2018

Analysis of Driver Behavior Modeling in Connected Vehicle Safety Systems Through High Fidelity Simulation

Ahura Jami
University of Central Florida

ANALYSIS OF DRIVER BEHAVIOR MODELING IN CONNECTED VEHICLE SAFETY SYSTEMS THROUGH HIGH FIDELITY SIMULATION

by

AHURA JAMI
B.S. National Aerospace University 'Kharkiv Aviation Institute', 2014

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2018

Major Professor: Yaser P. Fallah

ABSTRACT

A critical aspect of connected vehicle safety analysis is understanding the impact of human behavior on the overall performance of the safety system. Given the variation in human driving behavior and the expectancy for high levels of performance, it is crucial for these systems to be flexible to various driving characteristics. However, design, testing, and evaluation of these active safety systems remain a challenging task, exacerbated by the lack of behavioral data and practical test platforms. Additionally, the need for the operation of these systems in critical and dangerous situations makes the burden of their evaluation very costly and time-consuming. As an alternative option, researchers attempt to use simulation platforms to study and evaluate their algorithms. In this work, we introduce a high fidelity simulation platform, designed for a hybrid transportation system involving both human-driven and automated vehicles. We decompose the human driving task and offer a modular approach in simulating a large-scale traffic scenario, making it feasible for extensive studying of automated and active safety systems. Furthermore, we propose a human-interpretable driver model represented as a closed-loop feedback controller. For this model, we analyze a large driving dataset to extract expressive parameters that would best describe different driving characteristics. Finally, we recreate a similarly dense traffic scenario within our simulator and conduct a thorough analysis of different human-specific and system-specific factors and study their effect on the performance and safety of the traffic network.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ADAS	Advanced Driver Assistant Systems
BSM	Basic Safety Message
CAN	Controller Area Network
CCW	Cooperative Collision Warning
CFM	Car-Following Model
CSMA/CA	Carrier Sense Multiple Access/ Collision Avoidance
CVS	Cooperative Vehicular Safety system
DSRC	Dedicated Short Range Communication
ECDA	Enhanced Distributed Channel Access
ECDF	Empirical Cumulative Distribution Function
FCW	Forward Collision Warning
FIS	Fuzzy Inference System
GPS	Global Positioning System
IDM	Intelligent Driver model
ITS	Intelligent Transportation System
MAC	Media Access Control Layer
MF	Fuzzy Membership Function
NGSIM	Next Generation Simulation
NHTSA	National Highway Transport Safety Administration
PB	Periodic Beaconing

PDF	Probability Distribution Function
PER	Packet Error Rate
PHY	Physical Layer
THW	Time headway
TTC	Time-to-collision
UE	Unreal Engine
USDOT	U.S. Department of Transportation
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle

CHAPTER 1: INTRODUCTION

In 2016, the Centers for Disease Control and Prevention reported that the traffic data collected in US roadways from 2000 to 2013 show an estimate of over 32,000 motor vehicle fatalities each year [1]. Another report from the National Highway Traffic Safety Administration showed an increase of 7.7% in traffic fatalities, bringing the total number of death from 32,600 in 2014 to 35,200 in 2015 [2]. Although motor vehicle accidents constitute the highest cause of death in the USA [2], they are almost always predictable and preventable.

A preliminary step in transportation safety research is to equip every vehicle with a low-cost mechanism that would assist the driver in avoiding a dangerous situation. Depending on the level of autonomy of such safety systems, they can either warn drivers by recognizing hazardous instances or take over the vehicle's control by means of prevention mechanisms. Examples of active safety systems that provide advisory alerts include forward collision and lane departure warning. More advanced versions of these systems, known as forward collision avoidance and lane keeping assistance provide automated control in addition to recognizing dangerous scenarios. These Advanced Driver Assistant Systems (ADASs) utilize modern sensing and communication technologies to provide feedback to the driver. ADASs use the information supplied by these technologies to continuously analyze the state of the vehicle, traffic situation, and the environment [3]-[6].

Some of the most informative situational elements utilized by ADASs are vehicles' instantaneous position, kinematic information (heading, velocity, acceleration, yaw rate, etc.), driver-specific information (head pose, eye gaze, etc.), and environmental data (road type, weather

conditions, lighting characteristics, etc.), to name a few [3, 4]. In general, the least amount of data used by an ADAS can be separated into two groups: host vehicle's data and its surrounding objects. The host vehicle's positional and kinematic data can be collected from Global Positioning System (GPS) and the vehicle's Controller Area Network (CAN) bus which provides sensory information such as speed, acceleration, throttle position, and steering angle. The second type of data, i.e., remote objects' information, are obtained via local sensors such as camera, radar, and LiDAR, and through Vehicle-To-Vehicle (V2V) wireless communication. The current version of ADAS that is already deployed in some high-end vehicles employs local sensors to scan the environment and monitor the driving condition. However, these systems are costly and do not take full advantage of inter-vehicular communication. A modern vehicle, as a cyber-physical system, exploits the vehicle-to-vehicle (V2V) communication as an ideal platform for developing a rich, advanced, and inexpensive safety system (also known as cooperative vehicle safety systems [4, 5, 6]).

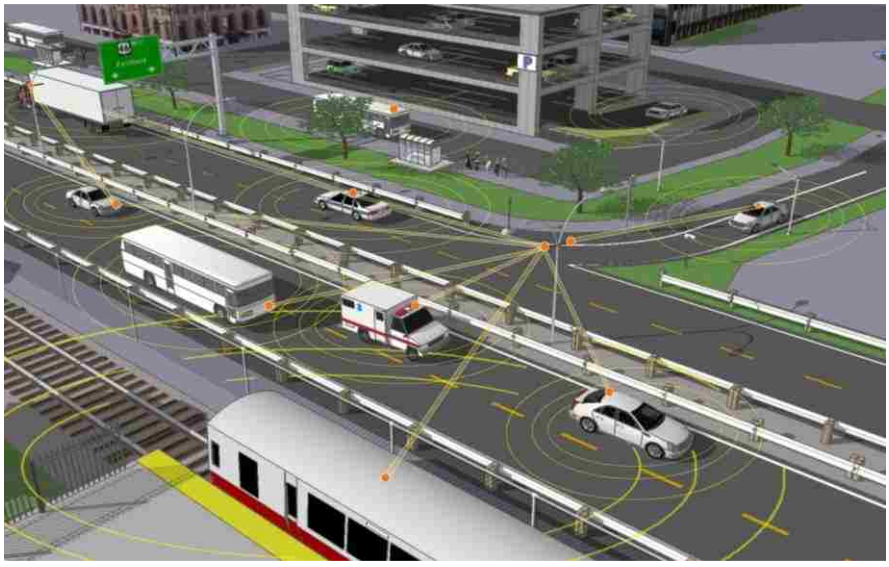


Figure 1: Visual representation of vehicle-to-vehicle communication *(Image courtesy of U.S. Department of Transportation)*

However, implementation of such automated mechanisms within a conventional transportation system brings about several technical challenges. More specifically, given the stochastic nature of host or remote vehicles' drivers, it is essential for automated systems to have a detailed consciousness of their surroundings. Moreover, to boost the performance and efficiency of such systems, it is necessary to design them within a flexible framework, taking into consideration the driver's nondeterministic behavior.

