A Study on the Impact of Driver Behavior on the Energy Consumption of Electric Vehicles in a Virtual Traffic Environment

by

Rodolfo Rodriguez

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Electrical Engineering)
In the University of Michigan-Dearborn
2020

Master's Thesis Committee:

Assistant Professor Youngki Kim, Co-Chair
Assistant Professor Alireza Mohammadi, Co-Chair
Assistant Professor Junho Hong

Acknowledgements

First and foremost, I’d like to give thanks to the Bill and Melinda Gates Scholarship Foundation for giving me the opportunity to pursue my master’s degree at University of Michigan-Dearborn. Without their financial support I would not have been able to go as far as I have in my education and in my engineering career.

I would also like to thank Professor Kim for giving me this research opportunity and guiding me along the way. He has given me a lot of insight into the academic world and a different perspective on how to approach problems and how to solve them. He has also pushed me to perform far beyond what I would have expected out of myself and has shown me what am I capable of.

Lastly, I’d like to thank my family and my girlfriend for their support throughout this endeavor. They’ve helped me stay focused on this study and have made sure that I’m always at a good mental state throughout the tough and stressful times when working on this thesis.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

Mild driving, i.e., accelerating slowly and smoothly, braking less frequently, and increasing spacing between vehicles to avoid harsh braking, has been shown to be effective in the improvement of fuel economy, especially for conventional vehicles with an internal combustion engine. For electric vehicles (EVs), it is implied that the energy consumption can also be improved by driving less aggressively. However, the extent in which the driver behavior reduces the energy consumption of an EV in various traffic environments has not been fully explored. A simulated environment can create a greater variety of driving cycles and conditions, thereby providing more insight as to how driving aggressiveness affects the vehicle's energy consumption.

The objective of this study is to evaluate the impact of the driving behavior on the energy consumption a battery electric vehicle (BEV) under various traffic scenarios. To simulate the driver behavior, a driver model is typically required to replicate the behavior of a human driver while traversing a given route (e.g., maintaining a safe distance from the preceding vehicle). Various driver models can be found in literature. Among these, the widely used Intelligent Driver Model is chosen in this study to characterize the different levels of driver aggressiveness. To that end, the microscopic traffic simulator, PTV Vissim, is used to simulate various realistic traffic environments in which a human driver's behavior can be evaluated. The co-simulation of the PTV Vissim Component Object Model (COM) interface in conjunction with MATLAB allows the energy consumption performance on an EV to be determined for various levels of driving aggressiveness. The results obtained from the co-simulation with a virtual traffic environment are

compared to those from single-lane car-following scenarios created using EPA (Environmental Protection Agency) standardized driving schedules.

The results of the single-lane car-following scenario shows that there is a slight increase (<1.5%) in energy usage per kilometer by changing from a mild driving style to an aggressive driving style. For the city driving cycle created in Vissim, aggressive driving can lead to a 6.6% decrease in the average energy usage per kilometer driven than mild driving if it allows the vehicle to avoid red traffic signals and general vehicle traffic. However, driving at medium-level aggression is not quick enough to avoid these obstacles and consequently increases the average energy usage per kilometer by 1.1% over mild driving. For the highway driving cycle, the benefits of driving milder can be realized, as switching from aggressive to mild driving results in a 3.4% decrease in average energy usage per kilometer. The results of these driving tests demonstrate that the level of driving aggressiveness cannot be fixed and should instead adapt to the traffic environment in order to maximize the battery life and range of an EV.

Chapter 1: Introduction

1.1 Background and Motivation

In an effort to reduce greenhouse gas (GHG) emissions and the impact vehicles have on global warming, battery electric vehicles (BEVs) have been growing alternative to internal combustion engine (ICE) vehicles owing to their lack of vehicle emissions. According to a study done by the Edison Electric Institute [1], the number of EVs sales, which includes BEVs and plug-in hybrid electric vehicles (PHEVs), has increased from 200,000 in 2017 to 361,000 in 2018 in the U.S. Since 2011, the number of EVs in the U.S. has continuously increased, reaching to more than 1.18 million EVs on the road in the U.S. as of March 2019.

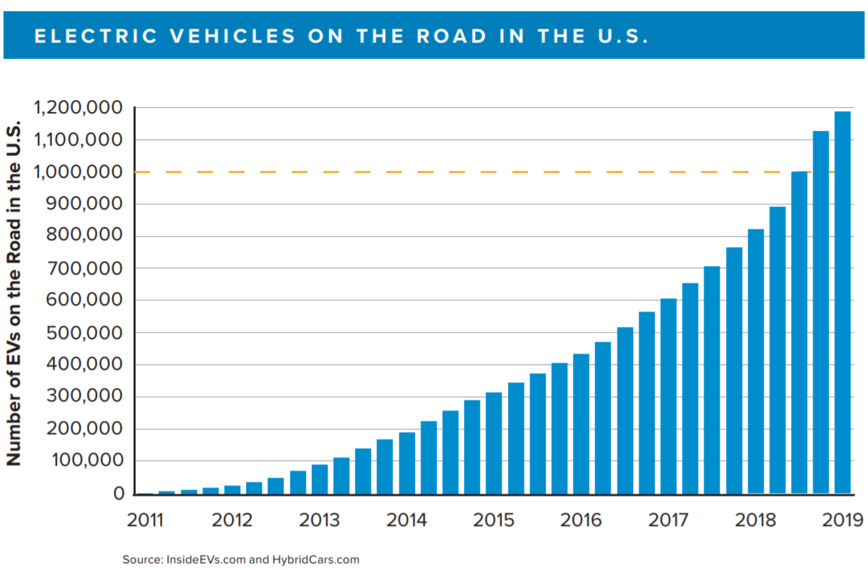


Figure 1: Electric vehicles on the road in the U.S. from 2011 to 2019 [1]

Although BEVs do not directly generate any greenhouse gases, they can generate indirect emissions based on their source of electricity. As a result, it is significant to reduce the overall energy consumption of BEVs in order to reduce the indirect production of GHGs and their impact to the electrical grid.

One key method of reducing the energy usage of a BEV is to adjust the driver behavior. Generally, reducing the acceleration rate, braking less frequently, maintaining a large gap between vehicles, and reducing the vehicle speed can improve the energy usage of the vehicle. [2] Since electric vehicles are capable of regenerative braking, some of the energy can be recuperated from braking, and thereby reducing the effect of driving more aggressively. While there have been several studies done on the impact of the driver behavior on the fuel economy of ICE vehicles, there are few that have studied the impact of BEVs.

1.2 Literature Review

Several studies that have investigated the impact of driving behavior on a vehicle's energy consumption. Bingham et. al. [3] investigated the effect of driving behavior on BEVs energy consumption and range using data collected from real-world driving performed by one driver along predefined routes in rural parts of the United Kingdom. The findings from this study showed that there was a 30% increase in energy consumption over the specified driving cycle by changing from moderate driving to more aggressive driving.

Since this study simulates the driver behavior using the Intelligent Driver Model (IDM), other studies were reviewed for their usage of the model. The authors in [4] had developed a modified version of the IDM called the Enhanced Driver Model (EDM), which adds a tuning parameter that is based on driver aggressiveness, as well as an algorithm to detect traffic signals and stop signs, so the model can react to more driving scenarios. The driver model was evaluated on a mild-hybrid vehicle model over real-world driving data to show that the EDM can capture the behavior of human drivers. In previous work [5], the IDM was used in a simulation study to evaluate the energy consumption of a BEV with an integrated thermal management system model. The model was tested under different levels of driver aggressiveness in a car-following scenario for 3 different EPA driving cycles.

The authors in [6] combined MATLAB/Simulink, the IDM, PTV Vissim, and CarSim to evaluate the fuel economy of an ICE vehicle for different levels of driver aggressiveness and traffic density over three driving cycles based on real roads in Columbus, Ohio and Ann Arbor, Michigan. The IDM was modified in this study to be capable of stopping for traffic lights and stop signs.

1.3 Contribution from this Study

The work done in this paper makes several contributions on top of the work done in [6]. The main contribution is studying the impact of the driving behavior on an electric vehicle instead of an ICE vehicle like in [6]. While [3] uses real-world results to show how much aggressive driving impacts the energy consumption of an electric vehicle, having a simulation environment would allow for more variety of electric vehicles, driving behaviors, and traffic environments to be tested. Additionally, it would allow for the test to be repeatable across different vehicles, without having to drive each vehicle, which costs time and resources. In [3], one driver and a fixed driving cycle based in the United Kingdom are considered, so the results may be different on roads in the U.S., or for U.S. drivers.

This study also modifies the IDM and adds an object detection algorithm for the vehicle to respond to other vehicles, traffic signals, and speed limits set in Vissim. A separate free-flow acceleration model based on the IDM is also used for situations where there are no objects in front the IDM-based vehicle. The levels of driver aggressiveness studied here also take into consideration the time headway as a variable since more aggressive drivers leave shorter gaps between vehicles. The time headway was not considered in [6] as a part of the levels of driving aggressiveness. Additionally, the vehicle is allowed to change lanes based on the driver model used in Vissim. Lane changing was also omitted in [6].

Lastly, this study aims to provide more detail on the processes involved in combining the driver model, vehicle model, and traffic environment all into one simulation using MATLAB and PTV Vissim, as it is briefly presented in [6]. This study will make this energy consumption evaluation simulation more accessible to the autonomous and connected vehicle and electrified vehicle research community.

1.4 Objectives and Organization

The chapters in this paper break up the combined model into several individual models, which describe the driver behavior, the vehicle powertrain model, and the traffic environment simulation. Chapter 2 discusses existing driver models and why the Intelligent Driver Model (IDM) was selected for this study. It also discusses how the IDM functions and how different levels of driver aggressiveness were set using its parameters. Lastly, the objection detection algorithm and its integration in the Vissim traffic environment are described.

Chapter 3 explains why an electric vehicle was chosen for this study and the significance of electric vehicles in reducing emissions. Moreover, this chapter presents the vehicle's specifications, the powertrain model, and how the energy consumption is determined.

Chapter 4 discusses the two driving cycles in which the driver model and vehicle will be evaluated. The first section uses three driving cycles from the Environmental Protection Agency (EPA) to simulate several car-following scenarios as a basis for comparison. The second section shows two driving cycles created in Vissim, one city-based and one highway-based, and the simulation set-up for both scenarios.

Chapter 5 analyzes the results of 100 simulation runs of each of these driving cycles and evaluates the energy consumption for each level of driver aggressiveness. The car-following driving cycles using the EPA driving cycles are compared to the driving cycles created in Vissim where the vehicle is capable of changing lanes and interacting with the objects in Vissim. Lastly, Chapter 6 summarizes the results of this study and discusses several directions for future work.

Chapter 2: Driver Model

2.1 Modeling Driver Behavior

One of the key factors in solving today's traffic congestion problem is understanding how people drive. Human driving behavior places constraints on how engineers design roads and highways, and thereby how traffic is regulated. To better understand this phenomena, researchers and engineers have developed mathematical models that represent human driving in order to simulate real-world traffic scenarios. These models are known as microscopic models since they describe the behavior of a single vehicle rather than the general traffic flow behavior of many vehicles.

In traffic simulation, the simplest driver models are usually car-following models, in which one vehicle is following another vehicle from a certain distance. Typically, these driver models assume that the vehicles are in the same lane and that the speed of the following vehicle is continuously adjusted to maintain a safe distance from the leading vehicle in order to avoid collision. Based on how the parameters are set in the models, the vehicles can follow more or less aggressively. More advanced models include features for changing lanes and making decisions such as deciding which direction to take at an intersection. However, since the focus is on the longitudinal driving behavior, which dominates the energy consumption of driving, only single-lane car-following acceleration models will be studied.

2.2 Literature Review of Driver Models

In the field of traffic flow dynamics, several driver models have been developed to describe the dynamics of traffic flow. According to [7], some of the earliest car-following models were created by Reuschel in 1950 [8] and Pipes in 1953 [9]. These models established the concept that the safety distance, which is the minimum bumper-to-bumper distance to the vehicle in front, should be proportional to the speed of the vehicle. Another early model to use this concept was the Optimal Velocity Model (OVM), which is a continuous-time car-following model that calculates the acceleration based on the difference between the actual speed and the optimal speed over the adaption time. [7] The Newell car-following model also applied this concept but had made it applicable to discrete time and had introduced the desired speed as an additional parameter. This created a free-flow region, so the vehicle was not restricted to only car-following scenarios. [10] The problem with some of these earlier models was that they were sometimes difficult to interpret because of their abstract parameters. The Gipps model [11] used more tangible parameters, such as the vehicle maximum acceleration, the most severe braking rate, and the desired speed, which could be more clearly defined. The author in [7] states that Gipps also ends up producing more realistic results than the Newell in highway scenarios. The Krauß model in [12] expands upon the Gipps model to gain further understanding of different types of congestion found in traffic flow. Other models, such as the Das and Asundi model, focused more on traffic density and the overall vehicle flow, which are more analogous to fluid dynamics [13]. Lastly, the Intelligent Driver Model [14] simplifies the parameters for microscopic vehicle dynamics and provides a robust model that functions well in many traffic applications. The reasons for focusing on this driver model will be discussed in the following section.

2.3 Intelligent Driver Model

2.3.1 Overview of Intelligent Driver Model

The Intelligent Driver Model (IDM) was first introduced in [14] as a microscopic single-lane car-following driver model to simulate congested traffic on Germany freeways. The IDM was proposed in this paper as a solution that avoids the problems with existing driver models. To summarize the shortcomings with other models according to the authors, the goal was to create a simple model that features few, straightforward parameters, asymmetric acceleration and braking, produces no accidents, has realistic driver behavior for acceleration and braking, accounts for the driver's desired speed, and can function in a deterministic approach. For this study, other driver models, such as the Gipps model [11], the Krauß model [12], and the Das and Asundi model [13] were reviewed and compared against the IDM. According the authors in [14], the Gipps and Krauß models do not function as well at the deterministic limit, which is where the driving behaviors can be studied for a given driver model. The Das and Asundi is better suited for applications where the speed and density of traffic flow are being studied and would not work well for this study since the focus is on a single vehicle.

Lastly, the IDM has been applied in various other literature and has been shown to be responsive to modification. In [15], the IDM was used to create a hypothetical lead vehicle driving cycle for an autonomous vehicle to follow for fuel economy evaluation. In [4], the IDM was improved upon by adjusting the braking dynamics, so it adapts more realistically to a greater variety of traffic conditions. The IDM was also modified to react to stop signs and traffic signals and implemented in the microscopic traffic simulation software PTV Vissim for fuel economy testing in [6]. In addition, [7] further explores the functionality and mechanics of the IDM, including how to add more human-like behavior to the model and how it responds to various traffic scenarios. As a result of its widespread application, the IDM is chosen as a basis for modeling human driving behavior.

2.3.2 Mechanics of Intelligent Driver Model

Before discussing the mechanics of the IDM, there is some terminology that must be defined to understand how the vehicles are positioned in space. In the given single-lane car-following scenario, the vehicle α using IDM follows the preceding vehicle α-1 down a road. The position, speed, and acceleration at each time step t for each vehicle are defined in Figure 1.

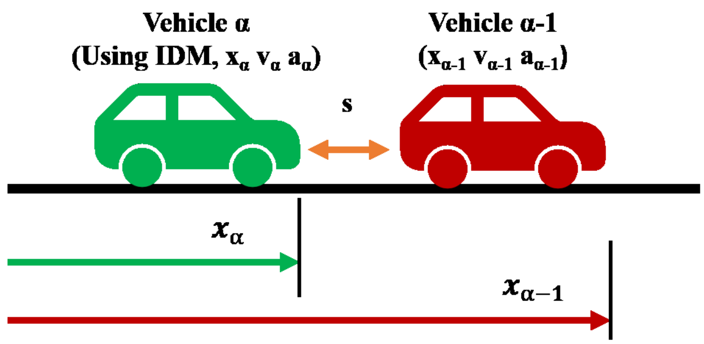


Figure 2: Overview of single-lane car-following scenario

The position of each vehicle is measured at the front bumper. The gap s is the distance between the rear bumper of the leading vehicle and the front bumper of the following vehicle at any time, shown in Eq. (1). The gap s takes into consideration the length of the preceding vehicle $l_{\alpha-1}$.

$$s = x_{\alpha-1} - x_\alpha - l_{\alpha-1} \tag{1}$$

The IDM is based on two equations, one for the acceleration of the IDM vehicle and one for the desired safe gap that the IDM vehicle wants to be at from the preceding vehicle. In a car-following scenario, the IDM determines the acceleration needed for the vehicle to maintain a safe distance from the preceding vehicle based on the desired safe gap $s^*(v_\alpha, \Delta v)$. The acceleration $a_\alpha(t)$ of the IDM vehicle is described by Eq. (2) and it is a function of the desired safe gap s^* between the two vehicles computed from Eq. (3) as follows:

$$a_\alpha = \frac{dv_\alpha}{dt} = a_{\alpha,max}\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha,\Delta v)}{s}\right)^2\right] \tag{2}$$

$$s^*(v_\alpha, \Delta v) = s_0 + s_1\sqrt{\frac{v_\alpha}{v_0}} + v_\alpha T + \frac{v_\alpha \Delta v}{2\sqrt{a_{\alpha,max} b_\alpha}} \tag{3}$$

The acceleration of the following vehicle is divided into two parts. The first part determines the acceleration of the vehicle to its desired speed v_0, which is shown in Eq. (4).

$$\frac{dv_\alpha}{dt} = a_{\alpha,max}\left(1 - \left(\frac{v_\alpha}{v_0}\right)^\delta\right) \tag{4}$$

The acceleration exponent δ in the acceleration controls the rate at which the acceleration is reduced as the vehicle speed approaches v_0. Most simulations set $\delta = 4$ as a standard value such as in [4], [14], and [16]. When accelerating from a complete stop, the vehicle will accelerate at its maximum acceleration and then taper off as it approaches its desired speed. While this may seem unrealistic, this is only done during the first time-step, and all subsequent acceleration values are continually reduced, resulting in a gradual increase in speed.

The latter part of Eq. (2) is the intelligent braking model based on the desired safety gap and the gap s between the vehicles:

$$\frac{dv_\alpha}{dt} = -a_{\alpha,max}\left(\frac{s^*(v_\alpha \Delta v)}{s}\right)^2 \tag{5}$$

The braking is primarily dependent on how the desired safe gap compares to the actual gap between the vehicles. The desired safe gap shown in Eq. (3) is also split into two parts, one for conditions where both vehicle speeds are the same (such as when stopped or cruising at constant speed), and when the vehicle speeds are different. When the IDM vehicle approaches a stop sign or stopped vehicle, the term s_0, as known as the jam distance, defines the minimum gap between the IDM vehicle and the object in front of it. [14] This is the only term not dependent on the vehicle speed and will affect the spacing of vehicles during traffic jams. The other jam distance s_1 is an additional gap that increases nonlinearly as the vehicle approaches its desired speed. The maximum value of this term is achieved when the vehicle reaches its desired speed. However, due to its nonlinear behavior, the term s_1 is typically set to zero. [14]

The time headway T is defined as the time that it takes for the front bumper of the IDM vehicle to reach the front bumper of the preceding vehicle. Essentially this is the time gap between the two vehicles plus the time for a vehicle to move a whole vehicle length l. It is analogous to the distance headway being the distance between the front bumpers of the two vehicles. Lastly, the final term of s^* is dependent on the approach speed Δv, which is defined as $\Delta v = v_\alpha - v_{\alpha-1}$. In cases where the speed of the IDM vehicle is greater than that of the preceding vehicle, Δv will be positive, thereby increasing s^* and causing the IDM vehicle to brake to avoid collision. On the other hand, when the IDM vehicle is slower than the preceding vehicle, Δv is negative and therefore the s^* decreases, as there is more distance between vehicles. As a result, the IDM vehicle will accelerate to reduce the gap.

