynamics

[18] E. E. Eggleton and D. T. Schwartz, "Incorporating GIS Data into a Battery Performance Model for Electric and Hybrid-Electric Buses," *ECS Meet. Abstr.*, vol. MA2019-01, no. 22, p. 1155, May 2019, doi: 10.1149/MA2019-01/22/1155.

[19] E. E. Eggleton, R. Carlin, H. Goldwyn, and D. T. Schwartz, "Route_Dynamics: An Open-Source Python Package for Estimating Electrified Vehicle Energy," presented at the 236th ECS Meeting (October 13-17, 2019), ECS, Oct. 2019. Accessed: Jun. 01, 2025. [Online]. Available: https://ecs.confex.com/ecs/236/meetingapp.cgi/Paper/130317

[20] E. Eggleton and D. T. Schwartz, "Physics-Based Use Case Analysis: Route Planning and Dynamic Load Prediction for Battery-Electric Buses," *ECS Meet. Abstr.*, vol. MA2022-01, no. 1, p. 149, Jul. 2022, doi: 10.1149/MA2022-011149mtgabs.

[21] "Operator's Guide Xcelsior Diesel-Electric 35FT. Transit Bus." New Flyer, Oct. 07, 2014. Accessed: Jun. 01, 2025. [Online]. Available: http://projects.seattletransitmap.com/library/Coach%20Operator's%20Manuals/XDE35%20revA%201761,%201851.pdf

[22] F. Adegbohun, A. von Jouanne, B. Phillips, E. Agamloh, and A. Yokochi, "High Performance Electric Vehicle Powertrain Modeling, Simulation and Validation," *Energies*, vol. 14, no. 5, Art. no. 5, Jan. 2021, doi: 10.3390/en14051493.

[23] A. Industries, "Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates." Accessed: Jun. 01, 2025. [Online]. Available: https://www.adafruit.com/product/746

[24] "phyphox - Your smartphone is a mobile lab.," phyphox. Accessed: Jun. 01, 2025. [Online]. Available: https://phyphox.org

[25] "GPS Accuracy." NOAA, Mar. 03, 2022. Accessed: Jun. 01, 2025. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/

[26] "King County GTFS Feed." Jun. 01, 2025. Accessed: Jun. 01, 2025. [Online]. Available: https://kingcounty.gov/en/dept/metro/rider-tools/mobile-and-web-apps

[27] "Speed Limit Map | Seattle GeoData." Accessed: Jun. 01, 2025. [Online]. Available: https://web.archive.org/web/20240302013158/https://data-seattlecitygis.opendata.arcgis.com/maps/SeattleCityGIS::speed-limit-map/explore

[28] “Traffic Signal Assemblies.” Accessed: Jun. 01, 2025. [Online]. Available: https://data-seattlecitygis.opendata.arcgis.com/datasets/SeattleCityGIS::traffic-signal-assemblies/explore?location=47.628082,-122.306805,11.27

[29] “Washington State Department of Natural Resures Lidar Portal.” Washington Lidar Portal. Accessed: Jun. 01, 2025. [Online]. Available: https://lidarportal.dnr.wa.gov/#47.63636:-122.11905:13

[30] “National Geodetic Survey.” NOAA. Accessed: Jun. 01, 2025. [Online]. Available: https://geodesy.noaa.gov/datasheets/index.shtml

[31] *rasterio: Fast and direct raster I/O for use with Numpy and SciPy*. C, Cython, Python. Accessed: Jun. 01, 2025. [Online]. Available: https://github.com/rasterio/rasterio

[32] “Grading.” City of Seattle, Jan. 06, 2017. Accessed: Jun. 01, 2025. [Online]. Available: https://streetsillustrated.seattle.gov/design-standards/roadway-construction/grading/

[33] “Traffic signals | WSDOT.” Accessed: Jun. 01, 2025. [Online]. Available: https://wsdot.wa.gov/travel/operations-services/traffic-signals

[34] “Seattle Climate, Weather By Month, Average Temperature (Washington, United States) - Weather Spark.” Accessed: Jun. 01, 2025. [Online]. Available: https://weatherspark.com/y/913/Average-Weather-in-Seattle-Washington-United-States-Year-Round

[35] “NF Xcelsior Brochure.” New Flyer, Jan. 2021. Accessed: Jun. 01, 2025. [Online]. Available: https://www.newflyer.com/site-content/uploads/2021/01/NF-Xcelsior-Brochure.pdf

[36] “Xcelsior Trolley Electric Bus.” New Flyer, Dec. 2024. Accessed: Jun. 01, 2025. [Online]. Available: https://www.newflyer.com/site-content/uploads/2024/12/Xcelsior-Trolley-Electric_Brochure.pdf

[37] T. D. Gillespie, *Fundamentals of vehicle dynamics*. Warrendale (Pa.): Society of automotive engineers, 1992.

[38] “Transit Bus In-Service Brake System Performance Testing - APTA BTS-BC-RP-001-05, Rev. 1.” American Public Transportation Association, Feb. 04, 2020. Accessed: Jun. 01, 2025. [Online]. Available: https://www.apta.com/wp-content/uploads/APTA-BTS-BC-RP-001-05_Rev1.pdf

[39] “MoComp Traction motors from Siemens Mobility,” Siemens Mobility Global. Accessed: Jun. 01, 2025. [Online]. Available: https://www.mobility.siemens.com/us/en/portfolio/rolling-stock-components/traction-motors.html

[40] E. Mohamed, M. Radhwi, and A. Abdel Gawad, “Computational investigation of aerodynamic characteristics and drag reduction of a bus model,” *Am. J. Aerosp. Eng.*, vol. 2, pp. 64–73, Jan. 2015, doi: 10.11648/j.ajae.s.20150201.16.

[41] “Drive Cycle Analysis Tool — DriveCAT| Transportation and Mobility Research | NREL.” Accessed: Jun. 01, 2025. [Online]. Available: https://www.nrel.gov/transportation/drive-cycle-tool/

[42] “Schedules and maps - King County, Washington.” Accessed: Jun. 01, 2025. [Online]. Available: https://www.kingcounty.gov/en/dept/metro/routes-and-service/schedules-and-maps

[43] S. Peck, *sebpeck/reRoute_Dynamics*. (May 14, 2025). Jupyter Notebook. Accessed: Jun. 01, 2025. [Online]. Available: https://github.com/sebpeck/reRoute_Dynamics