Another crucial issue that overshadows the design of an adaptive system is the need for rich and extensive driving data in which these systems are to perform. In general, the data collection phase can be very costly and time-consuming when approached through real-world measurement campaigns. Additionally, some of the most critical test scenarios can be very dangerous as they may expose real drivers to hazardous situations. For instance, to design an adaptive Forward Collision Warning (FCW) algorithm, a thorough analysis of driver behavior during a collision scenario is required. A collision scene, which could be potentially very dangerous, should be designed and executed in several trials. This crucial issue inspires the development of a versatile simulation platform, capable of performing any examination in which real drivers are not involved.

Despite transdisciplinary nature of an Intelligent Transportation System (ITS), traditional simulators were designed purely for studying traffic management topics. They offer limited capabilities for evaluating active safety mechanisms that operate in life-threatening situations. The analysis of such critical systems requires a comprehensive framework, allowing for an assessment with a sub-second accuracy under different traffic conditions.

Several attempts have been made to integrate novel ITS applications into the traditional simulators. Veins [7] is an example of a simulation framework that connects urban traffic simulator

SUMO [8] to the event-based communication network simulator OMNeT++ [9]. Despite its diverse use for simulation of vehicular ad-hoc networks, this coupled platform cannot furnish the capability for studying many innovative applications such as Cooperative Collision Warning (CCW) systems without inflicting significant changes to the software's underlying structure.

A different category of vehicular simulators has a particular focus on simulation and analysis of different algorithms on a single vehicle. The main disadvantage of such simulators is the lack of complexity of an urban driving setup which sets significant limitations on examination of an ADAS in a large-scale traffic scenario. Such traffic-related shortcomings may include the absence of intersections, traffic rules, and most importantly other vehicles. TORCS [10] is an instance of such simulators.

Recently, with the astounding rise and success of machine learning approaches, a new type of simulators is emerging that focuses on applications of machine learning, specifically deep learning, within ITS. They are, however, very similar to the category above as they too lack a comprehensive traffic setup and are mainly concerned with autonomous driving and performance of a single, fully-automated vehicle. CARLA [11], developed by researchers at Intel, and AirSim [12], developed by Microsoft, are examples of such simulators that are both open-source and are powered by Unreal Engine 4 [13].

To address these challenges, we propose a comprehensive co-simulation framework, designed exclusively for analysis of automated applications such as vehicular safety systems within a hybrid transportation network involving both human-driven and automated vehicles [14]. It comprises of carefully crafted modules including mobility and human driver models, a road transportation package, inter-vehicular communication protocols and safety systems such as FCW.

Thanks to the powerful technologies that shape its core, and with the significant rise in computational power, the proposed co-simulator is capable of representing various traffic environments with a comprehensive 3D physical response in real-time. The technical aspects of these components are discussed in the following chapters.

The rest of the thesis is organized as follows: In chapter 2 we describe the co-simulation architecture in detail and propose a driver-in-the-loop approach to simulate human-driven vehicles. In chapter 3 we discuss the details of the cooperative vehicular safety system and explain three different forward collision warning algorithm in detail. We also briefly evaluate the performance of each of these algorithms in a small near-crash scenario within our simulator. Next, we study and introduce different driving characteristics in chapter 4. In the second section of this chapter, we analyze a congested traffic dataset and perform a classification of three classes of drivers. Moreover, we extract descriptive information for different categories of drivers where we use all these information and recreate a congested traffic scenario in our simulation in section 5. Additionally, we look at the effect of communication and discuss the impact of a congested medium on the performance of cooperative vehicular safety systems. We also take a look at the results obtained in the early prototype of the simulator. We conclude this chapter by analyzing various drivers response to forward collision warnings and their impact on the performance and safety of the traffic system. Finally, the conclusion of the thesis is provided in chapter 7. We finish this work by looking at possible research directions that could use the simulator proposed in this work.

CHAPTER 2: SIMULATOR ARCHITECTURE/MODEL

As the current conventional transportation system is rapidly evolving into an intelligent one, a swift and efficient development of many novel ITS technologies require simulation-based testing and evaluation. The co-simulation framework presented in this work is built using the state-of-the-art game engine technology Unreal Engine 4 (UE4) [13] and its integrated NVIDIA PhysX Vehicle [15]. Unlike many traditional simulators where a Car-Following Model (CFM) is used to handle both vehicle dynamics and driver's characteristics [8, 16, 17], we used a modular pipeline to decompose the driving task into different interoperable components. These include a mobility model, a driver model, perception, and sensing packages, as well as a Cooperative Collision Warning (CCW) system which uses the information obtained from vehicle's local sensors and inter-vehicular communication to perceive the surrounding environment. In our simulation framework, this active safety mechanism assists the driver in avoiding dangerous scenarios by issuing imminent alerts. Figure 2 represents an overview of the proposed simulation platform and illustrates the relationship between its components. The scenario that is shown on the right side of the figure is an actual snapshot of the simulator and demonstrates vehicles equipped with CCW in a near-crash scenario [1].

The innovative approach used to simulate traffic is based on driver's vision/camera, sensors, and communication network. Data from all these sources can be used to simulate and study the

[1] A demo of the simulator demonstrating the evaluation and effectiveness of a CCW in avoiding a crash can be found at https://vimeo.com/252441087 [42]

behavior of a human-driven, semi-automated, or fully automated vehicles. The co-simulator represents two categories of human drivers: Cautious and distracted. A distracted driver is simulated by limiting the driver's field of vision and thus blinding him to surrounding environments.

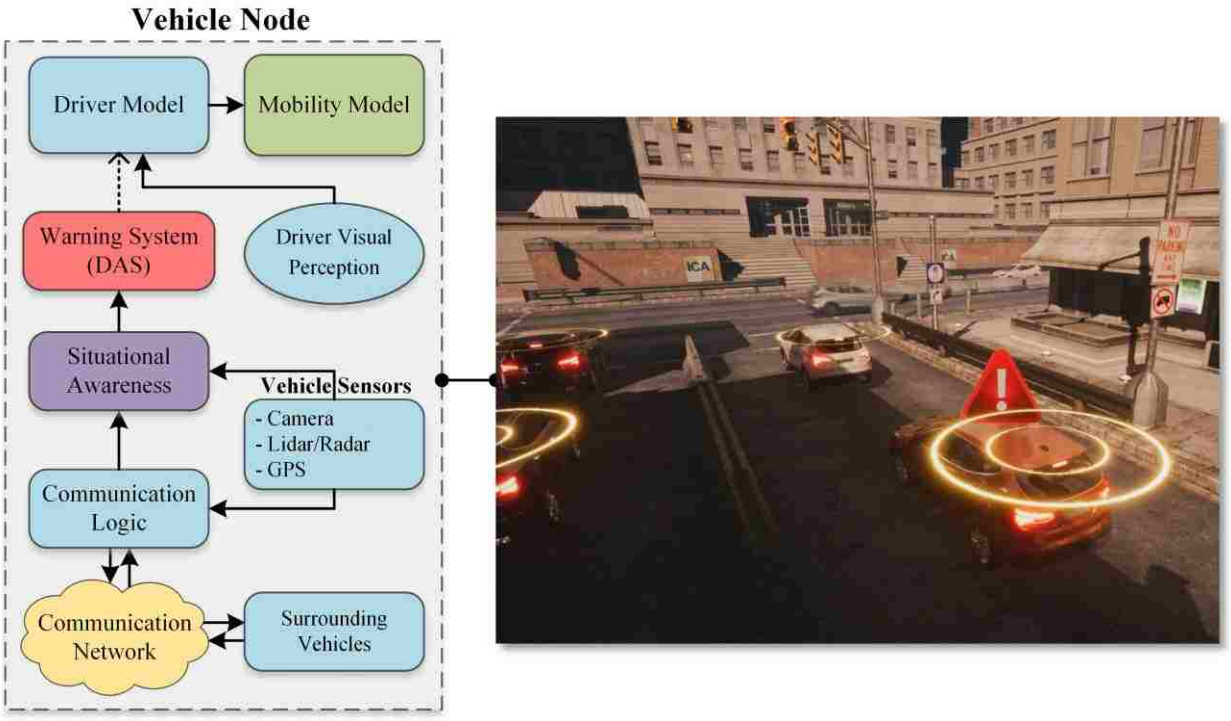


Figure 2: An architectural overview of the co-simulation framework along with a snapshot of the simulator environment