The braking is also affected by the comfortable braking term b. The authors in [7] studied the behavior of this term in isolation by removing the acceleration component and the equilibrium terms $s_0 + vT$. When the IDM vehicle is approaching a stop light or stopped vehicle, the preceding object speed $v_{\alpha-1}$ is zero and the distance between the IDM vehicle and the preceding object is s. Therefore, the difference in speeds Δv is equal to the IDM vehicle speed v_α, and results in the following:

$$a_\alpha = -\left(\frac{v_\alpha^2}{2s}\right)^2 \frac{1}{b} = -\frac{b_{kin}^2}{b} \tag{6}$$

Where the kinematic deceleration b_{kin} is defined as:

$$b_{kin} = \frac{v_\alpha^2}{2s} \tag{7}$$

In critical situations where b_{kin} is greater than b, the IDM vehicle will exceed b_{kin} to avoid collision with the preceding vehicle. This behavior of the IDM makes it difficult to estimate the maximum braking acceleration of the vehicle since it can exceed the value of b. In non-critical situations where b_{kin} is less than b, the deceleration is less than b_{kin}. Generally, the IDM will attempt to equalize the comfortable deceleration and kinematic deceleration [7].

To illustrate this effect, the following figure shows the effect of different values of b when a vehicle is approaching a stop light. The stop light is positioned 60m away and the vehicle has an initial speed of 54 km/hr. At t = 0s, the light turns red. The plot shows that setting the value for b too low causes the vehicle to initially overestimate the braking effort needed. For this scenario, b_{kin} is 1.875 m/s^2. Although b = 2 m/s^2 is greater than b_{kin}, b_{kin} is re-evaluated at every time step, and therefore the required braking effort is changing at every time step. This is why the value of b is not a hard limit and can be exceeded based on the driving conditions.

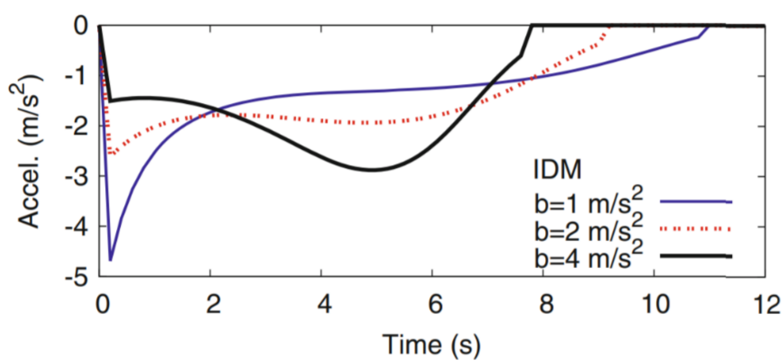


Figure 3: IDM vehicle braking for red light 60m away for different values of comfortable deceleration

2.3.3 Intelligent Driver Model Parameter Selection for Driving Behaviors

From the kinematic equations, the speed and position of the following vehicle can be found at every time step based on the computed acceleration. Based on how the parameters are set for the IDM, the vehicle behavior can be tuned to be more aggressive or more relaxed. By studying the effect of each parameter, one can create driving profiles that describe various levels of driver aggressiveness or mildness. Various sources are considered to determine how the IDM parameters should be set. Table 1 summarizes how different literature sets the parameters for the IDM.

Parameters	Source			
	Highway Driving [7]	City Driving [7]	Typical Vehicle [4]	Vehicle using IDM [17]
Desired speed v_0 [kph]	120	54	120	108
Maximum acceleration a_{max} [m/s^2]	1	1	0.73	1.4
Acceleration exponent δ	4	4	4	4
Comfortable deceleration b [m/s^2]	1.5	1.5	1.67	2
Time headway T [s]	1	1	1.6	1.5
Minimum gap s_0 [m]	2	2	2	2
Vehicle length l_{veh} [m]	5	5	N/A	4.5

Table 1: Literature review of typical IDM parameter values

For the studies in the table, a typical vehicle has a maximum acceleration of 0.73-1.4 m/s^2, a comfortable deceleration of 1.5-2 m/s^2, and a time headway of 1-1.6s. All of them uses the same value for δ and the s_0. To get a better perspective of how aggressive these values are, one can compare them to the acceleration and deceleration seen in the Environmental Protection Agency (EPA) driving schedules. The three driving schedules are the Urban Dynamometer Driving Schedule (UDDS) for city driving, the Highway Fuel Economy Driving Schedule (HWFET) for highway driving, and the US06 Supplemental Federal Test Procedure (US06) for aggressive driving. For each of these driving schedules, the average and maximum acceleration and deceleration are calculated and summarized in Table 2.

	Driving Schedule		
Parameter	EPA City (UDDS)	EPA Hwy (HWFET)	US06
a_{avg} [m/s^2]	0.5	0.19	0.67
a_{max} [m/s^2]	1.5	1.4	3.8
b_{avg} [m/s^2]	-0.58	-0.22	-0.73
b_{max} [m/s^2]	-1.5	-1.5	-3.1

Table 2: Speed, acceleration, and deceleration for EPA driving schedules

Comparing the information in Tables 1 and 2, one can see that the maximum acceleration values in Table 1 are lower than what is typically expected in both city and highway driving, and much lower than the maximum acceleration during aggressive driving. As a result, to meet the driving demands of most situations, the maximum acceleration of the IDM vehicle can be selected between about 1.5 m/s^2 to 4 m/s^2. As expected, higher maximum acceleration will result in more aggressive driving.

As previously discussed, the comfortable deceleration b does not represent the maximum deceleration the IDM vehicle experiences. However, the maximum deceleration should be considered for selecting b. While the average deceleration for the driving schedules are less than 1 m/s^2, the studies reviewed in Table 1 all set b to be at least 1.5 m/s^2 to avoid issues with the IDM overestimating the braking effort required. Higher values of b up to 3 m/s^2 should be considered in case of emergency scenarios where higher braking effort is needed or when decelerating from high speeds.

In Table 1, the time headway T ranges between 1s and 1.6s. According to [18], a 1s time headway is a reasonable minimum for heavy traffic scenarios, while 1-2s covers most emergency braking situations. For this study, the time headway will range from 1-3s because a higher value of T results in smoother, more efficient driving. While 3s may not always be feasible, it may cause the vehicle to drive slower to increase the gap between vehicle, thereby reducing the energy consumption.

Taking into consideration the values used in other works, the parameter sets for mild, medium, and aggressive driving are defined as follows:

Driving Behavior	**Maximum Acceleration *a* [m/s²]**	**Comfortable Deceleration *b* [m/s²]**	**Time Headway *T* [s]**	**Acceleration Exponent δ**	**Jam distance s_0 [m]**
Mild	1.5	1.5	3	4	2
Medium	2	2.5	2	4	2
Aggressive	3.5	3	1	4	2

Table 3: IDM parameters chosen for mild, medium, and aggressive driving styles

The values of the mild and aggressive category were chosen based on the maximum and minimum values of the parameters discussed, so the difference in behavior is more apparent. The maximum acceleration and deceleration of the aggressive driver is set to $3.5 m/s^2$ and $3 m/s^2$ because at the short time headway of 1s, higher acceleration and acceleration may result in oscillations and unstable behavior. The medium driving style falls somewhere in the middle, with slightly higher braking than the median of the mild and aggressive braking, and slight lower acceleration than the median of the maximum acceleration values of the other two styles. This was done to avoid the same issues as the aggressive driver given the time headway and prioritize braking over acceleration.

To verify the selection leads to the desired effect, a speed trace is made to visualize the driving behavior. A vehicle using the IDM is placed 2000m from a wall and attempts to accelerate up to its desired speed of 100kph. The IDM recognizes the wall and stops the vehicle to avoid collision. The figure confirms the selected parameters for each driving behavior result in higher acceleration, higher deceleration, and later braking as the driving behavior becomes more aggressive. Therefore, the selected parameters can reasonably represent the basic longitudinal dynamics of these three driving behaviors.

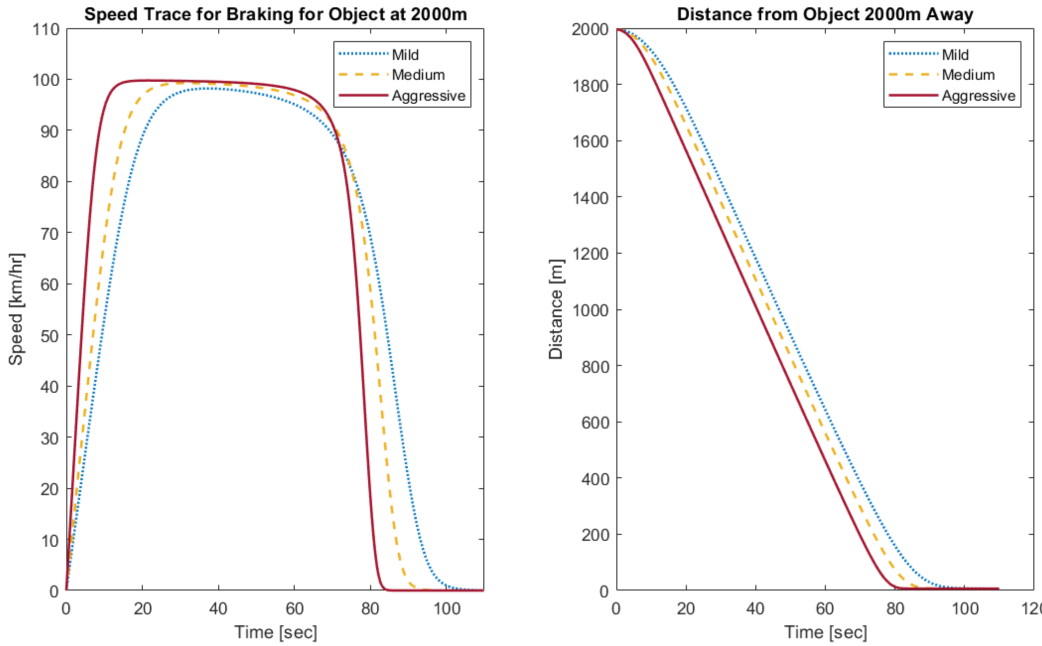


Figure 4: Speed traces for vehicle braking for object 2000m

2.3.4 Modifications to Intelligent Driver Model

To implement the IDM in MATLAB, some modifications were made to allow the change the equation to a discrete function. Equations (2) and (3) must be combined so that there is one equation for the acceleration at the current time step. The approach rate Δv can be substituted by v_α - $v_{\alpha-1}$, so that the desired safe gap s^* is only dependent on the speed of the IDM vehicle. By substituting s^* into the acceleration equation, one can define the acceleration of the IDM vehicle as a polynomial equation in terms of its speed. This results in the acceleration at the current time step i.

$$a_\alpha = \frac{dv}{dt} = k_0 + k_1 v_\alpha^\delta + k_2 v_\alpha^{0.5} + k_3 v_\alpha^1 + k_4 v_\alpha^{1.5} + k_5 v_\alpha^2 + k_6 v_\alpha^{2.5} + k_7 v_\alpha^3 + k_8 v_\alpha^4 \tag{8}$$

Where the terms k_0 through k_8 can be defined as the following.

$$k_0 = a_{max} - a_{max} \frac{s_0^2}{s^2} \tag{9}$$

$$k_1 = -\frac{a_{max}}{v_0^{\delta}} \tag{10}$$

$$k_2 = -\frac{2s_0 s_1 a_{max}}{v_0^{1/2} s^2} \tag{11}$$

$$k_3 = -\frac{2s_0 T a_{max}}{s^2} + \frac{s_0 v_{\alpha-1}\sqrt{a_{max}}}{s^2\sqrt{b}} - \frac{s_1^2 a_{max}}{s^2 v_0} \tag{12}$$

$$k_4 = -\frac{2s_1 T a_{max}}{v_0^{1/2} s^2} + \frac{s_1 v_{\alpha-1}\sqrt{a_{max}}}{s^2\sqrt{b v_0}} \tag{13}$$

$$k_5 = -\frac{s_0\sqrt{a_{max}}}{s^2\sqrt{b}} - \frac{T^2 a_{max}}{s^2} + \frac{T v_{\alpha-1}\sqrt{a_{max}}}{s^2\sqrt{b}} - \frac{v_{\alpha-1}^2}{4s^2 b} \tag{14}$$

$$k_6 = -\frac{s_1\sqrt{a_{max}}}{s^2\sqrt{b v_0}} \tag{15}$$

$$k_7 = -\frac{T\sqrt{a_{max}}}{s^2\sqrt{b}} + \frac{v_{\alpha-1}}{2s^2 b} \tag{16}$$

$$k_8 = -\frac{1}{4s^2 b} \tag{17}$$

Using the kinematic equations, the position and velocity are calculated for the next time step *i+1*.

$$v_\alpha(i+1) = v_\alpha(i) + a_\alpha(i)(t(i+1) - t(i)) \tag{18}$$

$$x_\alpha(i+1) = x_\alpha(i) + v_\alpha(i)t(i) + \frac{1}{2}a_\alpha(i)t(i)^2 \tag{19}$$

The IDM equation was modified to resolve an issue when the speed of the IDM vehicle and the leading vehicle are both zero. When both the speed of both vehicles is zero, Equation (2) and (3) should simplify to:

$$a_\alpha = a_{\alpha,max}\left[1 - \left(\frac{0}{v_0}\right)^{\delta} - \left(\frac{s_0}{s_0}\right)^2\right] = 0 \tag{20}$$

$$s^*(v_\alpha = 0, \Delta v = 0) = s_0 + s_1\sqrt{\frac{0}{v_0}} + 0 * T + \frac{0*0}{2\sqrt{a_{\alpha,max} b_\alpha}} = s_0 \tag{21}$$

However, if the IDM vehicle is not placed exactly s_0 distance away from the leading vehicle when both are stopped, then s^* is greater than s_0 and the IDM vehicle will accelerate at $a_{max}(1\text{-}s_0/s)$ to close the gap. This value tends to be very high and causes the vehicle to accelerate quickly and then braking harshly to avoid collision. This problem occurs because the IDM favors high initial acceleration when the vehicle speeds are different and then tapering off the acceleration to when sufficiently far from the leading the leading vehicle. To avoid this issue, a conditional statement is added to force the IDM acceleration to zero when both vehicles are stopped.

Since the acceleration is discrete, the sampling rate may not always be high enough to capture the small adjustments in acceleration needed for certain scenarios. In some cases, the IDM vehicle can also brake excessively and calculate a negative speed. During these cases, the vehicle is already coming to a stop, but had overestimated the amount of braking effort necessary. Therefore, a conditional statement can be added to force the speed at the next time step $v_\alpha(i+1)$ equal to zero. Given a sampling rate of t_s, the required deceleration to reach that speed from its current speed $v_\alpha(i)$ is simply

$$a_\alpha(i) = \frac{v_\alpha(i+1) - v_\alpha(i)}{t_s} = \frac{-v_\alpha(i)}{t_s} \tag{22}$$

Lastly, in some scenarios where there are no vehicles or objects in front of the IDM vehicle, the vehicle is in a free flow state. During this state, the acceleration is defined in [7] as the free flow acceleration $a_{free}(v_\alpha)$. The function is defined as follows:

$$a_{free}(v) = \begin{cases} a\left[1 - \left(\frac{v}{v_0}\right)^{\delta}\right] \; if \; v \leq v_0, \\ -b\left[1 - \left(\frac{v_0}{v}\right)^{\frac{a\delta}{b}}\right] \; if \; v > v_0 \end{cases} \tag{23}$$

The original creators of the IDM implement the free flow acceleration in [7] in an overall acceleration function that takes into consideration the gaps between the vehicle and other objects. Since the simulation will use a separate algorithm to detect the position of vehicles and objects, it is not necessary for the free flow acceleration function to consider the gap between vehicles. The authors in [7] use Equation (23) to develop the Improved Intelligent Driver Model (IIDM), which corrects some issues that occur when the IDM vehicle is traveling faster than the desired speed or following another vehicle near the desired speed. The authors state that the IIDM corrects some issues for city driving, but at the expense of different behavior during highway driving. Since most the driving done in these simulations are car-following or object-following scenarios, it was not necessary to fully implement the IIDM.

Compared to the EDM discussed in earlier found in [4], the IIDM keeps the base behavior of the IDM and adds some improvements in specific scenarios, whereas the EDM deviates more from the original IDM. The EDM breaks up the IDM based on the vehicle speed, adds another parameter to further tune the driver's aggressiveness, adds an equation for the desired speed as a function of aggressiveness, and adds an objection detection algorithm to give the vehicle the ability to react to traffic signals and stop signs. While the EDM contributes more depth for describing aggressive driving, the tunable parameter and equation it adds make it difficult to tie back the values to reality. Furthermore, it also removes the time headway and the desired gap equation, which help describe how aggressively the preceding vehicle is being followed. Therefore, the EDM was also not used for this study because it was more desirable to tune the driver aggressiveness based on the realistic parameters already used in the IDM.

2.4 Object Detection Model

For the driving cycles that are created in Vissim, there needs to be an objection detection model (ODM) in place so that the vehicle can respond to the traffic environment. This section provides an overview of the detection algorithm and explains how the data taken from Vissim is used to determine how the vehicle should react. A more detailed explanation of how the data stored in the objects in Vissim can be accessed and altered can be found in Appendix. Later chapters will discuss the traffic environment developed in Vissim in more detail.

Depending on the types of objects added to the traffic environment, the ODM requires that certain information be extracted from each object. For this application, only the information from other vehicles, the traffic signals, and the speed limits were needed. Data such as the object's position, the link number, the lane number, the speed limit value, the vehicle speed, and the traffic signal state were stored at each time step. The objects are separated based on what effect they have on the IDM vehicle. Speed limits only affect the desired speed of the vehicle, and only need to be recognized when in close proximity while traveling on the same road to update the desired speed. On the other hand, vehicles and traffic signals affect the actual speed of the vehicle and must be recognized even at far distances in order to allow the IDM vehicle sufficient room to react accordingly. Depending on what the object and its state is, there must be multiple options available to account for the variability of that object's behavior.