2.1 Choice of Development Environment

This project is mainly inspired and triggered by the work in [18] presented by Yaser P. Fallah and Masoumeh K. Khandani. The paper is one of the first to analyze the impact and mutual coupling of collision warning applications and vehicular communication technologies on the performance of safety systems. They performed their evaluations for several near-crash scenarios involving only two vehicles (see Appendix A), of which, the host vehicle's (leader) kinematic and

trajectory data were taken from a dataset involving several near-crash trials, while the following vehicle's movements were simulated using MITSIM car-following model [19]. This was the starting point of this work where we started expanding this module as a separate project that eventually turned into a complete co-simulator platform, designed exclusively for testing and evaluation of cooperative vehicle safety systems in a large-scale scenario. At the time, MATLAB was chosen as the preferred development environment for its rich libraries and simplicity in scripting and testing. In simulating the mobility of the vehicles, we took an approach similar to other traditional traffic simulators and used a pure car-following model. We developed a fully functional 2D simulator in MATLAB, using its object-oriented as well as undocumented features. The vehicles were simulated as a collection of 2D surfaces. Although MATLAB provided the tools to plot 2D surfaces, all other computer graphics algorithms concerned with the movement of these surfaces, including object detection, collision detection to name a few, were implemented from scratch.

Despite having fulfilled our short-term needs, we soon came to face several barriers both with the language and the 2D environment itself. MATLAB is a single-threaded scripting language designed exclusively for scientific and research purposes. It is not an ideal platform for the creation of large simulation software. Furthermore, we could see many potentials in this work and MATLAB simply could not satisfy our needs in implementing them. Additionally, we realized that a traditional approach in simulating vehicle's mobility through a car-following model presents many issues with the vehicles' physical response which are of crucial importance to studying vehicular active safety systems (this issue is further explained in chapter 5.2). Essentially, we would like to have a nanoscopic traffic simulator, integrating not only a near-realistic vehicle model but also emerging

technologies that may soon appear with autonomous and semi-autonomous vehicles. Our goal is to provide the research community with a suitable tool that would help in testing and developing the future cars. With this in mind, we looked at other options and eventually found the suitable technology and development platform – Unreal Engine 4.

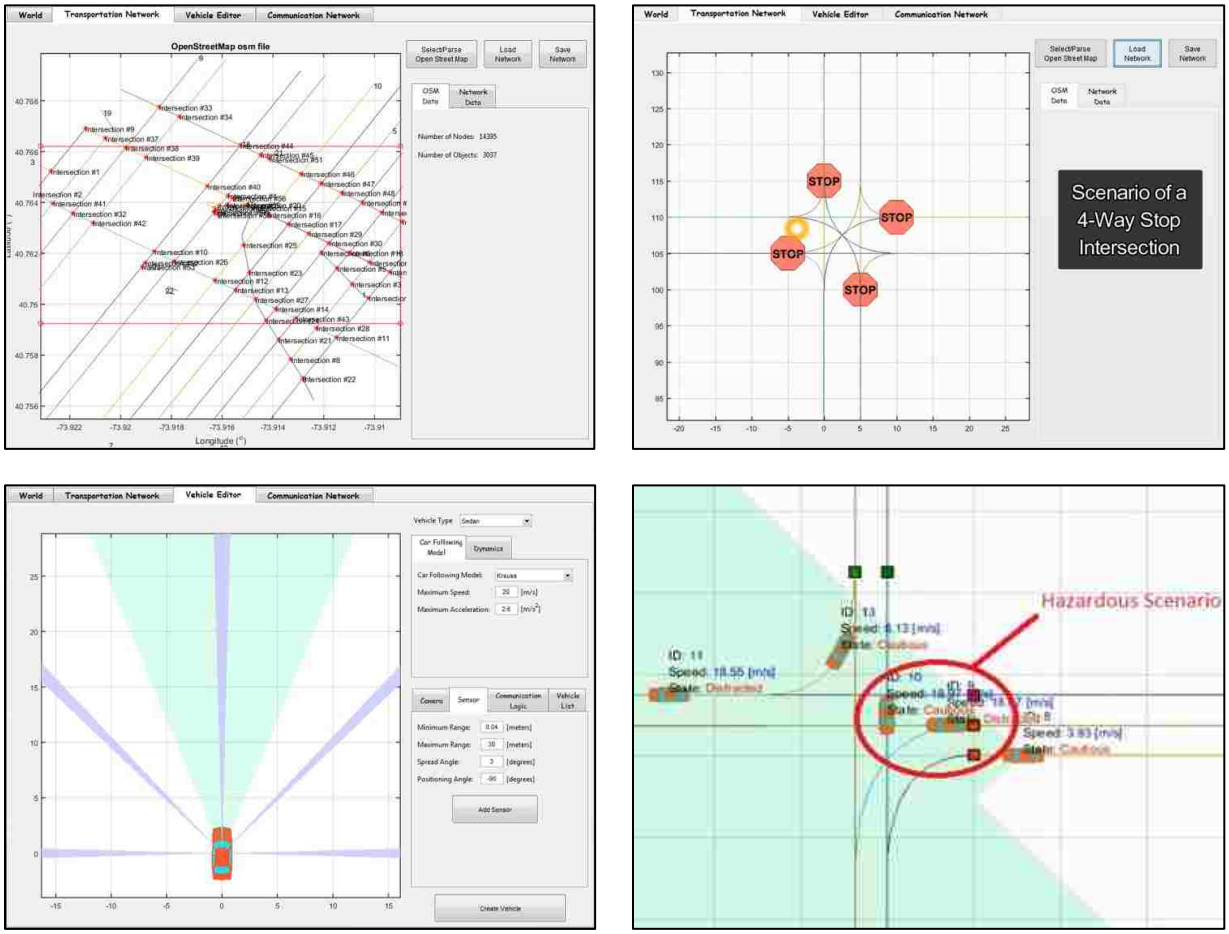


Figure 3: Snapshot of the early prototype of the co-simulator

Unreal Engine 4 (UE4) is a game engine technology. The engine has been developed and maintained by Epic Games, a leading industry in game technology development. It is open-source

and is free to use, with only a 5% royalty fee on gross product revenue after the first $3,000 per game per calendar quarter from commercial products. Many successful and visually stunning video games such Gears of War, Rocket League, etc. have been created using this engine. However, UE4 is being used for not only video game development, but also in many different areas including animation and movie industry, marketing, academics, and education, to name a few. Chevrolet uses UE4 with their blackbird car for creating advertising material. The real-time rendering that Unreal Engine provides has allowed Chevrolet to create appealing ads in a short time. NASA also uses this engine to simulate a mixed virtual reality environment of International Space Station, as well as Moon and Mars surfaces to train and prepare astronauts before launching them to actual missions. Additional projects that are worth mentioning are BMW Mixed Reality Vehicle, and MARS Field Trip that were both done with the Unreal Engine.

Hence, we can summarize our points for landing on Unreal Engine 4 in the following:

1. Its superior and epic real-time rendering capabilities
2. Fast and efficient development with its premade libraries and the node programming interface, i.e., blueprints.
3. Our previous experience with C++ and hence, the availability of many other libraries third-party libraries
4. Its open source code for the entire engine
5. Convenient profiler and debugging tools
6. Multi-threaded capabilities
7. Its complete 3D physics engine, including out-of-box vehicle dynamics provided with NVIDIA PhysX Vehicle

8. Compatibility and support for the emerging technologies such as virtual reality
9. Availability of many premade digital assets on the marketplace
10. Supportive community

2.2 Simulation Environment

The co-simulator delivers a dynamic world with an easy-to-use graphical user-interface. A virtual environment is comprised of 3D static and dynamic objects such as buildings, roads, traffic signs, vegetations, vehicles, and infrastructure. A user can build an urban traffic network by laying out roads topologies and manually placing buildings, vegetation, terrain, etc. They can also set the time of the day to either a sunny/cloudy day or night sky. Furthermore, the simulator is capable of directly importing a map from OpenStreetMap. Lastly, the simulator allows for importing of different types of vehicle. In addition to the material and color of each vehicle, their dynamical characteristics such as engine torque and transmission setup can be changed as well. Thanks to UE4's easy-to-use interface, and with additional custom-built tools and features, a user can effortlessly customize and build a desired driving environment.

The road and transportation network of the co-simulator are designed based on the recommendations provided in OpenDrive [20]. OpenDrive is a vendor-independent set of standards that is largely supported by simulation industry and is maintained by a team of simulation experts. It contains all the key features of the road and transportation topology. These recommendations are implemented as a set of waypoints and splines within our simulator (Figure 4). A user can use a premade 3D urban environment or create a new one that includes road layouts. The waypoint system

can then be laid upon this environment which would enable the vehicles to be spawned and navigate through it. Since the layout of the transportation network is independent of the analysis performed on the connected vehicle safety systems, we conclude our discussion here and encourage the interested parties to refer to the OpenDrive standard for details on the implementation of a transportation network.

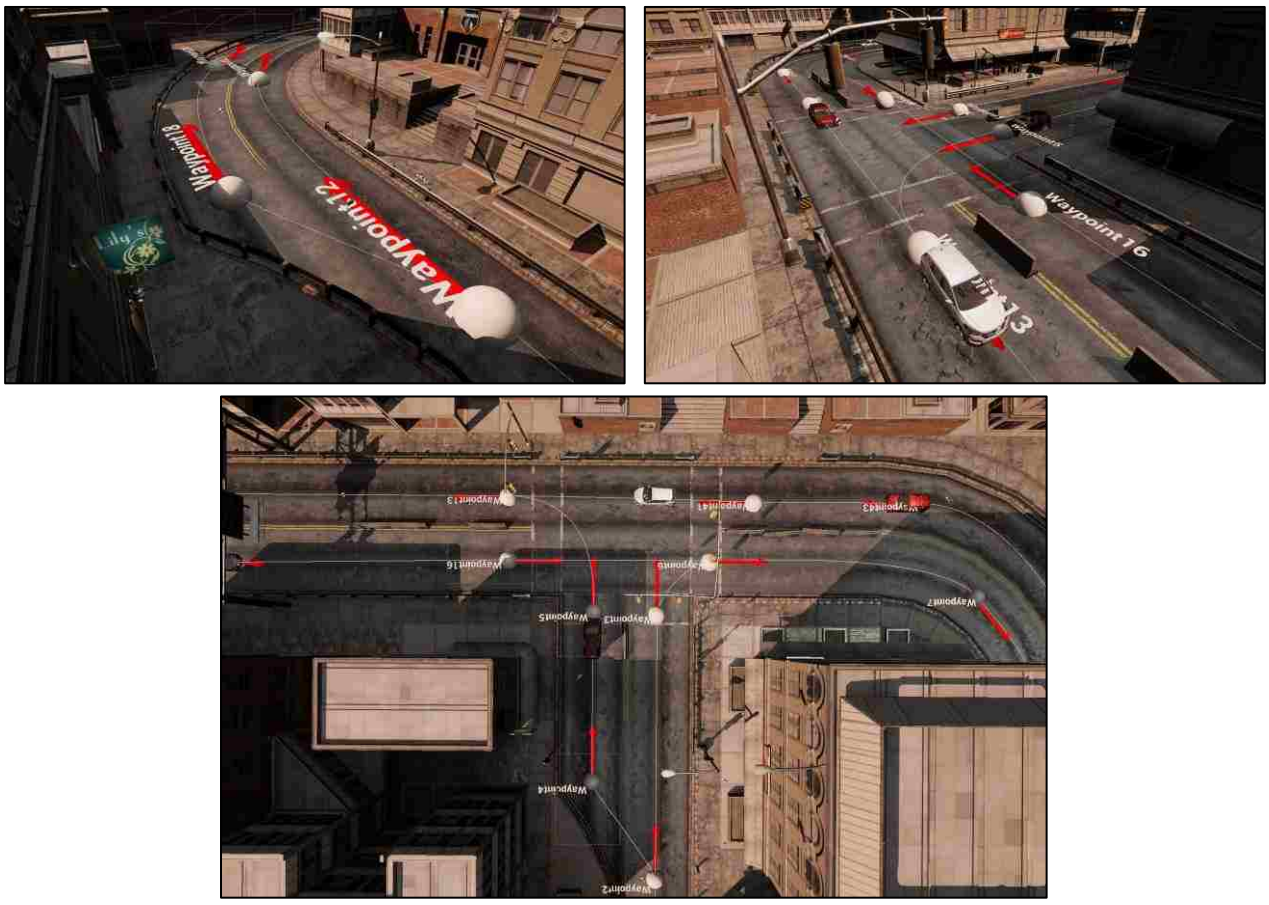


Figure 4: Snapshots of co-simulator's environment and its transportation network topology

2.3 Mobility Model/Locomotion Layer

The mobility model is responsible for handling the kinematics and dynamics of a vehicle and its subsystems. This module translates the values of throttle and steering angle supplied by the driver into lateral and longitudinal movement of the vehicle. We use the UE4 standard physics engine and its integrated component, NVIDIA PhysX Vehicle, to handle physical characteristic of the vehicles in 3D environments. The NVIDIA PhysX models a vehicle as a rigid body on a collection of sprung masses [15].

The inclusion of vehicle dynamics plays a vital role in testing and evaluation of any ADAS system that operates in a sub-second level. Past few decades have seen a great progress in theory and experiment of vehicle dynamics. In vehicle dynamics, the essential parts of the system are the vehicle body (sprung mass), the suspension component (spring and damper) and tire (unsprung mass). These collections of sprung masses can be represented as a rigid body whose mass, center of mass, and moment of inertia exactly matches the masses and coordinates of the sprung masses.

The PhysX Vehicle SDK works closely with PhysX SDK. The PhysX Vehicle SDK computes tire and suspension forces using the sprung mass model and then applies the aggregate of these forces in the form of a modified velocity and angular velocity to PhysX SDK rigid body. PhysX SDK then manages the position update and the interaction of this rigid body with other scene objects.