In order to determine how the IDM vehicle should react, the first thing to determine is the position of the critical objects in the simulations. In Vissim, the roads are referred to as links. Links can be connected together via other links called connectors to form intersections and roadways. The position of an object is always measured from the start of the link that it is placed on. When it drives another link, its position is reset and measured relative to the start of that new link.

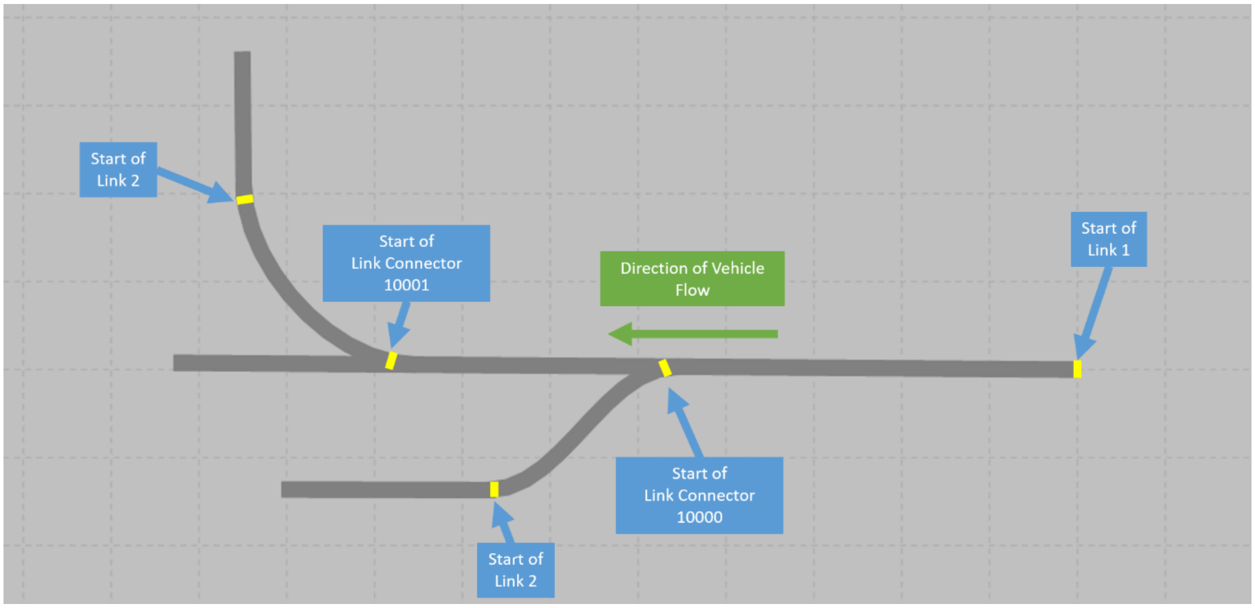


Figure 5: Starting position for links and connectors in Vissim

Therefore, it becomes challenging to recognize which objects on the subsequent link(s) are most important. While a vehicle set to specific vehicle route will show the next link in its sequence, there is the possibility that it cannot make it to that subsequent link if there is traffic preventing it from reaching that link. Additionally, creating vehicles routes for every possible combination of start and end points is inefficient. Therefore, to avoid the complexity associated with detecting objects in other links, the following measures are taken: 1. For all objects, the critical object will lie on the same link and same lane as the IDM vehicle. 2. Static objects such as speed limits and traffic signals will not be placed at the start of link. 3. Links merging onto other links will not be placed at the start of a link. 4. Use a vehicle route for the IDM that covers entire route to prevent it from going to the first connecting link it runs into. Using these design guidelines, the ODM can be simplified and a lot of potential errors can be avoided.

The decision tree for the speed limits outlines the checks needed to determine if the desired speed of the IDM vehicle should be updated.

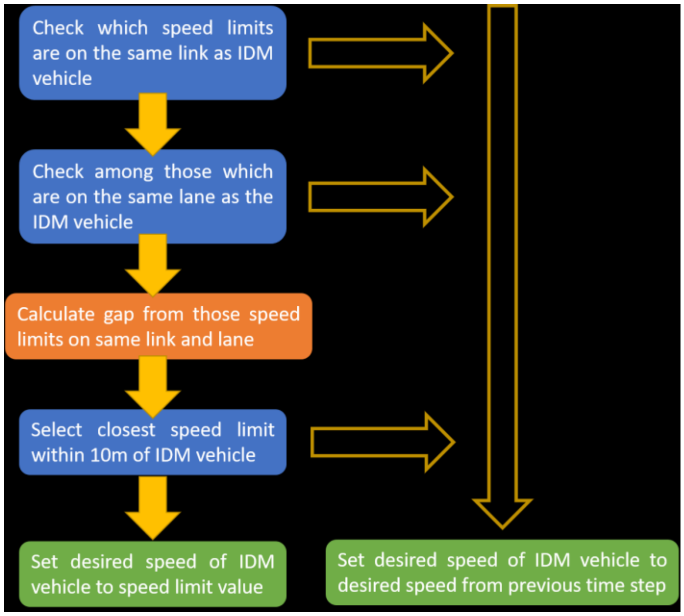


Figure 6: Speed limit detection decision tree

The algorithm sorts out the speed limits in the simulation for the ones that are on the same link and lane as the IDM vehicle. Once that group is found, the gap relative to the IDM vehicle is calculated for each speed limit in that group. The absolute value of each gap is taken to avoid choosing a large negative gap. The closest speed limit that is within 10m of the IDM vehicle is selected. The speed limit recognition distance is set to ±10m from the speed limit position because ensures that the IDM vehicle will travel over that 20m segment and recognize the speed limit. If the recognition distance is too small, or the IDM vehicle speed is too high, it could be possible that the next position that is calculated is past the speed limit recognition distance. With a 20m range and using a time step of 0.1s, the vehicle would have to travel more than 200m/s (720kph) to not recognize the speed limit, which is unrealistic but provides a more than sufficient margin. If a larger time step of 1s is used, then the range would have to be increased because any vehicle traveling more than 20m/s (72kph) could avoid it. While normally humans recognize speed limits that are far away from them, the other vehicles in Vissim only respond to a speed limit once they pass over its position, so setting the speed limit as proposed keeps the ODM consistent with other vehicles and prevents the IDM vehicle from speeding up in advance of the surrounding vehicles. Once the speed limit is detected, the desired speed of the vehicle is set according to the speed limit value. If any of these checks result in an empty vector, then the algorithm defaults to using the desired speed from the previous time step.

The algorithm for the objects that affect the vehicle's speed is slightly different, since the closest object needs to be identified first before the IDM can be set up properly. Below is the decision tree for objection detection.

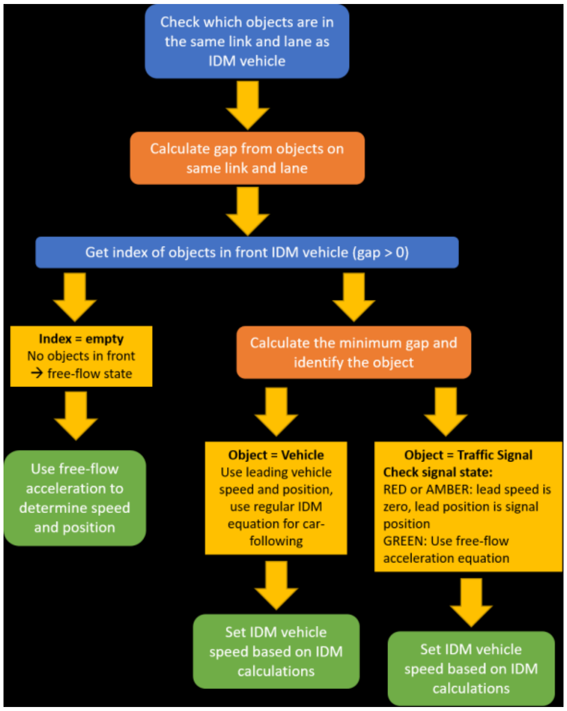


Figure 7: Object detection decision tree

Like the previous decision tree, first the objects that are in the same link and lane as the IDM vehicle are identified. Then the gap between those objects and the IDM vehicle is calculated. If there are no objects in front of the IDM vehicle, then this calculation would result in an empty vector. In that case, IDM vehicle is in a free-flow state and would use the free-flow acceleration equation to determine the acceleration needed to reach its desired speed. Otherwise, if the vector was not empty, then there are objects in front of the IDM vehicle, and the closest one would need to be identified. If the object is a vehicle, then the IDM equation can be used for following the vehicle as normal. However, if the object is a traffic signal, the behavior depends on the signal state. If the signal state is red or amber, the traffic signal should act as if there was a parked vehicle at the traffic signal position. In that case, the speed of the object is zero and the traffic signal position would be used in the IDM equation. Otherwise, if the light is green, the vehicle continues using the free-flow equation until it passes the traffic signal and identifies another vehicle.

Other objects such as reduced speed zones, conflict area, stop signs, pedestrian crossings, and yield signs could be added to the ODM but require a more complicated decision model. However, with this simplified algorithm one can develop the common city and highway driving scenarios to evaluate the performance of the driver model. The next section will discuss how the vehicle powertrain is designed so that the performance of the driver model can be measured.

Chapter 3: Battery Electric Vehicle Model

3.1 Importance of Battery Electric Vehicles

Due to the increased global demand from consumers, stricter emission regulations, and greater accessibility of charging networks, manufacturers are turning to building electric vehicles more than ever before. According to [19], the global market for EVs has increased about 60% each year from 2014-2018, with 2.1 million global sales in plug-in electric vehicles (PHEVs) and battery electric vehicles (BEVs) in 2018, with BEVs making up the greater percentage of sales. It is clear why consumers are interested in EVs, especially BEVs: BEVs offer a quieter driving experience, are less mechanically complex than internal combustion engine (ICE) vehicles, have instant power delivery, produce no tailpipe emissions, can partially recharge while braking, and can be recharged at home. There are still several major downsides to BEVs, such as low battery range, high battery cost, battery health degradation and slow charging speeds. However, despite the overall simplicity of the BEV platform, it is still not well understood how driving behavior affects their energy usage, especially considering that some energy can be regenerated when braking. While there are some BEVs that use up to 4 motors, only single-motor BEVs will be studied here.

3.2 Battery Electric Vehicle Powertrain Model

3.2.1 General Vehicle Specifications

The electric vehicle used in the simulation is a front wheel drive (FWD) car with some specifications based around the 2017 Chevrolet Bolt and 2014 Chevrolet Spark EV.

The battery is connected to a motor/generator unit, which transfers power to a differential and drives the front wheels.

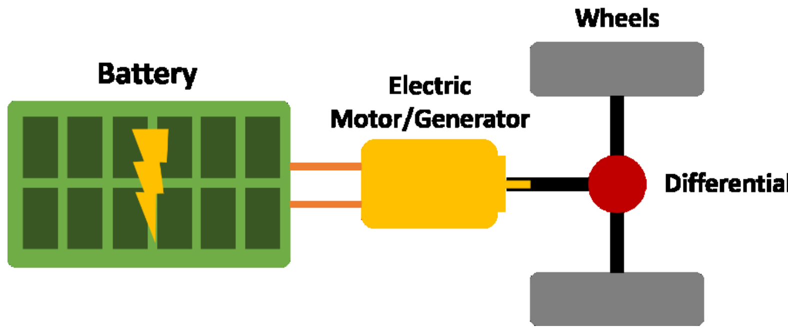


Figure 8: Electric vehicle powertrain model

The vehicle weight is set slightly higher than Bolt at 1633kg, whereas the drag coefficient is brought down to 0.30. The tire radius is 0.2921m, or 11.5inches, which is the same size as the Spark EV. The vehicle frontal area is based on multiplying the width and height of the Spark EV. The tires used on the vehicle have coefficient of rolling resistance of 0.009, since it is a typical value for Original Equipment Manufacturer (OEM) tire. [20] The vehicle specifications are summarized in the table below.

Specification	**IDM-based BEV**	**2017 Chevrolet Bolt**	**2014 Chevrolet Spark EV**
Weight	3600lbs (1633kg)	3580lbs (1625kg) [21]	2989lbs (1356kg) [22]
Drag Coefficient	0.30	0.308 [23]	0.326 [24]
Tire Radius	11.5in (0.2921m)	12.75in (0.3239m)	11.5in (0.2921m) [25]
Frontal Area	25.8ft^2 (2.4m^2)	N/A	2.5m^2 [22]
Coefficient of Rolling Resistance	0.009 [20]	N/A	N/A

Table 4: General vehicle specifications of simulated BEV compared to Chevrolet BEVs

3.2.2 Electric Motor Selection

The electric motor used in this vehicle is a Unique Mobility 100kW (peak) permanent magnet electric motor and inverter. The data for this motor is taken from the advanced vehicle simulator

Advisor, which is developed by the National Renewable Energy Laboratory (NREL). Although the motor chosen does not act as a generator, it is assumed that the motor can be used as a generator for regenerative braking, so the motor map is referenced to find the generator efficiency during regeneration. It is also assumed that during braking, regenerative braking is 100% efficient, so there are no losses due to friction braking. The electric motor is connected to a differential with a final drive ratio of 3.4:1 to centralize its operation around its wide torque range like the Spark EV.

Specification	**IDM-based BEV**	**2017 Chevrolet Bolt** [26]	**2014 Chevrolet Spark EV** [26]
Electric Motor Max Power	100kW	150kW	105kW
Electric Motor Max Torque	582N·m	360N·m	540Nm
Electric Motor Max Speed	4400RPM	8810RPM	4500RPM
Final Drive Ratio	3.4:1	7.05:1	3.17:1

Table 5: Electric motor specifications of simulated BEV compared to Chevrolet BEVs

3.2.3 Battery Modeling

The battery uses a simple open circuit voltage source and resistor model (also known as an OCV-R model).

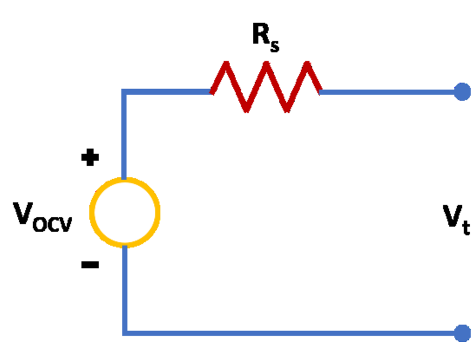


Figure 9: Battery electric circuit model

Since the focus of this study is primarily on the energy usage as a result of the driving behavior, the dynamics of the battery are kept simple and are only used to estimate how much energy was used. Normally the battery's open circuit voltage (V_{OCV}) can change depending on the SOC or ambient temperature, but for simplicity, the battery's nominal V_{OCV} is set to 450V and is assumed to be constant.

The battery shares some similarities to the Bolt. The battery capacity is set to 60kWh. Since the Bolt uses prismatic "nickel-rich lithium ion" battery cells, the battery cells used in this model are LiNiMnCo cells, based on the PL-7789182-2C battery cells found in [27]. The battery specifications are shown in the table below. Each cell has a V_{OCV} of 3.7V, an internal resistance of 8mΩ, and a capacity of 9Ah as stated by the manufacturer's specifications. Since the V_{OCV} for the battery is set to 450V, the battery cells specifications can be used to calculate the number of cells required and the battery pack voltage and resistance.

$$N_s = \frac{450V}{3.7V/cell} \approx 122\ cells \tag{24}$$

$$N_p = \frac{Q_{batt}}{V_{OCV}Q_{cell}} \approx 15\ cells \tag{25}$$

$$V_{OCV,pack} = V_{OCV}N_S = 451.4V \tag{26}$$

$$R_{pack} = \frac{R_{cell}N_s}{N_p} = 65.1m\Omega \tag{27}$$

The battery and battery cell specifications are summarized in the following table.

Specification	IDM-based BEV (PL-7789182-2C cell) [28]	2017 Chevrolet Bolt	2014 Chevrolet Spark EV
Battery Chemistry	LiNiMnCo	"Nickel rich lithium ion"	N/A
Battery Capacity Q_{pack}	60kWh	60kWh [21]	21.3kWh [22]
Battery Voltage (nominal) V_{OCV}	450V	350V	360V
Battery Pack Resistance R_{pack}	65.1mΩ	N/A	N/A
Battery Cell Capacity Q_{cell}	9Ah	N/A	N/A
Battery Cell Nominal V_{OCV}	3.7V	3.75V	N/A
Battery Cell Resistance R_{cell}	8mΩ	N/A	N/A
Number of Cells in Series N_s	122	96	N/A
Number of Cells in Parallel N_p	15	3	N/A

Table 6: Battery specifications of simulated BEV compared to Chevrolet BEVs

At each time step *t*, the battery power to meet the wheel demand is determined. The power drawn from the auxiliary components such as the HVAC is set to a constant value of 400W. Given the demanded battery power, the battery current at each time step is calculated as follows.

$$I = \frac{V_{OCV} - \sqrt{V_{OCV}^2 - 4R_{pack}P_{demand}}}{2R_{pack}} \tag{28}$$

Positive current denotes discharging, whereas negative current denotes charging. The state-of-charge (SOC) of the battery for the next time step is calculated from the amount charged or discharged from the battery.

$$SOC(t+1) = SOC(t) - \frac{It_s}{Q_{pack}} \tag{29}$$

3.2.4 Vehicle Model Implementation with Driver Model

The vehicle's energy management is seamlessly integrated with the IDM and the Vissim traffic environment since the energy usage can be calculated at each time step. Before running the simulation in Vissim, the EV's specifications and electric motor data are loaded and are used to determine the motor and battery specifications based on the EPA driving cycles. At each time step, Vissim's functions are used to get the position and speed of the IDM vehicle and the vehicle or object in front of it. The IDM determines the vehicle's acceleration a_α at the current time step in order to calculate its speed and position for the next time step. The total tractive force F_{tr} is calculated as follows to determine the required wheel torque and speed.

$$F_{tr} = ma_\alpha + F_a + F_g + F_{rr} \tag{30}$$

The vehicle is assumed to be on a flat surface, so there is no angle of inclination, and therefore the force due to gravity F_g is zero. The remaining force to aerodynamics and force due to the coefficient of rolling resistance C_{rr} can be written as shown.

$$F_{tr} = ma_\alpha + \frac{1}{2}\rho C_d A_f v_\alpha^2 + C_{rr}mg \tag{31}$$

With the radius of the wheel r_{wheel} known, the required wheel torque T_{wheel}, wheel speed ω_{wheel}, and wheel power P_{wheel} can be determined.

$$T_{wheel} = F_{tr}r_{wheel} \tag{32}$$

$$\omega_{wheel} = \frac{v_\alpha}{r_{wheel}} \tag{33}$$

$$P_{wheel} = T_{wheel}\omega_{wheel} = F_{tr}v_\alpha \tag{34}$$

The wheel demand can be used to determine the motor torque, speed and power. The motor power is first passed through the differential with a gear ratio GR_{mw} and motor to wheel efficiency of η_{mw} to reduce the speed and increase the torque as it goes from the motor to the wheels. All efficiency factors used are the ratio of the output over the input. The efficiency factor is used differently depending on whether the wheel power is positive, meaning energy is used, or negative, meaning energy is regenerated. Working backwards, the demanded motor output power $P_{motor,out}$ is calculated as shown below.

$$\omega_{motor} = \omega_{wheel} GR_{mw} \tag{35}$$

$$P_{motor,out} = \begin{cases} \frac{P_{wheel}}{\eta_{mw}}, & P_{wheel} > 0 \\ P_{wheel}\eta_{mw}, & P_{wheel} < 0 \\ 0, & P_{wheel} = 0 \end{cases} \tag{36}$$

$$T_{motor} = \frac{P_{motor,out}}{\omega_{motor}} \tag{37}$$

The motor speed and torque are used to determine the motor efficiency η_{motor} on the motor map. During braking regeneration, the absolute value of the motor torque is taken to be able to reference the corresponding efficiency value on the motor map. The input motor power $P_{motor,in}$ is found using the motor efficiency η_{motor} and the output motor power $P_{motor,out}$. The resultant battery power $P_{battery}$ is equal to the sum of the motor input power $P_{motor,in}$ and the constant auxiliary load P_{aux}, exception for cases where the vehicle is stopped when P_{wheel} is zero.

$$P_{motor,in} = \begin{cases} \frac{P_{motor,out}}{\eta_{motor}}, & P_{wheel} > 0 \\ P_{motor,out}\eta_{motor}, & P_{wheel} < 0 \\ 0, & P_{wheel} = 0 \end{cases} \tag{38}$$

$$P_{battery} = \begin{cases} P_{motor,in} + P_{aux}, & P_{wheel} < 0 \; or \; P_{wheel} > 0 \\ P_{aux}, & P_{wheel} = 0 \end{cases} \tag{39}$$

$$P_{demand} = P_{battery} \tag{40}$$

Based on the battery power used the change in SOC is calculated as discussed in the previous section and subsequently the SOC at the start of the next time step. The process is repeated for the new position and speed calculated from the IDM at the next time step. With the driver model and vehicle model combined, the combined model can be tested on different traffic simulations to evaluate the effect of the driving behavior on the vehicle's energy usage. The next section will discuss the driving cycles used for the simulation and how they were created. The driving cycles created using Vissim will be compared to those created using EPA driving schedules.