From the UE4 perspective, a vehicle is implemented as a wrapper around NVIDIA PhysX Vehicle. Through adjusting the vehicle's mechanical setup such as engine torque and differential setup, and by incorporating additional components including tires model and road friction, the

complete UE4 physical engine allows for the near-realistic dynamical response of vehicles within its virtual environment (Appendix B).

2.4 Driver Model

The driver model is essentially the brain of the system, representing a human driver and is tasked with controlling the vehicle while adhering to traffic laws. During a longitudinal car-following regime, a driver controls the vehicle by adjusting the throttle and brake pedals to maintain a safe speed and distance from the leading vehicle. According to control theory, this interaction between the driver and the vehicle forms a closed-loop feedback control system, thus, representing the driver as a feedback controller [21]. As the most popular type of controller in the industry, PID controllers are often used to describe robot driver models [21, 22]. Despite their popularity and ease of implementation, solely PID-like models cannot characterize ideal human drivers as they are merely based on the speed/acceleration errors and do not consider driver-specific characteristics. A more accurate representation of a driver should consider the nonlinearities in human response time, as well as his/her specific decision processes.

Given that every driver has his/her own driving style, we propose a mixed driver model employed through a CFM-Fuzzy-PD closed-loop feedback controller. The combination of these models enables us to incorporate the most beneficial attributes of each subsystem and lead it into a natural and human interpretable system that is suitable to embody a driver-in-the-loop model [23]. The CFM perceives the information and recommends a target acceleration. The Fuzzy-PD components then decide how fast to adapt to this supplied acceleration value. A Fuzzy Inference

System (FIS) can be a potential candidate in dealing with the uncertainties in the car following regime, mapping the variables that the driver perceives to the variables he/she can control with arbitrary accuracy based on fuzzy reasoning. The Partial-Derivative (PD) component stabilizes the response time and guarantees the performance of the controller.

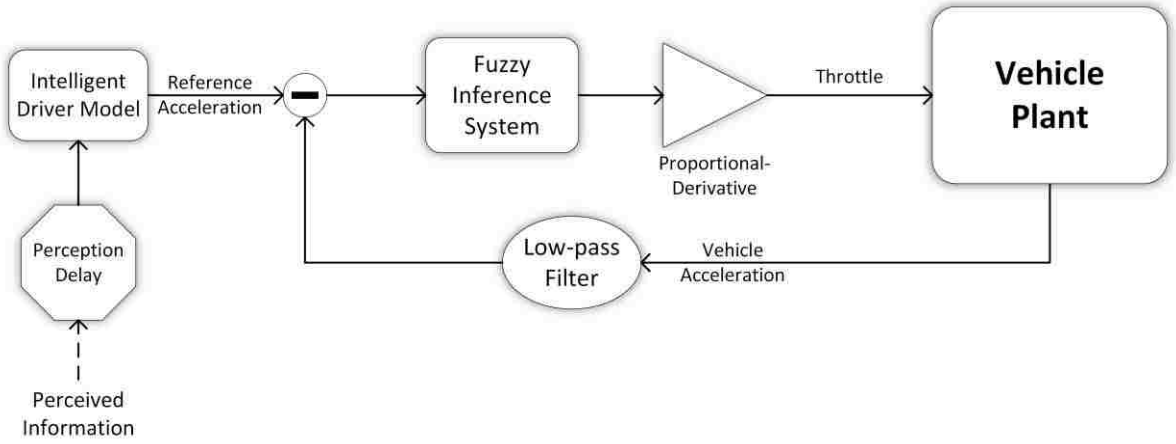


Figure 5: The Fuzzy-PD control structure used to represent a human-driver

We start the discussion of our driver-in-the-loop system by first looking at the Intelligent Driver Model (IDM) component. IDM is a microscopic model of traffic flow [24], which, in our system, supplies the reference acceleration $a_i(t)$ to the driver's fuzzy component. The objective of a microscopic traffic model is to describe the dynamics of each vehicle as a function of positions and velocities of its neighboring vehicles [25]. The output of such model, which can be either the velocity or acceleration of the vehicle, can be interpreted as the intended velocity/acceleration that the driver would like to have while following other vehicles in a smooth and safe driving regime. The IDM was designed by Treiber et al. [24], which is also implemented in several microscopic traffic simulators such as MovSim [17] and SUMO [8]. The model is formulated as follows:

$$v_i(t) = \frac{dx_i(t)}{dt} \tag{1}$$

$$a_i(t) = \frac{dv_i(t)}{dt} = \alpha\left(1 - \left(\frac{v_i(t)}{v_0}\right)^{\delta} - \left(\frac{G(t)}{s_i(t)}\right)^2\right) \tag{2}$$

$$\text{with} \quad G(t) = s_0 + v_i(t)\mathrm{T} + \frac{v_i(\mathrm{t})\Delta v_i(t)}{2\sqrt{ab}} \tag{3}$$

where $x_i(t), v_i(t)$, and $a_i(t)$ represent position, velocity and acceleration of host vehicle at time t respectively; α is the comfortable acceleration of the driver; v_0 is the desired velocity; s_0 represents the minimum gap; $s_i(t) = x_{i-1} - x_i - l_{(i-1)}$ is the net distance between host vehicle i and leader vehicle $i-1$; T is the minimum desired safe time headway to the vehicle in front; $b > 0$ represents comfortable braking deceleration and $\Delta v_i(t) = v_i - v_{i-1}$ represents the range rate, i.e., the velocity difference of the host and leader vehicles.

Although IDM incorporates some of the most important driver characteristics, it merely supplies the reference acceleration $a_i(t)$ as a desired acceleration value that the driver would like to reach. It is the responsibility of the rest of the controller to determine how fast the driver would like to reach that value. Given the availability of a complete dynamical model of a vehicle, the Fuzzy-PD part of the controller attempts to map these supplied acceleration values to the changes in throttle and brake pedals. The statistical analysis in [26] shows that the integral of throttle position over time has a direct relationship with fuel consumption, and thus needs to be considered in modeling and identification of human drivers. However, due to lack of a reliable and publicly available traffic data that involved throttle measurement, we took a heuristic and human interpretable approach to construct the Mamdani FIS component. Hence, the FIS here is designed

to be as simple yet as interpretable as possible, with only one input $\Delta a_i(t)$, denoting the difference in accelerations and one output δp, representing changes in accelerator or brake pedal positions. The seven linguistic variables of this Mamdani FIS are shown in the Figure 6. The fuzzy rules are characterized in a way that the normalized Membership Functions (MF) are within the interval $[-1, 1]$. These values are then mapped to the maximum acceleration and deceleration values of the vehicle itself. The linguistic variable $\Delta a_i(t)$ has seven linguistic values (large negative, medium negative, small negative, zero, small positive, medium positive, large positive). Similarly, the fuzzy output δp has seven linguistic variables as well. The MFs' characteristics of these linguistic variables are represented in Figure 6.

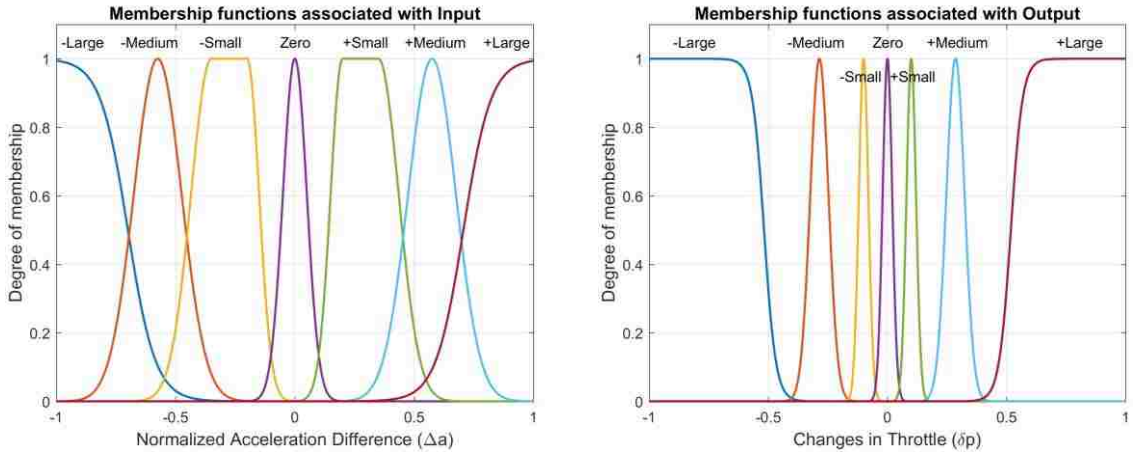


Figure 6: Membership functions associated with inputs (left) and outputs (right)

A simple inference rule system derives the fuzzy command of FIS by mapping the linguistic values of the input to linguistic values of output one-to-one:

- If $\Delta a_i(t)$ is Large Negative then δp is Large Negative.
- If $\Delta a_i(t)$ is Medium Negative then δp is Medium Negative.

- If $\Delta a_i(t)$ is Small Negative then δp is Small Negative.
- And so on.

Figure 7 shows the output plot of the FIS for the MFs' characteristics that are represented in Figure 6.

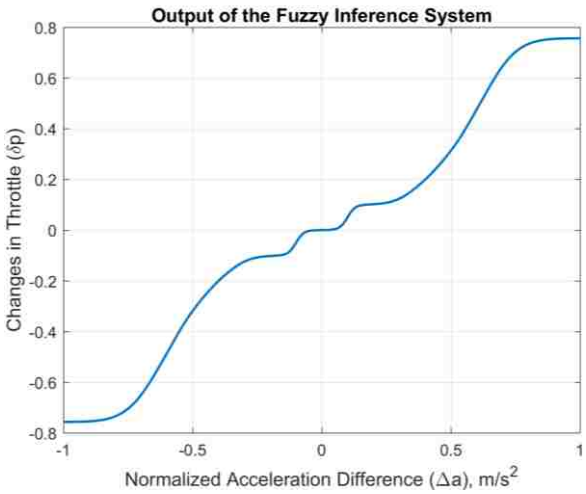


Figure 7: Output of the fuzzy inference system for the above membership functions

The intersection of the fuzzy sets are defined using a product rule:

$$\Delta a \cap \delta p = \{x | \mu_{\Delta a \cap \delta p}(x) = \mu_{\Delta a}(x) \cdot \mu_{\delta p}(x)\} \tag{4}$$

Finally, the fuzzy command obtained from the intersection of the fuzzy sets is transformed into a crisp command u^* using the center-of-sum defuzzification method of fthe ollowing form:

$$u^* = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} A_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}} \tag{5}$$

where c_{ij} is the center of the MF obtained from the intersection of MF_i and MF_j. A_{ij} is the area of this obtained MF.

These input and output MFs are tuned intuitively to represent different driving behaviors. It is found that the steeper the slope of the curve in Figure 7, the faster and more aggressively a driver would respond to acceleration changes dictated by IDM.

Many car-following models including IDM do not consider the driver perception-reaction time (time taken by a driver to sense the stimulus and start reacting by either accelerating or decelerating) [25]. Perception-reaction time is a driver-specific parameter and can be affected by many implicit and explicit factors such as driver's mood, drowsiness, comfort level, vehicle condition, and weather condition. Olson and Sivak [27] conducted a study on human reaction time and found that the average perception-reaction time for 95th percentile was 1.6 seconds. Similarly, we adopt a constant perception-reaction time of 1.4s and a constant pedal switch time of 0.2s.

The final component of this feedback control system is a low-pass filter. Since a vehicle's dynamical behavior is represented by a complete 3D physics model, different factors, other than throttle value, affect the acceleration of the vehicle. These include tires and suspension models, as well as road characteristics. The moving average filter adapted here helps in reducing the noise artifacts of vehicle's acceleration value to help stabilize the system's response. This filter is represented by the following expression:

$$\bar{a} = \frac{a_{t-n-1} + a_{t-n-2} + \cdots + a_t}{n} = \frac{1}{n}\sum_{i=0}^{n-1} a_{t-i} \tag{6}$$

where a_t is the latest value of the acceleration of the vehicle and n is the number of previous successive samples.

Lastly, we disintegrated the IDM into individual states and introduced a driver state parameter which identifies the appropriate state of the CFM that the driver needs to be in. The

driver's state is determined by information perceived from different sources including vision and safety alerts. Driver's vision is represented by a field-of-view, allowing to sense the objects within it and scheduling an appropriate action accordingly (section 2.5). These actions are in fact representative of states that the driver can enter. In addition to vision, factors such as safety warnings may trigger a state transition and hence, force the driver into a new state, i.e., emergency reaction state. The main driving tasks in the form of a state machine are represented in Figure 8.

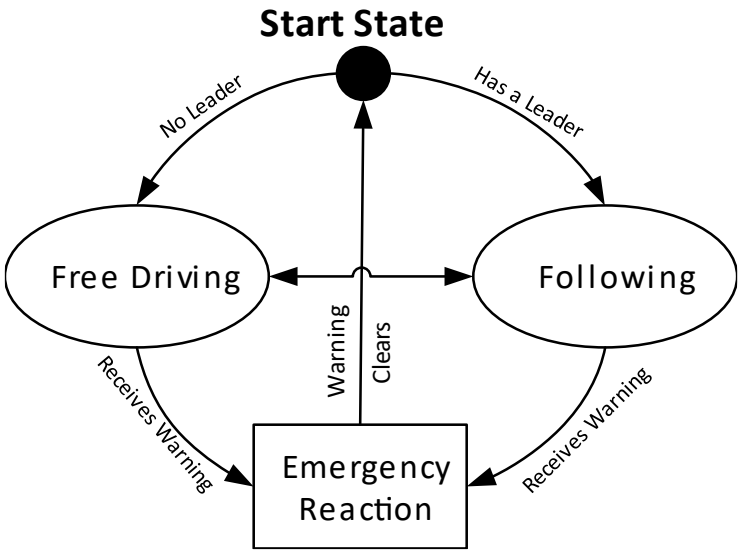


Figure 8: Main driving tasks represented through a state machine

The Emergency Reaction state helps study the potential role of a specific crash avoidance system in preventing near-crash and actual crash scenarios. The study in [3] shows that braking was the initial reaction choice of many drivers to an FCW alert. In fact, the primary objective of any collision avoidance algorithm is to understand when braking is needed to avoid a hazardous scenario.