Chapter 4: Traffic Environment Simulation

4.1 Car-Following Driving Cycles Using EPA Driving Schedules

To determine the fuel economy and emissions produced by new vehicles, the Environmental Protection Agency (EPA) developed several standardized driving cycles to test the performance of each new vehicle. The first driving cycle created was the Urban Dynamometer Driving Schedule (UDDS), which comprised of 7.46mi route developed based on the LA 4 road. [29] The speed trace for the UDDS has an average speed of 19.6 mph and duration of 1372s. The UDDS was developed to capture the peak smog-producing driving conditions when driving in Los Angeles.

The EPA also developed other standardized driving cycles were created to capture other common driving scenarios. The Highway Fuel Economy Test (HWFET) replicates highway driving conditions under 60mph and the US06 Supplemental Federal Test Procedure (SFTP) captures aggressive driving due to its high acceleration and high maximum speed of 80mph. Other driving cycles have been developed for varying weather conditions, such as the SC03 SFTP, which captures city driving conditions in hot ambient temperatures with the air conditioning on, and an additional test cycle of the UDDS at cold ambient temperatures.

The EPA driving cycles are used in this study to compare with the driving cycles created in the Vissim. Since ambient temperature is not factored in this study, only the EPA UDDS, HWFET, and US06 driving cycles will be used. The vehicle using the IDM will follow another vehicle driving at the speeds indicated in each of the selected driving cycles.

To sufficiently deplete the battery, each driving cycle is repeated five times with a 5 second pause in between each cycle, as shown below for the HWFET cycle:

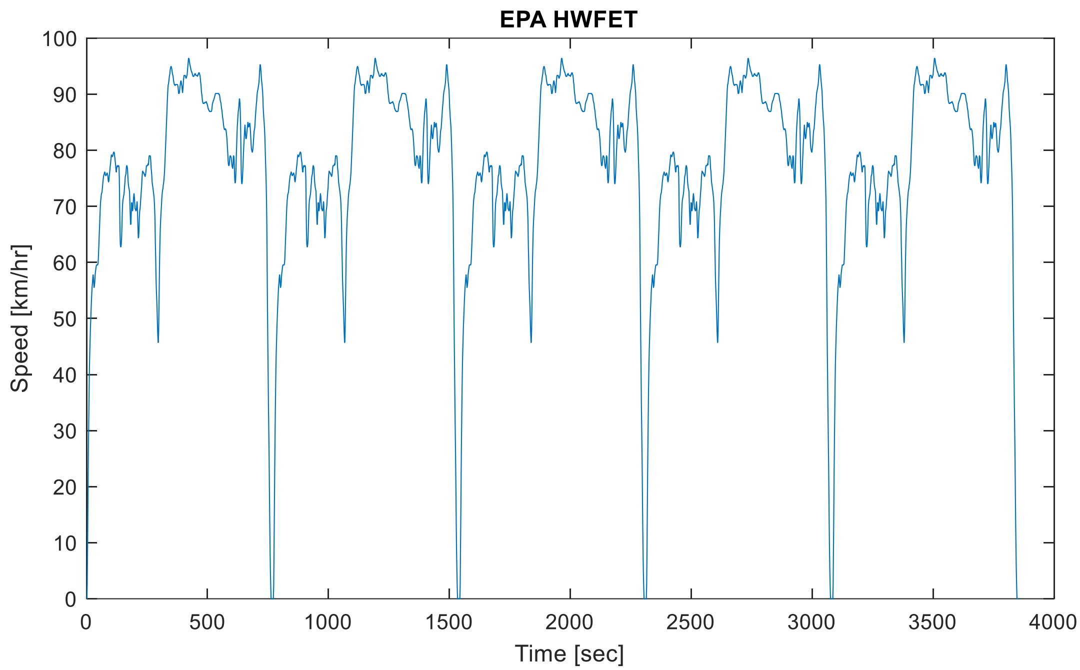


Figure 10: HWFET driving cycle repeated 5 times with 5 second pause in between

The IDM vehicle is evaluated under three different parameter sets that correspond to mild, medium, and aggressive driving. Each driving behavior is tested on all three driving cycles. At the end of 5 cycles, the results are divided by the distance traveled because the actual distance the IDM vehicle travels can vary based on the driving behavior.

4.2 Simulated Driving Cycles Using Vissim

4.2.1 City-Based Driving Cycle Using Vissim

The vehicle is also tested over two driving cycles created in Vissim. The first driving cycle represents city driving, where there are multiple traffic intersections in succession. The vehicle route is marked by the yellow arrow starting from the left, crossing five intersections, and ending at the road on the right.

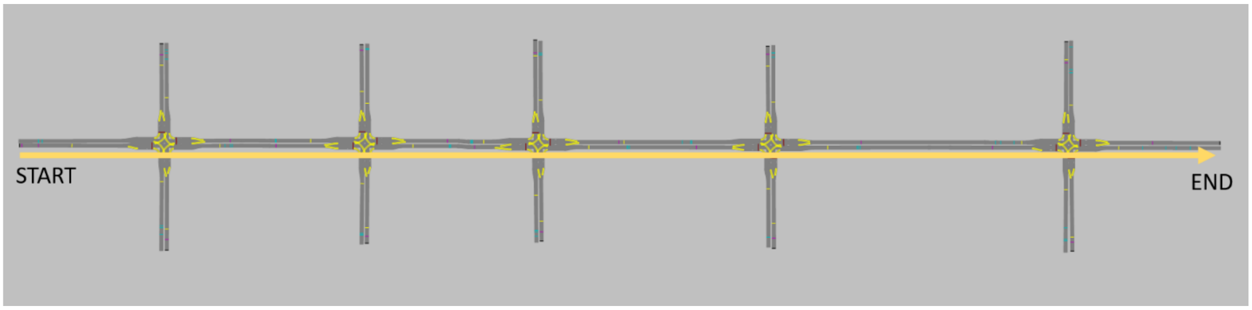


Figure 11: City driving cycle created in Vissim

The main road is inspired by SW 22nd St (Coral Way) in Miami, FL, where there is typically high volume of city traffic. The design of the intersections is modeled after the intersection of Coral Way and SW 27th Ave, where there are two lanes for going straight and two separate lanes for turning right and turning left at the intersection. This design was used over other 3 or 4 lane intersections because it prevents issues with the IDM vehicle getting trapped in a lane that only turns left or right but does not continue straight. The dimensions of the intersections are not set to the exact measurements of the Coral Way and 27th intersection.

Figure 12: On left: Google Maps StreetView of SW 22nd St (Coral Way) and SW 27th Ave in Miami, FL. On right: Intersection in Vissim

All roads in the Vissim model have speed limits set to 35mph (56.3kph), based on the speed limits on Coral Way. The connecting roads comprise of two lanes going in each direction. All of the inbound roads are set to randomly generate a volume of 500 vehicles per hour. These vehicles use the default Wiedemann model in Vissim. Vehicle routes are placed over each intersection to allow any randomly generated vehicle to select between multiple routes, thereby creating a more realistic traffic flow.

The traffic signal controllers have a cycle time of 60 seconds, with the signal state sequence set to the intervals shown in Table 7. There is a 2 second gap between the end of signal 2 and the start of signal 1 to allow any vehicles to cross the intersection before the green state is activated, thereby avoiding collisions. The controller used for each traffic signal along the road alternates between Signal Group 1 and Signal Group 2. Each intersection has potential conflict zones identified and marked such that any randomly generated vehicle will wait for another vehicle to pass before making a left or right turn on the intersection. Each turn is also marked with a reduced speed decision to slow down vehicles before making a sharp turn.

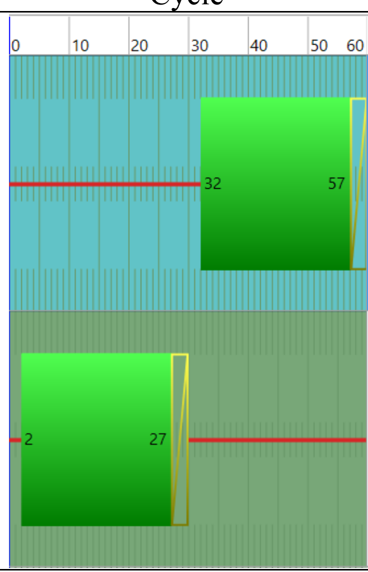
<table>
<tr><th>Signal Group</th><th>Signal Sequence During Cycle</th><th>Green State Duration [s]</th><th>Amber State Duration [s]</th><th>Red State Duration [s]</th></tr>
<tr><td>1</td><td rowspan="2">
</td><td rowspan="2">25</td><td rowspan="2">3</td><td rowspan="2">32</td></tr>
<tr><td>2</td></tr>
</table>

Table 7: Signal controller configuration for Vissim city-based driving cycle

The city-based cycle simulation has a duration of 1800 seconds with a sampling time t_s of 0.1 seconds. Each time the simulation is ran, a new random seed is set to randomize the quantity, starting time, and behavior of the vehicles generated in Vissim. Since there are variations in the driving behavior of the surrounding vehicles, the resulting behavior of the IDM vehicle will be quasi-stochastic as it adapts to the traffic conditions. 100 simulations are conducted for each driving behavior over the created route to account for variations. Since the simulation is ran with respect to time and not distance, the IDM may not travel the exact same distance every run. If the vehicle is too slow, it may not cross all intersections. To account for these variations in distance traveled, the energy usage for each run is normalized by the distance traveled just like the EPA cycles discussed earlier. In addition, the number of stops and the duration of each stop is also calculated to account for situations where the signal timing may allow the vehicle to cross before it changes to red.

4.2.2 Highway-Based Driving Cycle Using Vissim

The second driving cycle is a highway comprised of 2 lanes with 4 onramps and 3 offramps to allow for vehicles to enter or exit the highway. This driving cycle does not resemble any particular real-world highway section since highways can vary greatly based on the geography and surrounding infrastructure.

Figure 13: Highway driving cycle created in Vissim

The speed limit is set to 65mph (104.6kph). All other onramps and offramps connect to roads set to 35mph (56.3kph). The IDM vehicle begins at the first onramp from a stop and accelerates to merge into the highway. The vehicle continues on the highway until it reaches the end of the highway. The end of the highway has a speed reduction zone of 55mph (88.5kph).

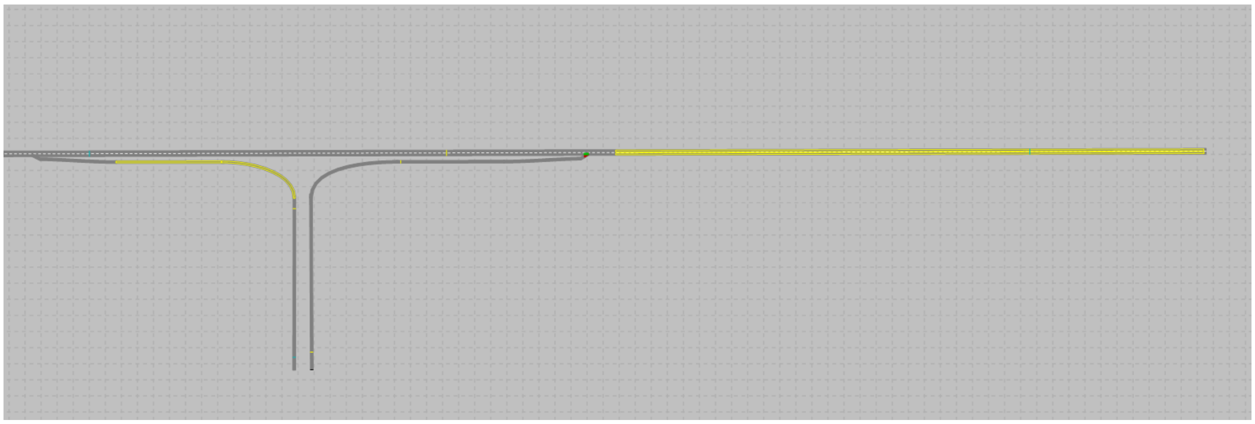


Figure 14: Highway speed reduction zone, marked in yellow

The offramps also have speed reduction zones of 50mph (80kph), and then down to 35mph (56.3kph). The highway is set to have a vehicle input of 2000 vehicles per hour, whereas the onramps are set to 600 vehicles per hour. Routing decisions were added to allow vehicles to continue or exit the highway at any of the offramps, with the exception of the IDM vehicle. The simulation for the highway cycle also has a duration of 1800s and has a sampling time of 0.1s. Like the city-based cycle, the simulation is performed 100 times for each driving behavior to account for variations and averaged over the distance traveled.

Chapter 5: Driving Simulation Results

5.1 Results of Car-Following Simulation Using EPA Driving Cycles

The following results give some perspective of how each of the driving behaviors compare when following a vehicle. As mentioned previously, EPA driving schedules were used for the leading vehicle speed trace. The cycles were stitched together 5 times back-to-back with a 5 second pause in between. The figure below shows the average speed, acceleration, and deceleration of the IDM vehicle for each driving behavior and for each driving cycle.

Figure 15: Average speed, acceleration and deceleration for EPA cycle car-following

Since the IDM vehicle is following another vehicle throughout the duration of the cycle, it can only go as far as the vehicle in front, and therefore its average speed does not vary significantly for each driving behavior type. However, as the driving behavior becomes more aggressive, higher acceleration and braking can be achieved, and the gap becomes shorter between the vehicles.

Since the distance traveled over each driving cycle is different, the percent change in SOC (%SOC) over the entire driving cycle and the net energy usage over the cycle is normalized by the total distance traveled in each cycle. The figures below show the %SOC/km and energy usage in kWh/km for each of the driving behaviors and driving cycles. The arrows show the percentage change by changing from mild to aggressive driving. Since there aren't many opportunities for accelerating or braking in the HWFET cycle, the decrease in %SOC remains approximately the same despite different driving behaviors. For the UDDS and US06 cycles, there is a greater drop in the %SOC per kilometer as the following behavior gets more aggressive.

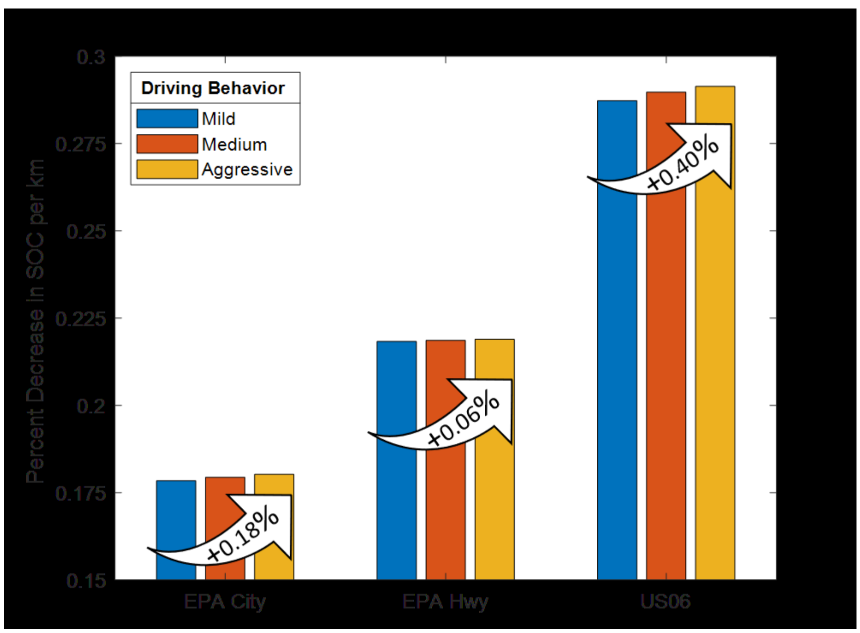


Figure 16: SOC change comparison for 3 driving styles over 3 EPA driving cycles

In terms of the battery energy used in kWh/km, changing from mild to aggressive driving can result in more than 1% in increased energy usage for the UDDS and US06 driving cycles. Despite the percentage difference being small, for car-following scenarios, there is little benefit of driving aggressively since the leading vehicle controls the speed to the destination.

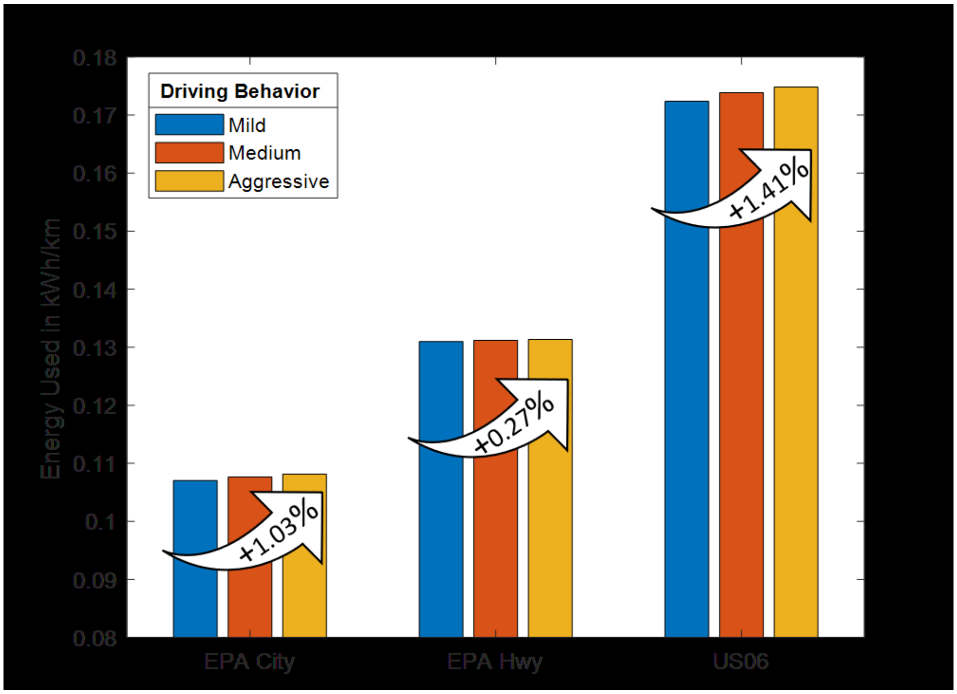


Figure 17: Battery energy consumption comparison for 3 driving styles over 3 EPA driving cycles

5.2 Results of Stochastic Traffic Simulation Using PTV Vissim

5.2.1 Results from City-Based Driving Cycle

The results of the city simulation performed in Vissim is less straightforward than the simulations using the EPA driving cycles. Unlike the EPA driving cycles, the IDM vehicle is not always in a car-following scenario, since it and the other vehicles can change lanes. Additionally, the simulations are not deterministic, because of the random seed changing the dynamics of each run. The simulations have a fixed time period of 1800s, so the distance traveled varies in each simulation run. To compare each run, the data related to the energy usage is normalized by the distance traveled over that run. The average of each distribution is displayed in bold in each figure. The results show that aggressive driving had the greatest variation and the greatest distance traveled on average than the other two driving styles.

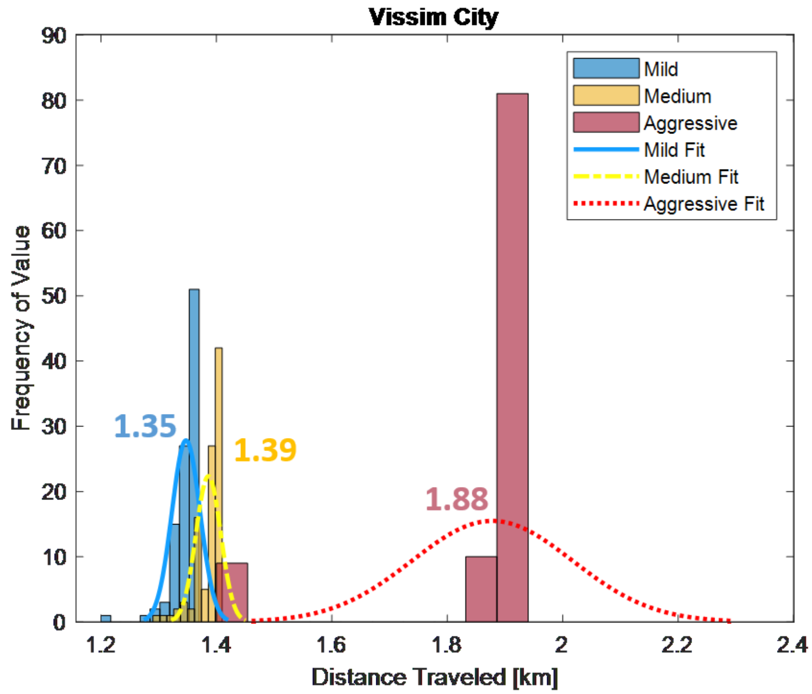


Figure 18: Total distance traveled in Vissim city driving cycle for each driving behavior

Among the three driving behaviors, aggressive driving resulted in the lowest average energy usage per kilometer and the average percent decrease in SOC per kilometer, at 0.137±0.0091kWh/km and -0.228±0.0151%SOC/km. One would expect this to be the highest. Instead, the mild and medium driving styles showed higher energy usage per kilometer, and with medium being the highest at 0.148±0.0077kWh/km and -0.247±0.0129%SOC/km. The normal distribution for the plots of the mild and medium driving have similar dimensions, and also have similar profiles for the distance traveled. Therefore, the distribution profiles suggest that the driving cycle for the aggressive driving style was much more different, since the results from the EPA driving cycles would suggest that the energy used for the aggressive driving style would be a slighter higher than the medium driving style for the same driving cycle.

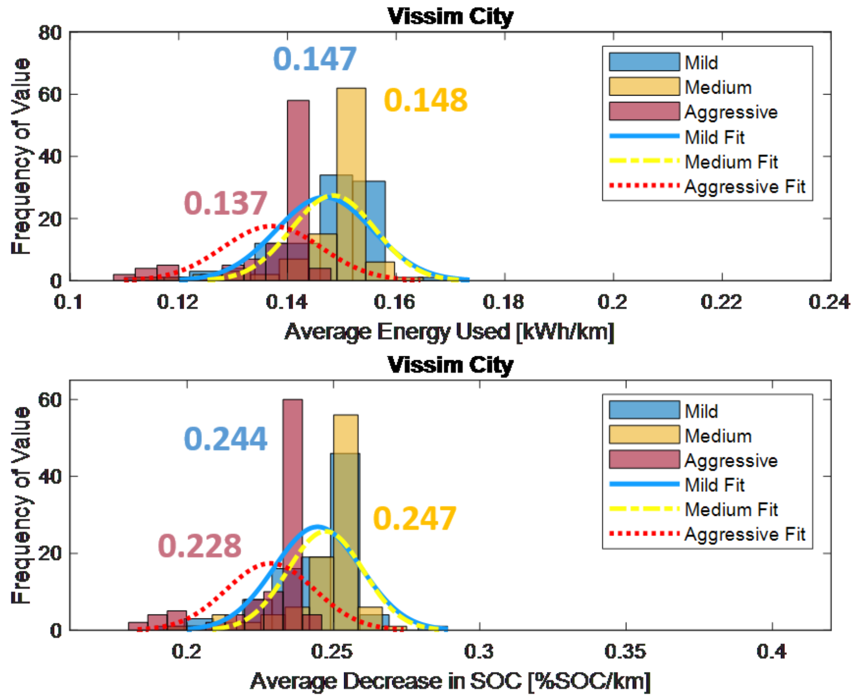


Figure 19: Average energy used per kilometer and average decrease in SOC per kilometer for Vissim city driving cycle

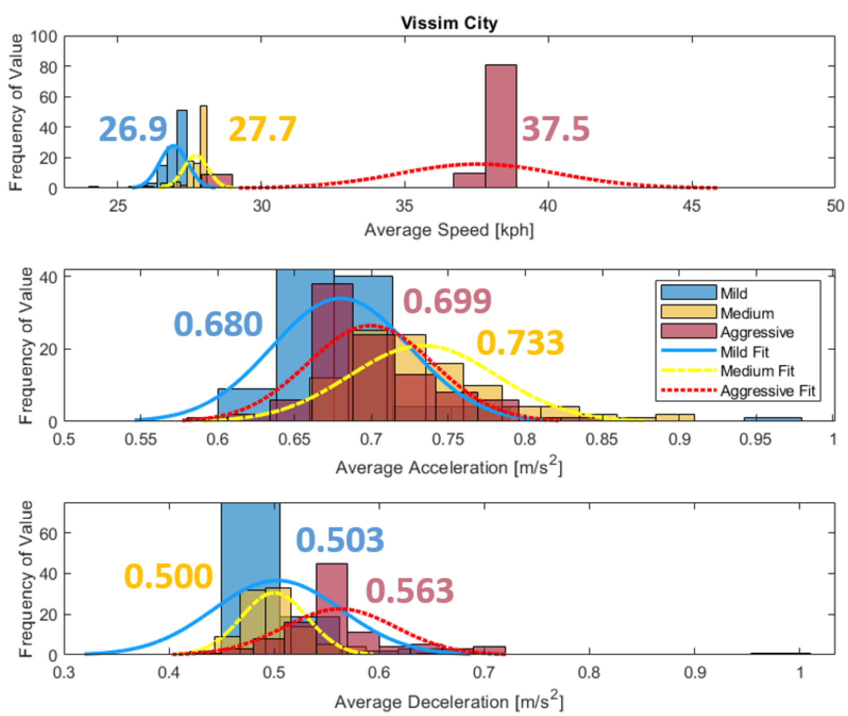


Figure 20: Average speed, acceleration and braking for Vissim city driving cycle

Looking at the average speed, acceleration, and deceleration of each combination of driving behavior and driving cycle, one can find that the trend is not as linear as it was for the EPA cycles. The average speed for mild and medium are close at 26.9kph±0.48kph and 27.7kph±0.43kph, whereas aggressive showed higher average speed of 37.5kph and a greater deviation of 2.8kph, meaning that the vehicle was able to maintain its speed. One would expect the average acceleration and deceleration to increase as the driving becomes more aggressive, but it is not the case here. The average acceleration for aggressive is closer to mild, with mild at $0.680 \pm 0.045 m/s^2$ and aggressive at $0.699 \pm 0.041 m/s^2$. Medium is the highest at $0.733 \pm 0.048 m/s^2$. Lastly, while the average deceleration is highest for the aggressive driving style at $-0.563 m/s^2$, it is not significantly higher than mild and medium, which are both around $-0.50 m/s^2$. To get a better understanding of the traffic environment, one has to also look at the time spent stopped and the number of stops for the city cycle, as the signal timing and signal spacing could have influenced these results.

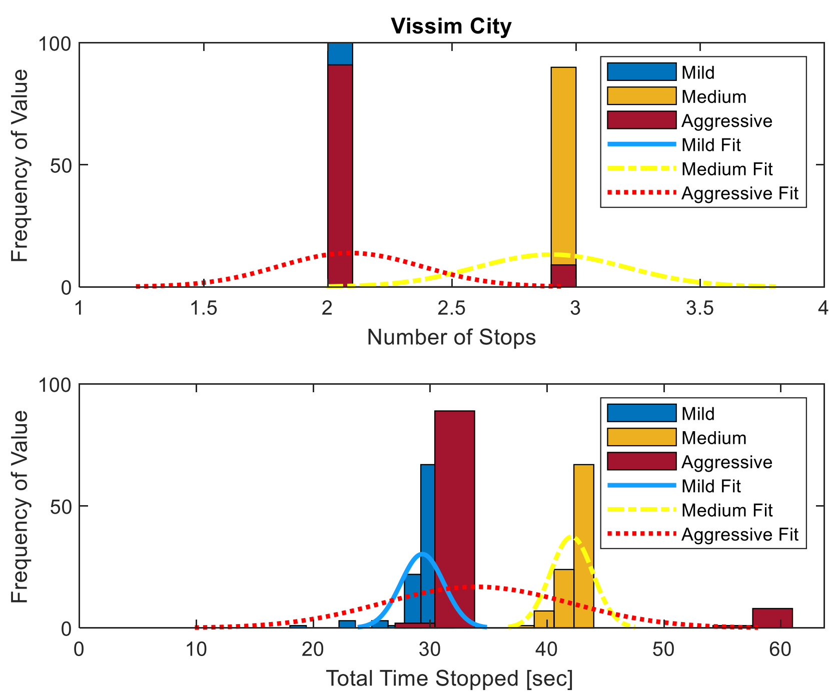


Figure 21: Average number of stops and time stopped for Vissim driving cycles

The mild driving style always had two stops, and the duration was on average around 29s. The medium driving style had three stops on average, which coincides with the time spent stopped being higher. The aggressive driving style had two stops on average and averaged 42s stopped, with some cases having three stops and close to 60s of stop time. Depending the number of stops made, the driving cycle could be different. Looking at a typical run when 2 stops and 3 stops occur, one can see the variation between each driving behavior.

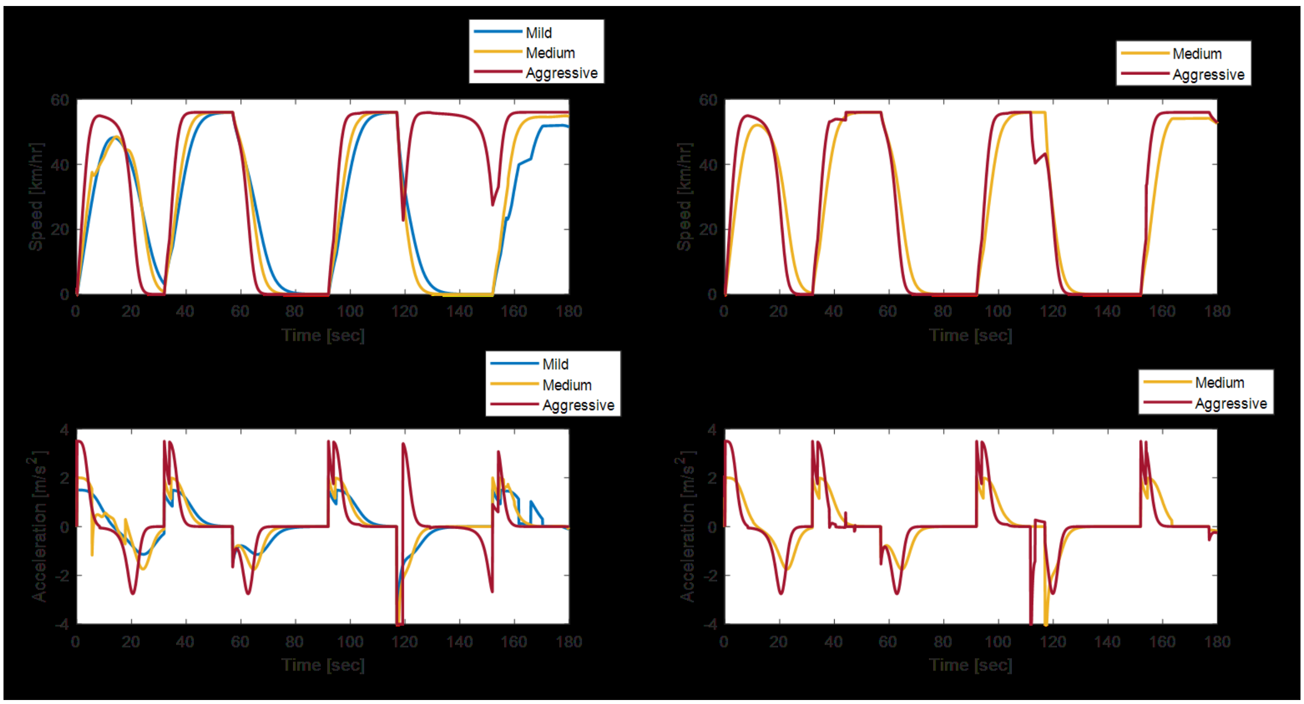


Figure 22: Typical speed and acceleration trace with 2 or 3 stops during city driving cycle

In the speed trace for Run #10, both the mild and medium driver do not come to a complete stop at the first traffic light, which is why they are counted as only having 2 stops in this run. The aggressive driver stops for the first and second traffic signal but catches the third and fourth signal in the transition phase, allowing it to maintain its speed throughout the run. This demonstrates why the aggressive driver usually only stops twice and maintains a higher average speed. Furthermore, because the vehicle was reaching its desired speed sooner, it could avoid more of the traffic being generated because the volume of vehicles on each inbound road is generated gradually. The rate is in vehicles per hour rather than per minute, so in three-minute-long simulation the roads wouldn't reach the full volume.

In the few scenarios where the aggressive driver stops thrice, like in Run #6, its behavior is more similar to the medium driver, but achieves slighter lower average energy usage of 0.143kWh/km as shown in the figure below.

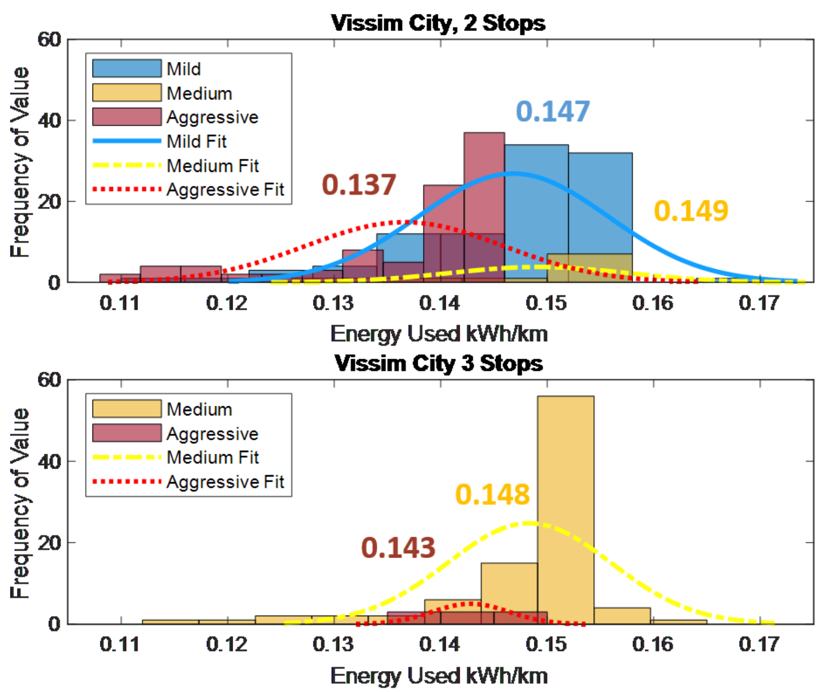


Figure 23: Comparison of energy used per kilometer with 2 or 3 stops during city driving cycle

Looking at the acceleration trace for Run #6, the aggressive driver had taller, narrower acceleration peaks, whereas the medium driver had shorter and wider peaks, which explains why the medium driver achieved a higher average acceleration, and subsequently higher energy usage. However, more data is needed to be able to confidently confirm if aggressive driving always performs better than medium driving in these scenarios with 3 stops.

For all cases, the average battery power discharge rate and charge rate data also suggest why the aggressive driving style used less energy. While it was assumed that the vehicle would be able to regenerate all the energy during braking and no friction braking would be used, the general trend is still as expected. Since the aggressive driving average deceleration rate is higher, the amount of energy recovered when slowing down also increases, which results in the higher charging rate of 9.50kW/km. The aggressive driver maintained a higher average speed, so less energy on average was expended in bringing its speed back up to its desired speed. Therefore, less energy was used overall. Aggressive driving can be beneficial if it helps avoid situations that force the vehicle to slow down. However, in all of these scenarios, the vehicle never exceeded the speed limit to get to the destination faster.