A visual representation of the changes in the vehicle's kinematic data using the proposed driver-in-the-loop model for each state of the IDM (Figure 8) is represented below. In the first two figures (Figure 9 and 10) the free flow and following states are also compared with the case of traditional simulators approach, i.e., based on a pure CFM. The third figure (Figure 11) shows the response of the host vehicle in a near-crash scenario where the leading vehicle displays a sudden braking behavior. It is worth mentioning that the parameters of CFM-Fuzzy-PD system are tuned to represent a moderate driving behavior.

In addition to the main driving events, many other important decision processes are also considered. They define continuous behaviors such as adapting to and classifying vehicles into leader, follower and surrounding vehicles, adapting the behavior according to a traffic regulation signal, and also performing the lane change decisions. Due to their nature, these tasks are state independent and are run in parallel along the primary driving task. In the UE4 context, these behaviors are governed by a behavior tree, the nodes of which represent a state the driver can be at any time. The behavior tree used in this model is slightly different from the traditional behavior trees. Upon a state change, the executing branch may fail at any upper level that determines the state change. In such a scenario, the system will pick another branch rooted at the affecting level which is expected to be valid at the time, hence representing the new state the driver must be at.

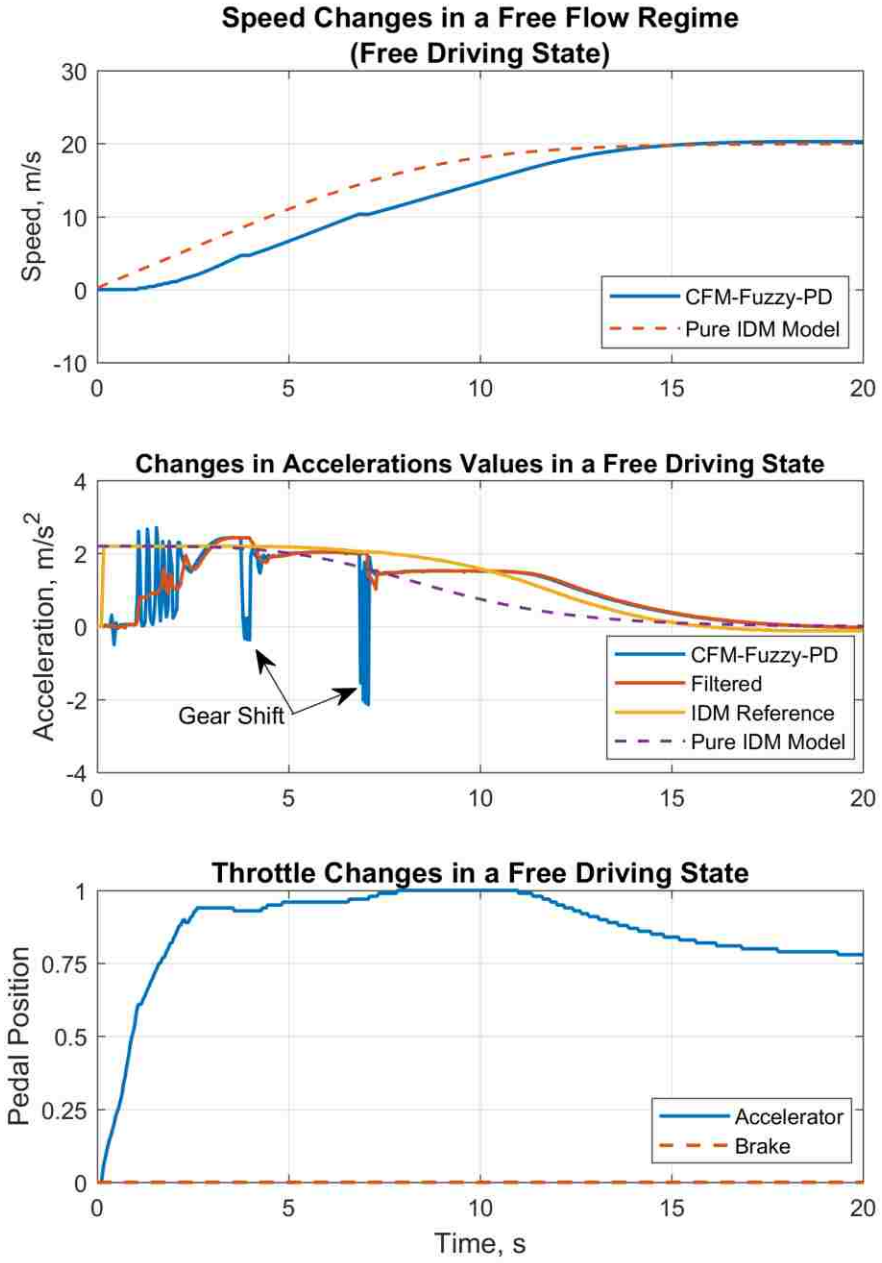


Figure 9: A visualization of the kinematic responses of the proposed driver-in-the-loop model for Free Flow driving state