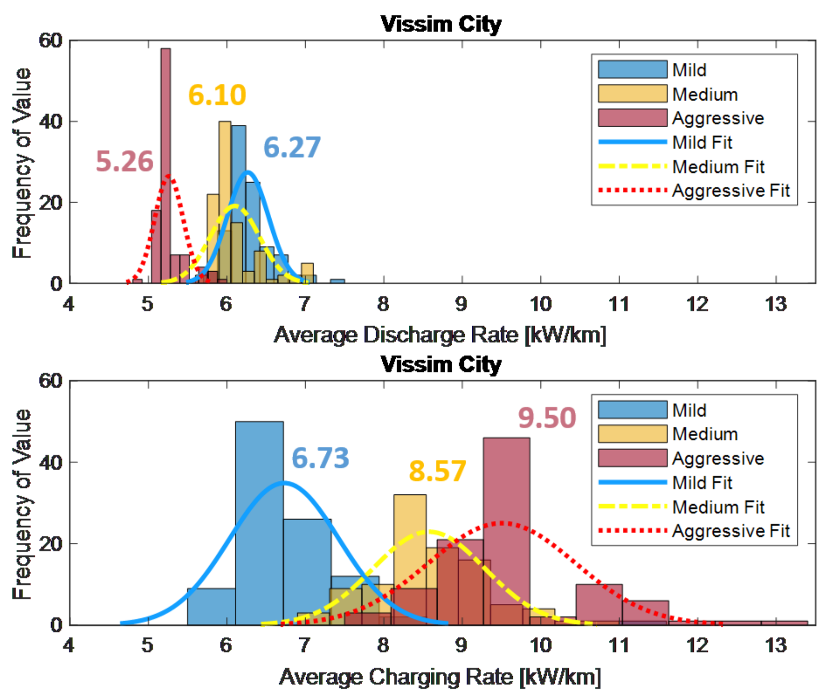


Figure 24: Average discharging and charging rate for city driving cycles

There is still a benefit to mild driving in city driving scenarios. Compared to the medium driving style, mild driving resulted in fewer stops, less time stopped, and a 1.07% decrease in average energy usage in kWh/km. The only drawback is that driving milder is slower on average and in the same period of time the mild driving style traveled 2.8% less distance on average than the medium driving style.

Projecting the average energy usage for each of the driving behaviors in the Vissim city-based driving cycle, one can determine the potential range of the vehicle. The vehicle's battery capacity is 60kWh, so the potential range based on the battery size is over 400km for each of the cases. Assuming that the average energy usage is constant, the aggressive driver can expect 437km of range in city driving scenario, 28km more than the mild driver.

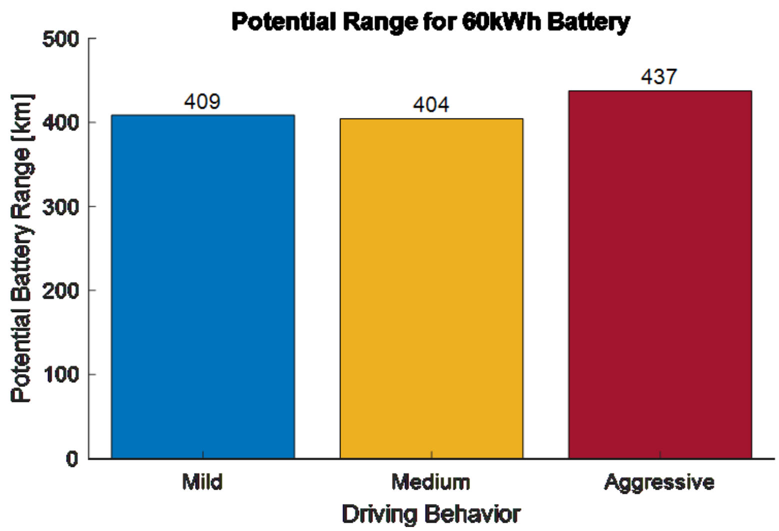


Figure 25: Potential range based on the average energy use per kilometer in Vissim city driving cycle for each driving behavior

5.2.2 Results from Highway-Based Driving Cycle

The data from the highway driving cycle yields more straightforward results since the energy usage is no longer affected by the timing of traffic signals. The total distance traveled increases as the driving behavior becomes more aggressive as there are less obstructions to slow down or stop the vehicle, and the desired speed can be achieved much quicker. Like the city driving cycle, the energy consumption data is divided by the distance traveled in each cycle to account for variations in driving distance over the fixed simulation time. The following figure shows the total distance traveled as a histogram for each driving behavior.

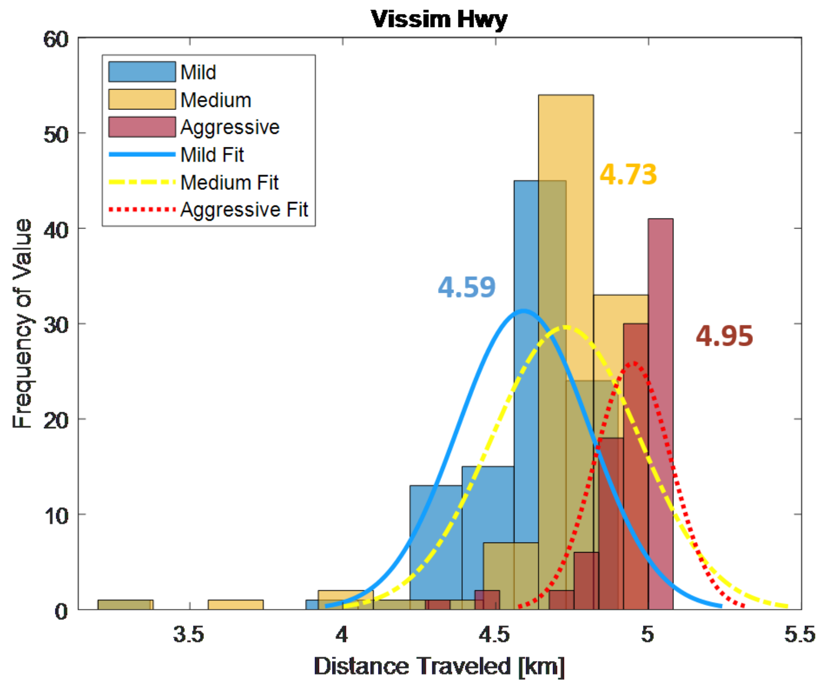


Figure 26: Total distance traveled in Vissim highway driving cycle for each driving behavior

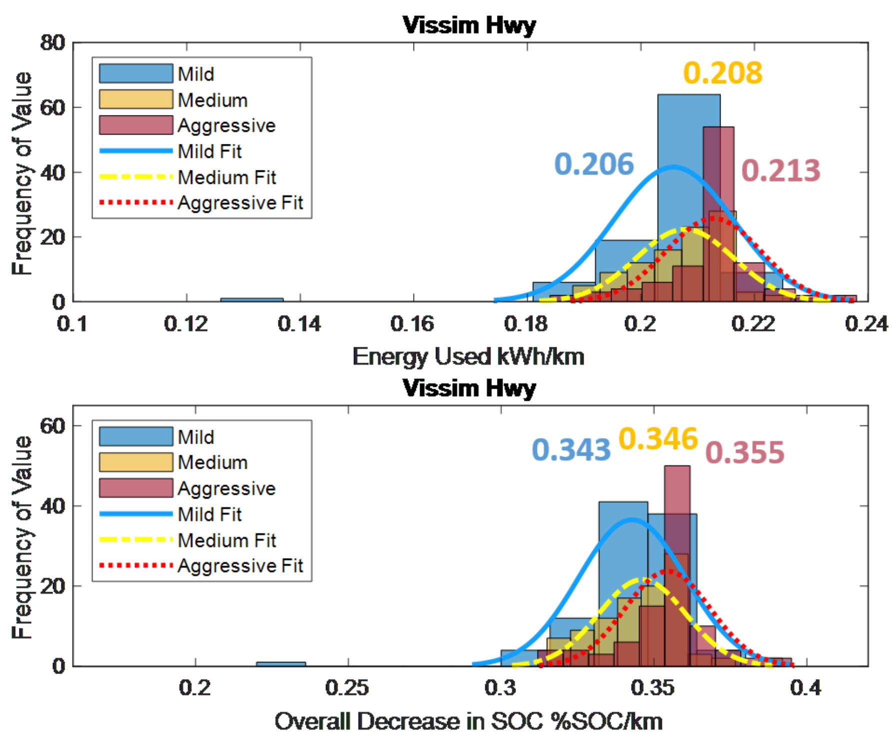


Figure 27: Average energy used per kilometer and average decrease in SOC per kilometer for Vissim highway driving cycle

For the highway driving cycle, the energy usage per kilometer and decrease in SOC per kilometer both increase in magnitude as the driving gets more aggressive. For the mild driving style, the energy usage is 0.206±0.0105kWh/km and increases to 0.208±0.0086kWh/km for the medium driving style, and again up to 0.213±0.0084kWh/km for the aggressive driving style. There is greater variability in energy usage for the mild driving style than the aggressive driving style due to the slight increase in vehicles on the main highway by the time the vehicle merges onto the highway under the mild driving style. The additional vehicles can prevent the vehicle from reaching its desired speed and cause the vehicle to drive slower overall.

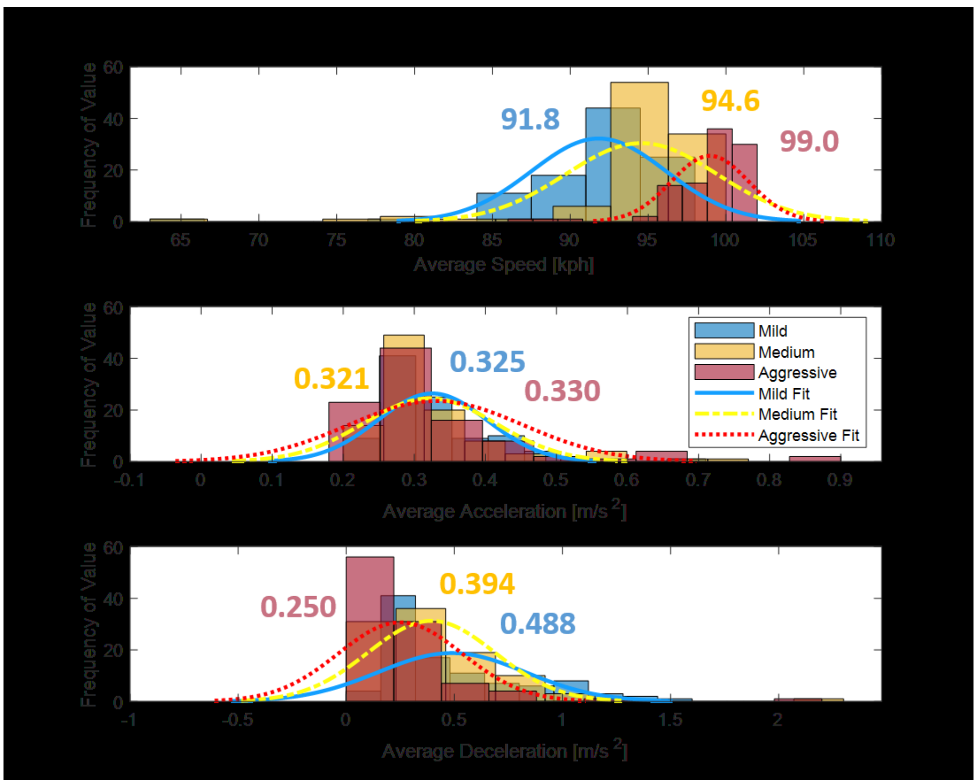


Figure 28: Average speed, acceleration and braking for Vissim highway driving cycle

The histogram of the average speed, acceleration, and deceleration show that the lower average speed, higher average braking, and similar distribution of average acceleration of the mild driving style are the causes for the reduced energy usage. On the other hand, the aggressive driving style has a narrower average velocity distribution, imply that it was able to achieve higher highway speeds more consistently. This is supported by the low average acceleration and low average deceleration, which could have been even lower if not for some outlier cases. The medium driving style fits in the middle between these two, reinforcing that the parameters selections correctly represent a middle ground between these two extremes.

Run #10 and Run #6 highlight the typical speed and accelerations traces found in the Vissim highway cycle and provide further clarification. Both runs show that the mild and medium drivers tended to run into more traffic, especially traffic entering the highway. This merging traffic caused more situations where the vehicle has to slow down and follow the vehicle ahead. As a result, the mild and medium drivers end up driving slower on average and braking more often, as shown in the previous figure. The figure below also confirms that the aggressive driver was decelerating less often and maintaining its speed throughout the driving cycle.

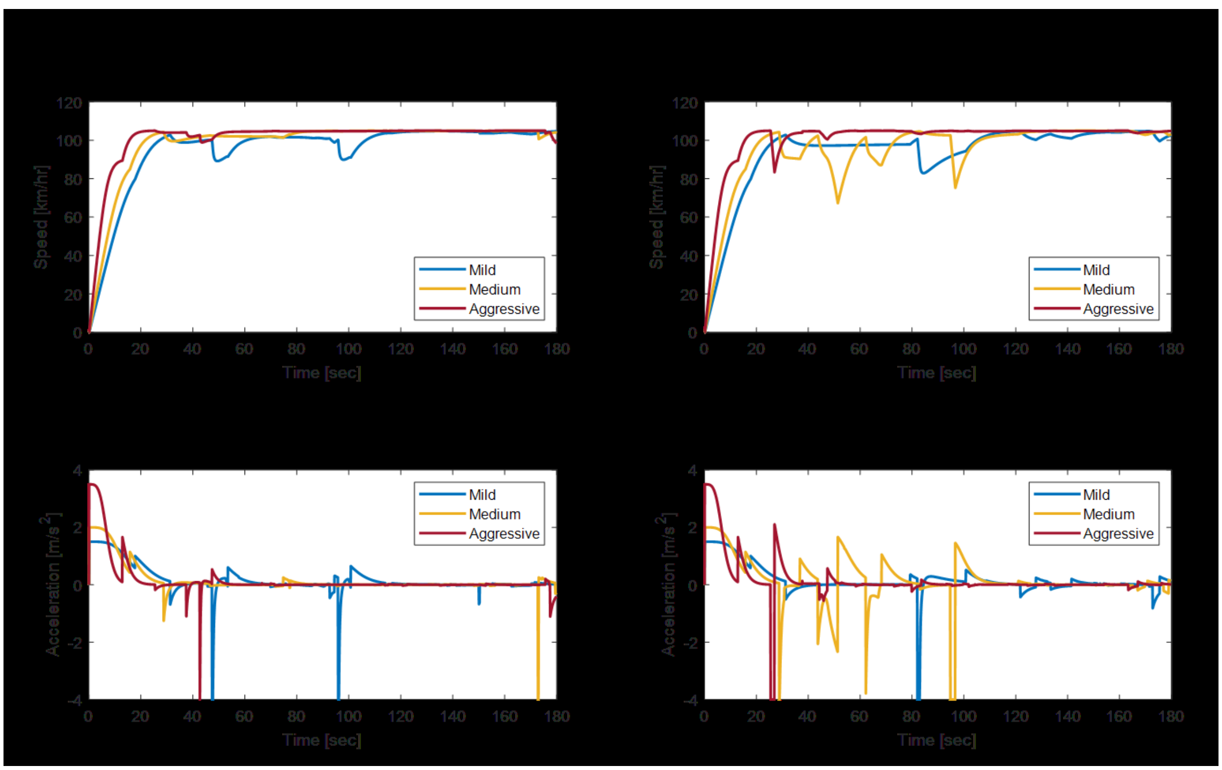


Figure 29: Speed and acceleration traces with 2 or 3 stops during highway driving cycle

In terms of average discharging and charging rates, the mild driving style showed both the highest discharge and regeneration rate of all the driving styles, at 4.96±0.559kW/km and -10.6±4.42kW/km. The medium 4.93±0.648kW/km and -9.06±4.71kW/km, while the aggressive driving style had values of 4.78± 0.623kW/km and -8.32±5.78kW/km. All had similar distributions in terms of width for the discharge rates, but there was more variability in regard to the charging rates. The difference in the charging rate between mild and aggressive driving is greater than the difference in the discharging rate between mild and aggressive driving, therefore overall mild driving leads to less power delivered per kilometer because of the higher charging rates from regenerative braking.

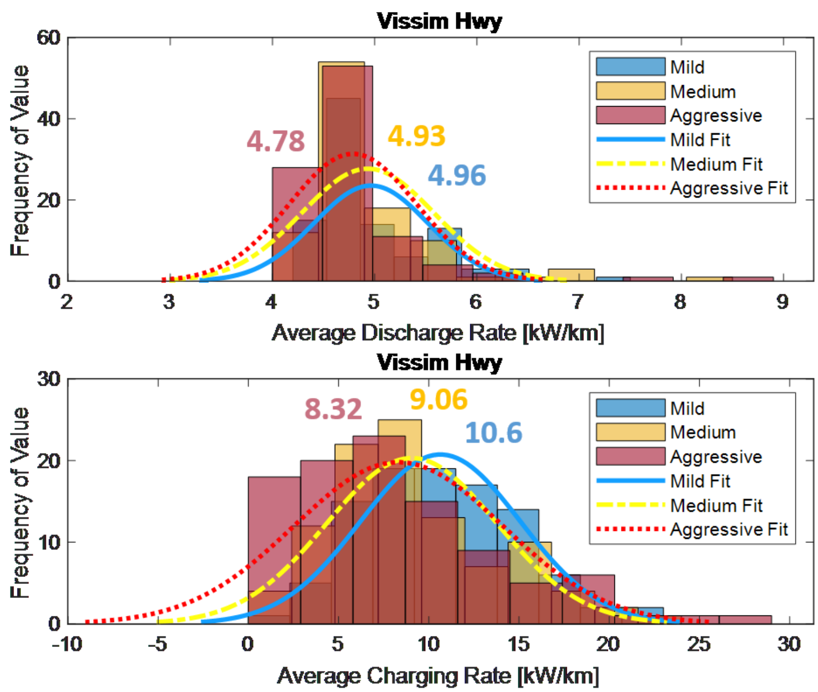


Figure 30: Average discharging and charging rate for highway driving cycles

As another measure for comparison, the projected range using the average energy usage for each of the driving behaviors in the Vissim highway-based driving cycle shows a potential range under 300km for each. Assuming that the average energy usage is constant, the mild driver can expect 292km of range in highway driving scenario, 10km more than the aggressive driver. The range here is much lower than the city driving scenario because the overall energy usage is much higher on the highway because of the aerodynamic losses from maintaining a higher speed and the vehicle does not use regenerative braking as often in these scenarios.

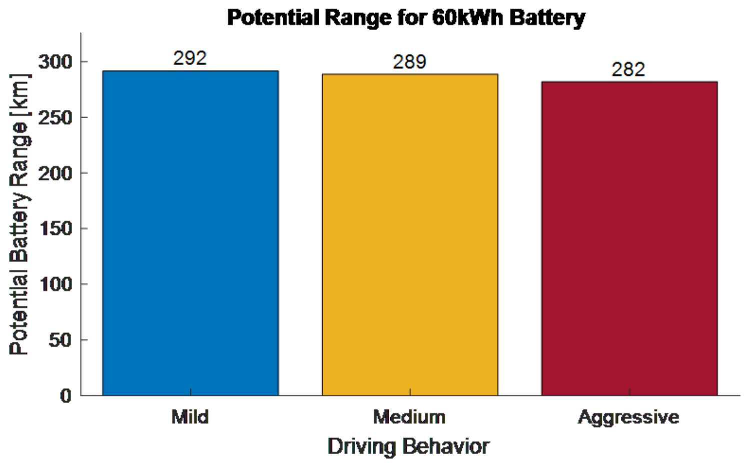


Figure 31: Potential range based on the average energy use per kilometer in Vissim highway driving cycle for each driving behavior

Chapter 6: Conclusion and Future Work

6.1 Conclusion

To summarize, the objective of this study was to evaluate the impact of the driving behavior on the energy consumption of a BEV. A driver model was developed to simulate different levels of driving aggressiveness and BEV powertrain model was created to determine the energy usage of the vehicle. Using the EPA standardized driving schedules and PTV Vissim, several driving cycles were created to simulate common driving scenarios for evaluation.

The results of the single-lane car-following scenario created with the EPA driving schedules show that there is a slight increase in energy usage per kilometer by changing from a mild driving style to an aggressive driving style for the UDDS, HWFET, and US06 driving cycles. For the UDDS-based cycle, going from mild to aggressive increases the energy consumption per kilometer by +1.03%, for the HWFET-based cycle, it increases by +0.27%, and for the US06 cycle, by +1.41%. Switching from mild to medium the incrementation is half as much, at +0.57%, +0.15%, and +0.85% for the UDDS, HWFET, and US06 cycle, respectively. Since the vehicle is unable to go faster than the vehicle it is following, there is no benefit to driving aggressively, and results in wasted energy.

For the driving cycles created in Vissim, the driving behavior makes a more significant impact in the energy consumed. For the city driving cycle created in Vissim, changing from mild driving to aggressive driving can lead to 6.6% decrease in the average energy usage per kilometer driven. Analyzing the driving data, driving aggressively reduced the number of stops, reduced the time spent stopped, greatly increased the average vehicle speed, and reduced the average acceleration in each run. Therefore, if driving aggressively allows the vehicle to avoid red traffic signals and general vehicle traffic, then there is possible to reduce the long-term energy usage. Medium driving is not sufficiently quick to avoid these obstacles; it ends in more frequently stops and more time spent stopped and consequently results in a 1.1% increase in average energy usage over mild driving.

For the highway driving cycle, changing from mild to aggressive driving results in a +3.4% increase in average energy usage per kilometer, with a +7.78% increase in distance traveled compared to mild driving over the 3-minute period simulated. The increased energy usage going from mild to medium driving is less pronounced, at +0.98%, and increases the average driving distance by 3%. Therefore, the small penalty in energy usage from changing from mild to medium driving may be worthwhile if time was a concern.

To give more context on the impact of each driving behavior for each driving scenario, the cost to fully charge the vehicle per year can be calculated using the average energy usage for each scenario. Assuming an average of 12,000 miles per year, and $0.13 per kWh of electricity, switching for the least efficient driving behavior to the most efficient one can result in the following savings: For the EPA car-following driving cycles, switching from aggressive to mild driving can save $2.77/year in the UDDS cycle, $0.09/year in the HWFET cycle, and $6.13/year in the US06 cycle. For the Vissim city cycle, switching from medium to aggressive driving can save $28/year. Lastly, for the Vissim highway cycle, $17/year is saved from switching from aggressive to mild driving. While the savings may not seem significant for car-following scenarios, there is a great benefit to optimizing the driving behavior in regular driving conditions. For the vehicle used in this simulation, the savings in the city could be used to fully recharge the vehicle 4 more times, or it could mean 2 less stops to charge during a road trip on the highway.

Ultimately, these driving tests demonstrate that there is no single solution for all driving scenarios. In car-following scenarios, mild driving is the most efficient. In situations where the traffic environment can slow down the vehicle, it may be beneficial to drive aggressively, so long as the speed limits are obeyed, and the periods of increased acceleration are used to avoid obstructions. Lastly, in highway driving, medium driving might provide the best compromise in terms of energy efficiency and distance traveled over the given time period. Therefore, the level of driving aggressiveness should adapt to the traffic environment in order to minimize charging costs and maximize the range of an EV.

6.2 Future Work

To expand upon the work done in this study, the addition of more complexity in the models can provide greater detail the impact of the driver behavior. A nonlinear battery model could show how the driving aggressive affects the energy consumption at different levels of SOC. A gradual regenerative braking system could help provide more accurate representation of the actual energy usage of the vehicle, since some of the energy is wasted due to friction braking. The driver model could be improved to include more human-like factors such as reaction time, as well as a model for lane-changing based on different levels of aggressiveness. The traffic environment could also vary the traffic density to compare the effect of the driving behaviors in high density traffic. The performance of the vehicle model could be evaluated against a real electric vehicle to validate the model.

The work done in this study could be expanded upon many applications. Since it is programmed in MATLAB, it is accessible and easy to modify in order to evaluate the impact of different models or parameters. Future applications of this work could include vehicle to vehicle communication, vehicle to infrastructure communication, and advanced safety systems by accessing the data stored in the objects in Vissim; applying advanced battery models, cooling system models, and vehicle powertrains; integrating more complex variations of the Intelligent Driver Model; and evaluating the performance of any of these applications over more sophisticated traffic environments using Vissim’s toolset.

References

[1] "Electric Vehicle Sales: Facts & Figures," Edison Electric Institute, Washington, D.C., 2019.

[2] H. Abbas, Y. Kim, J. B. Siegel and D. M. Rizzo, "Synthesis of Pontryagin's Maximum Principle Analysis for Speed Profile Optimization of All-Electric Vehicles," *Journal of Dynamic Systems, Measurement, and Control,* vol. 141, no. 7, July 2019.

[3] C. Bingham, C. Walsh and S. Carroll, "Impact of Driving Characteristics on Electric Vehicle Energy Consumption and Range," *IET Intelligent Transport Systems,* vol. 6, no. 1, pp. 29-35, 2012.

[4] S. Gupta, S. R. Deshpande, P. Tulpule, M. Canova and G. Rizzoni, "An Enhanced Driver Model for Evaluating Fuel Economy on Real-World Routes," *IFAC-PapersOnLine,* vol. 52, no. 5, pp. 574-579, 2019.

[5] J. Kim, R. Rodriguez, K. H. Kwak, D. Jung and Y. Kim, "The Effect of Driver's Behavior and Environmental Conditions on Thermal Management of Electric Vehicles," *SAE Technical Paper,* 2020.

[6] M. Cantas, S. Fan, O. Kavas and S. Tamilarasan, "Development of Virtual Fuel Economy Trend Evaluation Process," *SAE Technical Paper 2019-01-0510,* 2019.

[7] M. Treiber and A. Kesting, Traffic Flow Dynamics Data, Models and Simulation, Heidelberg: Springer, 2013, pp. 181-204.

[8] A. Reuschel, "Fahrzeugbewegungen in der Kolonne bei gleichfoermig beschleunig-tem oder verzoegertem Leifahrzeug," *Oesterreichischer Ingenieur und Architektenverein,* vol. 95, pp. 59-62, 73-77, 193-215, 1950.

[9] L. Pipes, "An Operational Analysis of Traffic Dynamics," *Journal of Appl. Physics,* vol. 24, pp. 274-281, 1953.

[10] G. Newell, "A Simplified Car-Following Theory: A Lower Order Model," *Transportation Research Part B,* vol. 36, no. 3, pp. 195-205, 2002.

[11] P. Gipps, "A Behavioral Car-Following Model for Computer Simulation," *Transportation Research Part B,* vol. 15, no. 2, pp. 105-111, 1981.

[12] S. Krauß, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics," *DLR - Forschungsberichte 8,* 1998.

[13] A. Das and J. Asundi, "A Simple Explicit Model Approximating the Relationship Between Speed and Density of Vehicular Traffic on Urban Roads," *International Journal of Critical Infrastructures,* vol. 8, no. 2/3, pp. 195-204, 2012.

[14] M. Treiber, A. Hennecke and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E,* vol. 62, no. 2, pp. 1805-1824, 2000.

[15] N. Prakash, A. Stefanopoulou, A. Moskalik and M. Brusstar, "Use of Hypothetical Lead (HL) Vehicle Trace: A new method for Evaluating Fuel Consumption in Automated Driving," in *American Control Conference*, Boston, 2016.

[16] Q. Xin, R. Fu, W. Yuan, Q. Liu and S. Yu, "Predictive intelligent driver model for eco-driving using upcoming traffic signal information," *Physica A: Statistical Mechanics and its Applications,* vol. 508, no. 15, pp. 806-823, 2018.

[17] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng and F.-Y. Wang, "Capturing Car-Following Behaviors," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 19, no. 3, pp. 910-920, 2018.

[18] T. J. Ayres, L. Li, D. Schleuning and D. Young, "Preferred Time-Headway of Highway Drivers," in *IEEE Intelligent Transportation Systems Conference*, Oakland, 2001.

[19] P. Hertzke, N. Mueller, P. Schaufuss, S. Schenk and T. Wu, "Expanding electric-vehicle adoption despite early growing pains," McKinsey & Company, August 2019. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/expanding-electric-vehicle-adoption-despite-early-growing-pains. [Accessed 20 March 2020].

[20] "Tires and Passenger Vehicle Fuel Economy: Informing Consumers, Improving Performance Special Report 286," 2006.

[21] "Chevrolet Bolt EV - 2017," Chevrolet, [Online]. Available: https://media.chevrolet.com/media/us/en/chevrolet/vehicles/bolt-ev/2017.tab1.html. [Accessed 16 March 2020].

[22] "2014 Chevrolet Spark Specifications," Chevrolet, [Online]. Available: https://media.gm.com/media/us/en/chevrolet/vehicles/spark-ev/2014.tab1.html. [Accessed 16 March 2020].

[23] "2017 Chevy Bolt EV Is Less of a Drag Than Originally Believed," HybridCars.com, [Online]. Available: https://www.hybridcars.com/2017-chevy-bolt-ev-is-less-of-a-drag-than-originally-believed/. [Accessed 16 March 2020].

[24] "Chevrolet Spark EV – 2016," Chevrolet, [Online]. Available: https://media.chevrolet.com/media/us/en/chevrolet/vehicles/spark-ev/2016.html. [Accessed 16 March 2020].

[25] "2014 Chevrolet Spark EV Tires," TireSize.com, [Online]. Available: https://tiresize.com/tires/Chevrolet/Spark/2014/EV/. [Accessed 17 March 2020].

[26] F. Momen, K. Rahman and S. Y., "Electrical Propulsion System Design of Chevrolet Bolt Battery Electric Vehicle," *IEEE Transactions on Industry Applications,* vol. 55, no. 1, pp. 376-384, 2019.

[27] "LiNiMnCo Polymer Cell: 3.7V 9Ah (33.3Wh, 18A rate) PL-7789182-2C," BatterySpace.com, [Online]. Available: https://www.batteryspace.com/limnnipolymerli-ioncell37v12ah444wh24arate.aspx. [Accessed 16 March 2020].

[28] "LiNiMnCo Polymer Cell: 3.7V 9Ah (33.3Wh, 18Ah rate)," BatterySpace.com, [Online]. Available: https://www.batteryspace.com/limnnipolymerli-ioncell37v12ah444wh24arate.aspx. [Accessed 16 March 2020].

[29] R. Kruse and T. Huls, "Development of the Federal Urban Driving Schedule," *SAE Technical Paper 730553,* 1973.

[30] PTV Vissim 11 User Manual, Karlsruhe: PTV AG, 2018.

[31] B. Higgs, M. M. Abbas and A. Medina, "Analysis of the Wiedemann Car Following Model over Different Speeds using Naturalistic Data," in *Procedia of RSS Conference*, 2011.

[32] "Global EV Outlook 2019," International Energy Agency, 2019. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2019 . [Accessed February 2020].

Appendices

A.1 Tutorial on PTV Vissim and Component Object Model using MATLAB

A.1.1 Introduction to PTV Vissim

This chapter of the appendix serves as a guide about the traffic environment in PTV Vissim and how the simulation was set up in MATLAB to connect with the Component Object Model (COM) in Vissim. Since there are not many detailed resources available explaining how the COM works in conjunction with MATLAB, this section aims to provide additional clarity on the topic.

PTV Vissim is a traffic flow simulation software developed in Karlsruhe, Germany. It is capable of microscopic simulations, meaning that each object (such as a vehicle, pedestrian, or train) can be simulated individually, with its own properties and behavior. It is also multi-modal, meaning that it can simulate multiple traffic types like cars, pedestrians, bicycles, and so on. Its primary use is simply to design roadways and simulate traffic flow, but it offers a lot of flexibility and accessibility, thus allowing the user to create complex traffic environments. It can also be combined with PTV Group’s other software, such as PTV Visum for example, for creating map network to import into Vissim. Additionally, software languages, such Python, MATLAB, and Microsoft Visual Basic can be used to access the information stored in Vissm’s objects and run external programs. Before discussing how that can be done, first the driver model used in Vissim must be understood as it is a key component to the simulations performed in Vissim.

A.1.2 Wiedemann Car-Following Driver Model

The driving behavior of the vehicles in Vissim is based on a modified version of the psycho-physical Wiedemann car-following model created in 1974 [30]. The model is based on four driving states or regions. The states are free flow, approaching, following, and braking. According to [7], each of these states have a different acceleration function $a(s,v,\Delta v)$. The states are separated by speed and gap-related thresholds which are based on nonlinear functions. The convention used by the authors in [7] and [31] refer to the thresholds as CLDV (Closing Difference in Velocity), OPDV (Opening Difference in Velocity), SDV (Sensitivity Difference in Velocity), ABX (minimum gap for free flow, following, and approaching regions), and SDX (maximum gap for following region). Looking at the figure below, the line coming down from the top right of the plot describes the flow of state changes in a scenario where a vehicle is approaching a slower moving vehicle. The vehicle starts off at the free-flow state, and then transitions to the approaching state, and then following state when the gap and approach rate decrease sufficiently. The vehicle remains in that region until there is a significant change in the gap or the approach rate.

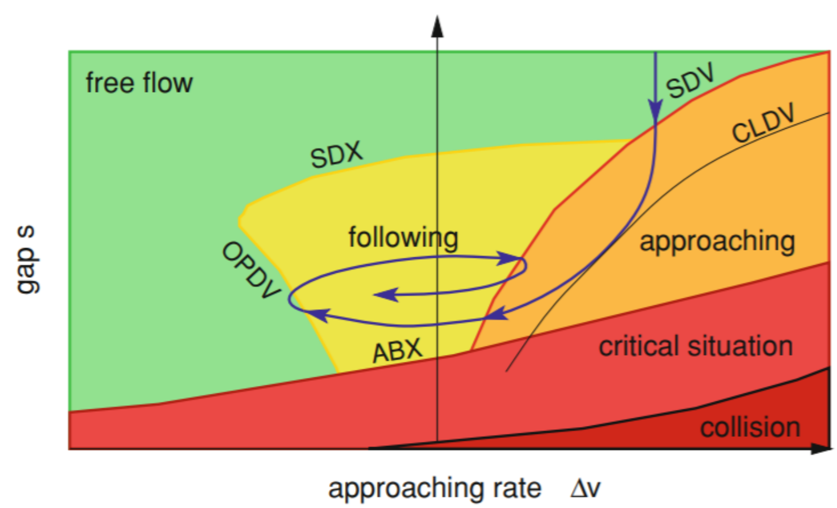


Figure 32: States of the Wiedemann model [7]

The model used in Vissim also uses a rule-based algorithm for lateral movement, such as changing lanes. The vehicle can consider the behavior of 4 vehicles in front by default, and two vehicles in the adjacent lanes [30]. The overall model is stochastic, allowing for slight variations in the behavior of each vehicle in each simulation. Each simulation run has a random seed that acts as a starting point for generating randomness in the vehicle's velocity, perception of other objects, and other factors. The seed can be changed so that each simulation run results in slightly different results.

A.2 PTV Vissim Component Object Model and MATLAB

A.2.1 Overview of the Component Object Model

One of the advanced features in Vissim is the Component Object Model (COM) programming. The COM allows users to access the data and functions contained in Vissim through programming languages like Visual Basic for Applications (VBA), Python, C/C++, and MATLAB. This feature allows users to implement other functions or scripts that are not available in Vissim. Most of the help guides are presented in VBA or Python, so the following sections will explain how to use the COM in MATLAB.

The Vissim COM uses a hierarchy-based system for accessing different objects. Objects can range from real things like vehicles, links, and stops signs, to simulation settings, conflict areas, and camera positions. In the figure below, each object is written with an "I" for "Interface" in front of the object name. While it is typically not used when calling out objects, it is mentioned here because it is sometimes used for referencing object types.

Objects are divided into three classes: objects, containers, and collections. Objects are simply all the things in Vissim, like a vehicle (Vehicle #25), a traffic signal (Traffic signal #2), or a reduced speed area (Reduced speed area #3).

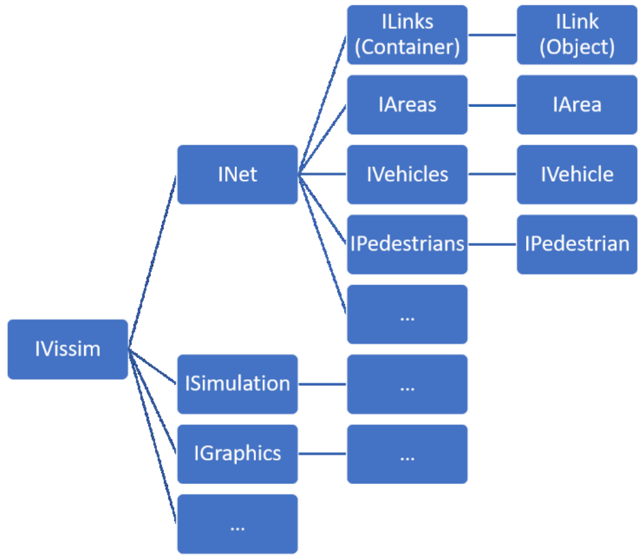


Figure 33: Vissim hierarchy of objects

Collectively, all these objects are put into a container, which contains all of the same objects. Generally, all containers are the plural form of the object they contain. For example, the container "ILinks" would contain all the objects "ILink", which are all the links in the simulation. A container would be used to access a certain property for all the objects within that container. Lastly, a collection contains references to objects. As an example, a static vehicle route "IVehicleRouteStatic" has the property "ILinkSeq", which is the order of the links in that vehicle route. ILinkSeq only references the links, so if a link is removed from ILinkSeq, it is no longer in the vehicle route, but the link will still be in the simulation. All of the objects, containers and collections in Vissim can be found in the "Objects" section of the "COM Help" document.

A.2.2 Accessing Data Using COM and MATLAB

Objects in Vissim have attributes, which are the parameters that define that object. For the object IVehicle, attributes include its speed, its desired speed, its current position, and its weight, to name a few. Attributes can be accessed using various functions for an object or a container. For example, to edit the volume of vehicles traveling on a link to 600 vehicles per hour, one can set the "Volume(1)" attribute to 600 on the first IVehicleInput as shown:

```
set(Vissim.Net.VehicleInputs.ItemByKey(1),'AttValue','Volume(1)', 600);
```

Here the "ItemByKey" term is a property, which are functions that call out a single object within a container. The key "1" is entered to call out the first IVehicleInput in the container IVehicleInputs. Then the term "AttValue" is used to state that an attribute is being accessed, called "Volume(1)". The 1 in this case denotes the first defined time interval. This is specific to some attributes. Finally, the value of the new volume is entered. One can substitute the volume and key callout value for a variable in MATLAB if desired. To illustrate this, here is another example which gets the link number of the first IVehicleInput.

```
LinkNum=get(Vissim.Net.VehicleInputs.ItemByKey(VehInput_Num),'AttValue
','Link');
```

In this example, the variable VehInput_Num contains the key or number of the desired IVehicleInput. Entering "Link" as the attribute gives the link number corresponding the desired IVehicleInput, and is saved in the variable LinkNum.Typically, it is preferred to access the attribute values for all the objects in a container. Most containers have specific functions called "Methods" that allow access to the information inside the container. To implement the IDM in Vissim, it is necessary to able to access all the parameters of each vehicles at every time step. One option is to use a for loop and the "GetAll" Method in Vissim to get the desired attributes for all the objects at that time step, as shown below.

```
All_Vehicles = Vissim.Net.Vehicles.GetAll; % get all vehicles in the
network at the actual simulation second
for cnt_Veh = 1 : length(All_Vehicles),
    veh_number = get(All_Vehicles{cnt_Veh}, 'AttValue', 'No');
```

```
    veh_type = get(All_Vehicles{cnt_Veh}, 'AttValue', 'VehType');
    veh_speed = get(All_Vehicles{cnt_Veh}, 'AttValue', 'Speed');
    veh_position = get(All_Vehicles{cnt_Veh}, 'AttValue', 'Pos');
    veh_linklane = get(All_Vehicles{cnt_Veh}, 'AttValue', 'Lane');
end
```

This method is available in the "COM Basic Commands" MATLAB script found in the "Examples Training" folder in Vissim. This is the slowest method for getting the attributes. A faster approach is to use the "GetMultiAttValues" method, which gets the attribute values for all the objects in the container. The output is a cell with the indices in the first column, and the values in the second column.

```
veh_numbers = Vissim.Net.Vehicles.GetMultiAttValues('No');
veh_numbers = cell2mat(veh_numbers(:,2)); % convert second column to a
matrix
```

The fastest approach is the "GetMultipleAttributes" method since it creates a cell with each column being the desired attribute as they are called out.

```
all_veh_attributes = Vissim.Net.Vehicles.GetMultipleAttributes({'No';
'VehType'; 'Speed'; 'Pos'; 'Lane'});
```

A.2.3 Setting Up a Simulation Using COM and MATLAB

Prior to starting a new simulation, it is important to configure MATLAB properly for Vissim. All of the functions in Vissim can only accept vectors, not matrices. By default, MATLAB will pass matrices, so this may cause some errors. To avoid this issue, adding this line of code will cause MATLAB to only pass vectors to Vissim, even if a matrix is used.

```
feature('COM_SafeArraySingleDim', 1); % Matlab should only pass one-
dimensional array to COM
```

To open a Vissim network and layout file in Vissim using MATLAB, first a new Vissim window must be opened. This step allows gives to access Vissim's objects and functions.

```
% Connecting the COM Server => Open a new Vissim Window:
Vissim = actxserver('Vissim.Vissim');
```

Then the Vissim Network file Traffic_Environment.inpx and the Layout file Traffic_Environment.layx are loaded. The variable Path_of_Traffic_Environment_network is set to “cd”, which stands for current directory. The network and layout files must be stored in the same folder as the MATLAB script in order to be loaded. The files can be opened using the “LoadNet” and the “LoadLayout” functions.

```
Path_of_Traffic_Environment_network = cd;
%'C:\Users\Public\Documents\PTV Vision\PTV Vissim 11\My Files\Thesis
Files\IDM_main';

% Load a Vissim Network:
filename =
fullfile(Path_of_Traffic_Environment_network,'Traffic_Environment.inpx
');
flag_read_additionally = false;
Vissim.LoadNet(filename, flag_read_additionally)% Load a Layout:
filename =
fullfile(Path_of_Traffic_Environment_network,'Traffic_Environment.layx
');
Vissim.LoadLayout(filename);
```

Any changes to the network or layout file can be saved using the “SaveNetAs” and “SaveLayout” methods.

```
% Saving
Filename = fullfile(Path_of_Traffic_Environment_network,
'Traffic_Environment.inpx');
Vissim.SaveNetAs(Filename)
Filename = fullfile(Path_of_Traffic_Environment_network,
'Traffic_Environment.layx');
Vissim.SaveLayout(Filename)
```

Once a network and layout file are loaded, any commands ran in the Command Window in MATLAB or from a script will act on the open network. It is recommended that most objects related to the traffic environment should be created and set in Vissim instead of MATLAB for each of use. Once the traffic environment is created, MATLAB can access the information stored in those objects through the COM.

Before running a simulation, it is important to set up the simulation settings. These can be set in MATLAB so changes can be made quickly without having to edit the settings on the Vissim network file. The object ISimulation contains all the attributes related to simulation such as the number of runs, resolution, and the duration. Similar to the past examples, the set function can be used to set the value of each attribute. The first attribute is the SimPeriod, which is the duration of the simulation measured in simulation seconds. This value is not equal to real world time; it is only a measurement of the simulated duration.

```
t_sim = 60; % duration of simulation in seconds
set(Vissim.Simulation, 'AttValue', 'SimPeriod', t_sim);
```

The simulation resolution SimRes is the number of time steps per simulation second. The default value is 1, and the max is 10. The difference in time between each time step will be equal to 1/SimRes. To improve the performance of the IDM, the SimRes is set to 10 so the vehicle's speed and acceleration is calculated every 0.1s.

```
SimRes = 10; % resolution, time step [s] per simulation second (max is 10). Each time step is 1/SimRes seconds long
set(Vissim.Simulation, 'AttValue', 'SimRes', SimRes);
```

The Wiedemann car-following model used in Vissim is stochastic by design, and to ensure randomness, random seeds are introduced to create variation between simulations. A random seed is a number used to generate random numbers. If one wishes to calculate the average value over several runs, the random seed can be set for the first run and then incremented after every run so that the random values used in each simulation are different. This allows the average to represent the results over many simulations with slight variations in the vehicle's speed and routes.

```
RandSeed = 45; % random seed, can be any integer value
set(Vissim.Simulation, 'AttValue', 'RandSeed', RandSeed);
RandSeedIncr = 1; % incrementation of random seed after each run
set(Vissim.Simulation, 'AttValue', 'RandSeedIncr', RandSeedIncr);
NumRuns = 5; % number of simulation runs
set(Vissim.Simulation, 'AttValue', 'NumRuns', NumRuns);
set(Vissim.Simulation, 'AttValue', 'UseMaxSimSpeed', true);
```

To start a simulation run, one can use the method "Vissim.Simulation.RunSingleStep" to run a single time step. The simulation can be ran continuously using the method "Vissim.Simulation.RunContinuous". A break point can be added by setting the "SimBreakAt" attribute of ISimulation to the time the break point occurs.

```
set(Vissim.Simulation,'AttValue','SimBreakAt', 40); % break point at
40s
```

To stop the simulation, use "Vissim.Simulation.Stop". To close Vissim, use "Vissim.Exit".

A.2.4 Adding Vehicles to a Simulation

Vehicles can be added to a simulation through random generation or deterministically through various methods. Vehicles are generated randomly by setting an IVehicleInput to a link, which determines the number of vehicles per hour going through that link. The vehicles generated in this manner follow the Wiedemann car-following model but do not have a set route. Instead, they travel along the same link until they reach a connector with its direction attribute set to "All" starting at the link they are traveling on. In the figure below, the four vehicles traveling on Link 1 reach the Connector 10000 along their route. Since the vehicles do not have a set route, they will always change to Connector 10000. If the direction attribute of Connector 10000 was set to "Left" or "Right", then the vehicles will not travel on Connector 10000 because they cannot make the choice to travel of the connector set to a direction other than "All".

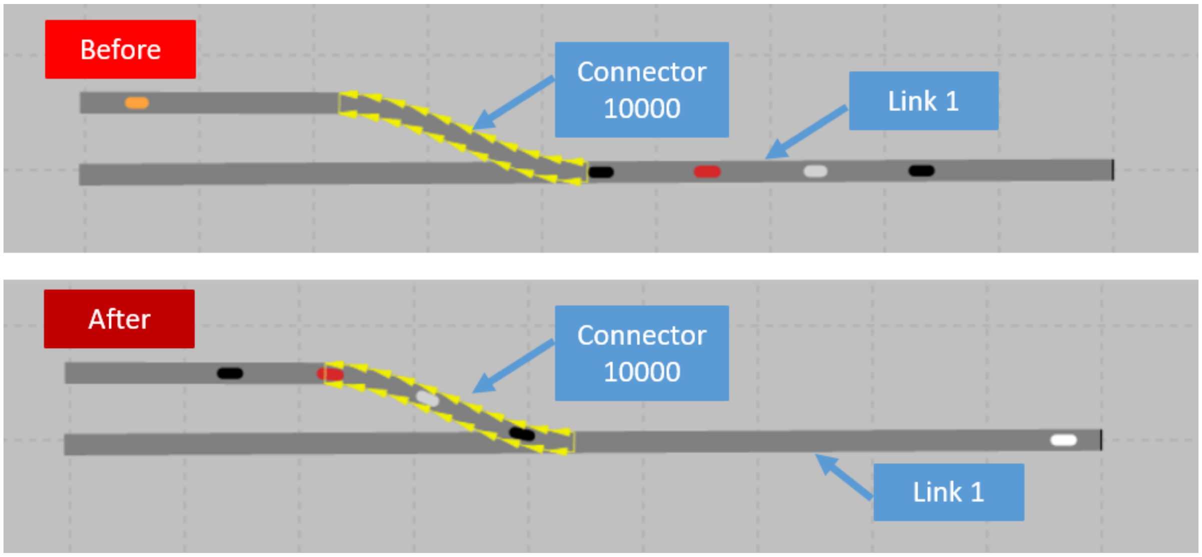


Figure 34: Route selection of vehicles generated without a set vehicle route

To correct this issue, vehicle route can be added so that the vehicles know what route are possible. Two vehicle routes are added in the figure below. The route starts at the magenta line and ends at either of the two turquoise lines. The vehicles decide when crossing the start line of which of the two routes to select.

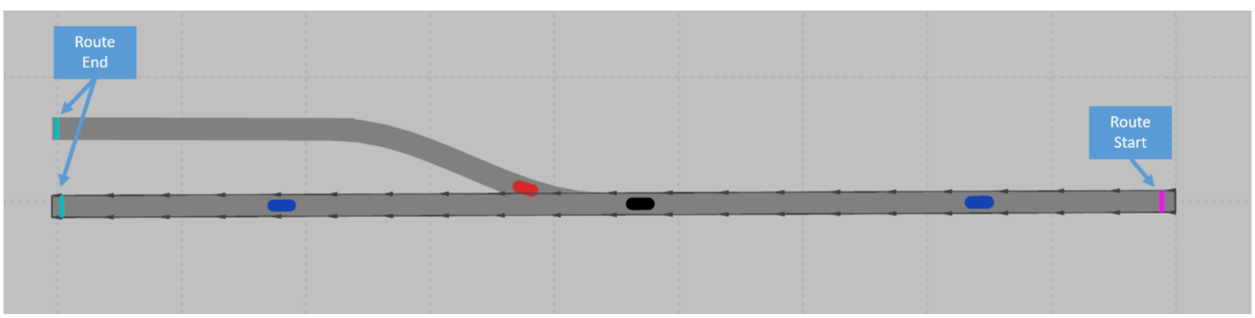


Figure 35: Use of static vehicle routes resolves vehicle routing problem

The static vehicle route does not need to terminate at the end of a link and can start at any point of a link. Therefore, it's best to use them around intersections or split paths where multiple routes are possible. In addition, the routes should be created from the same starting point, since the vehicles will set a route depending on the first starting point they cross. Otherwise if there is another static vehicle route further down the same link, it will never be used because the vehicles already have a route.

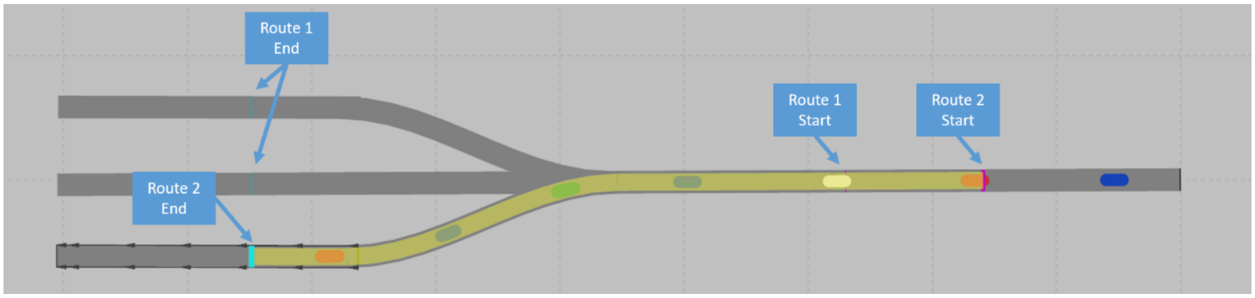


Figure 36: Vehicles stay on Route 2 until completed and will ignore Route 1

The other approach is to manually add vehicles using MATLAB. The method "AddVehicleAtLinkPosition" takes 6 inputs: the vehicle type, its desired speed, the starting link, the starting lane, the position on that link, and whether it interacts with other vehicles or objects. The vehicle type describes if the vehicle is a car, bus, tram, HGV (heavy goods vehicle), pedestrian, or a bicyclist. The interaction Boolean determines whether the vehicle interacts with surrounding vehicles and responds to the traffic environment using the Vissim's modified Wiedemann driving model. Setting it to "False" means that the vehicle travels at its desired speed regardless of other vehicles or objects in the environment. Similar to the vehicles generated using IVehicleInput, the vehicles created this way do not follow a specific route. The vehicle will always go to the first connector set with the direction "All" unless it runs into a static vehicle route along its path that gives the opportunity to travel to a different link. The limitation of this method of adding vehicles is that the simulation run must be started, or an error will occur. Run a single time step using "Vissim.Simulation.RunSingleStep" and then enter the following to add a vehicle:

```
% Add vehicle to network
vehicle_type = 100; % set to 100 for car
desired_speed = 25; % set initial speed of first vehicle [km/h]
link = 1;
lane = 2;
xcoordinate = 10; % set initial position of first vehicle [m]
```

```
interaction = true; % optional boolean
Vehicle_1 =
Vissim.Net.Vehicles.AddVehicleAtLinkPosition(vehicle_type,...
    link, lane, xcoordinate, desired_speed, interaction);
```

A.2.5 Incorporating the Intelligent Driver Model

Using the techniques discussed in the previous sections the IDM can be incorporated into a vehicle in Vissim. It must be noted that Vissim has built-in menus for running external driver models without having to operate Vissim externally using a programming language. An external driver model can be added to a vehicle type in the vehicle type settings menu.

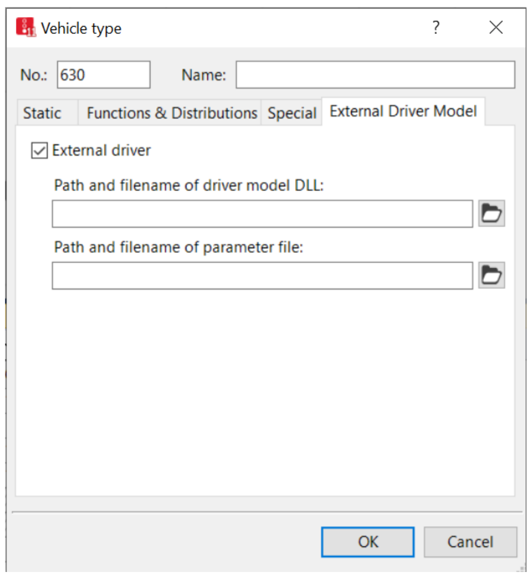


Figure 37: Adding an external driver model to a vehicle type

The driver model DLL file and its parameter file can be tied to a specific vehicle type for comparisons to the default vehicles in Vissim. The driver model DLL file is called out at each time step and can affect all or some of the vehicles in the network. The DLL file must be written in C or C++, which limits the accessibility of this method. However, the focus here is to implement the IDM using MATLAB, which is still feasible without using Vissim's built-in menu for external driver models. More information about adding an external driver model can be found in Section 18.3 of the Vissim manual [30].

To incorporate the IDM in a vehicle in Vissim, the AddVehicleAtLinkPosition method is used to generate the IDM vehicle. To use AddVehicleAtLinkPosition, the simulation must be already running, so it is necessary run a single time step using Vissim.Simulation.RunSingleStep outside of the main for-loop used for the IDM. The interaction Boolean must be set to "False" to avoid the Wiedemann model from overriding the IDM. It is important to note that although the generated vehicle will not use the Wiedemann driver model, but it is still capable of performing lane changes in accordance to Vissim's algorithm. The output should be saved to a variable so that the attributes from the IDM vehicle can be easily accessed.

```
Vissim.Simulation.RunSingleStep; % Run first simulation step i = 1;
Vehicle_IDM=Vissim.Net.Vehicles.AddVehicleAtLinkPosition(vehicle_type,
link, lane, xcoordinate, desired_speed, interaction);
```

The vehicle now appears in the Vissim traffic environment. Now the information stored in the vehicle can be used. Any of the methods discussed earlier to assess the stored vehicle data, but it is recommended to use the "get" function to access the data specific to the IDM vehicle. For example, the attribute "Speed" contains the speed of the vehicle in km/hr. The current vehicle speed can be stored in a vector at every time step so it can be used in the IDM or referenced later.

```
v_IDM(t_step) = get(Vehicle_IDM,'AttValue','Speed')/3.6; % converted
from km/hr to m/s
```

For other vehicles, it is recommended to use the GetMultiAttValues method to the data stores in all vehicles and remove the first row, as the IDM vehicle should be the first vehicle to spawn in the simulation.

```
veh_speeds = Vissim.Net.Vehicles.GetMultiAttValues('Speed');
veh_speeds = cell2mat(veh_speeds(2:end,2)); % [km/hr]
```

With these tools, one can implement their own driver model and run their model along with the other vehicles in Vissim.