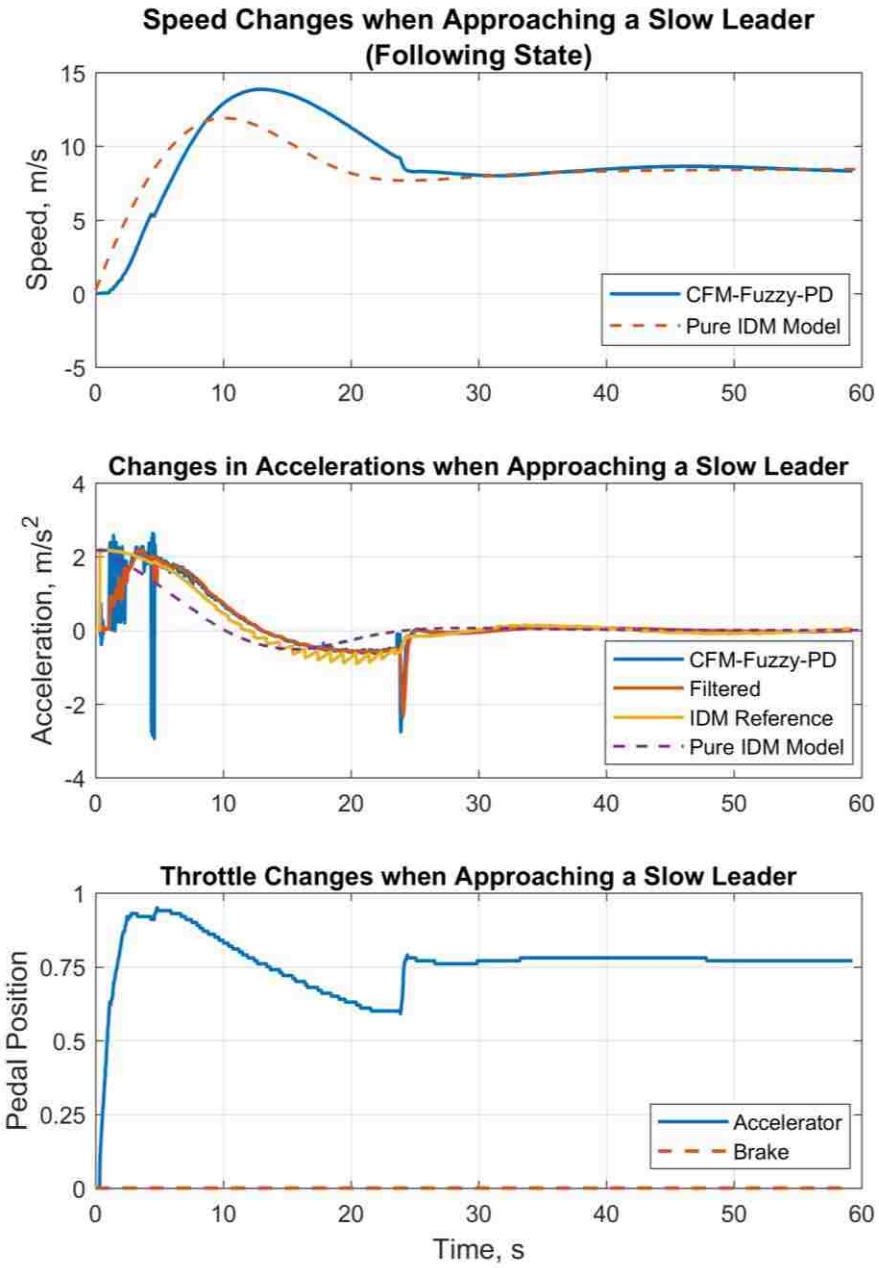


Figure 10: A visualization of the kinematic responses of the proposed driver-in-the-loop model for Following driving state

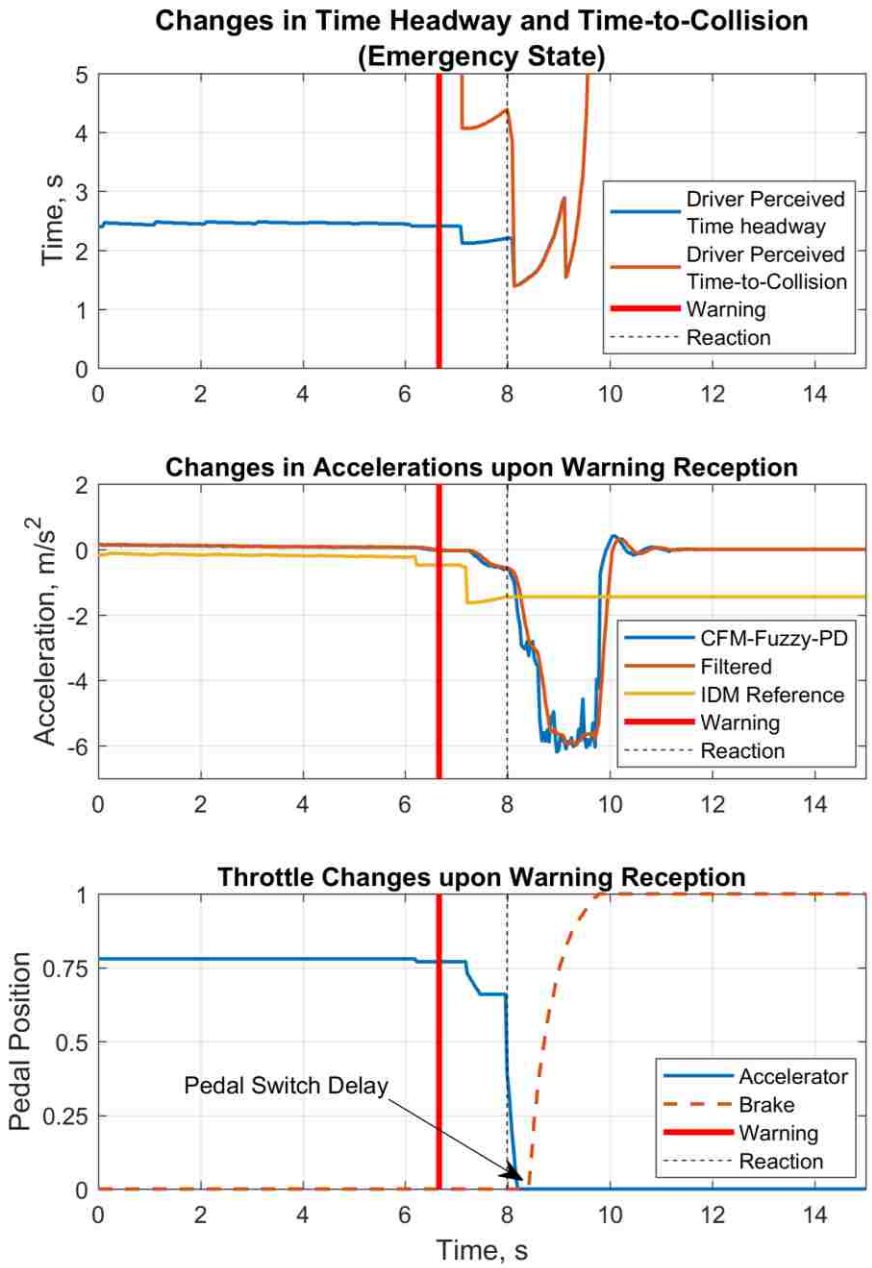


Figure 11: A visualization of the kinematic responses of the proposed driver-in-the-loop model for Emergency state

In conclusion, the introduction of a separate driver model in such a modular manner paves the way for a straightforward, human interpretable system that can be easily tuned to represent different classes of drivers. Additionally, it breaks down the traditional one-to-one approach of CFMs and allows for the introduction of different classes of drivers

2.5 Driver's Visual Perception

Driver's visual perception play an integral part in simulation of dangerous scenarios. We simulate the visual behavior of a driver through a cone-shaped field-of-view. A driver sees and senses the objects that are within his field-of-view and ignores whatever that is outside. Depending upon how far ahead the driver can see, he/she tries to adapt to a leading vehicle (if any), follow the traffic regulation signs such as speed limits, traffic control signals, or yield/stop signs, and schedule a proper action for the next iteration. Altering the field-of-view, we can introduce two types of drivers, a cautious driver, and a distracted one. A Cautious driver is one who pays special attention to his surrounding, while a distracted one has a limited vision range and as a result can cause collision before he can ever react (more on this in chapter 4). Following figures illustrate how the vision is represented in both the early prototype and current iteration of the simulator in UE4.

Figure 12: Simulation of the visual behavior of a driver through a field-of-view in early MATLAB prototype (left); and in the current iteration of the simulator (right)

The detection is most obvious in the right picture in Figure 12. The selected host vehicle (highlighted by yellow) successfully sees only the vehicle in front of it (seen object indicated by a green sphere at their root bone) and is blinded to the vehicle behind it (no green sphere). Next, an algorithm attempts to identify the leading vehicle by checking for their heading and lane number. It is also worth mentioning that the vision update happens in parallel with other driving events with a frequency of 2Hz.

CHAPTER 3: COOPERATIVE VEHICULAR SAFETY SYSTEM

In a Vehicle-to-vehicle (V2V) communication network, Cooperative Vehicular Safety (CVS) systems operate based on the information broadcasted between vehicles in a Dedicated Short-Range Communication (DSRC) medium. A safety algorithm uses the communicated data between vehicles to survey the thread zones to discern any possible danger. When a vehicle performs local sensing, it transmits its currents state, as well as the information perceived by local sensors to neighboring vehicles through DSRC communication. As the other vehicle receives these data, it constructs a map of its surrounding. Next, a Cooperative Collision Warning (CCW) algorithm scans the hazardous zones in this map every 100ms for any dangerous situations. If an imminent threat is detected, the system will warn the driver (or in the case of automated systems, the vehicle will maneuver itself out of the danger by means of automated mechanisms). The general architecture of an inter-vehicular CVS system is shown in Figure 13.

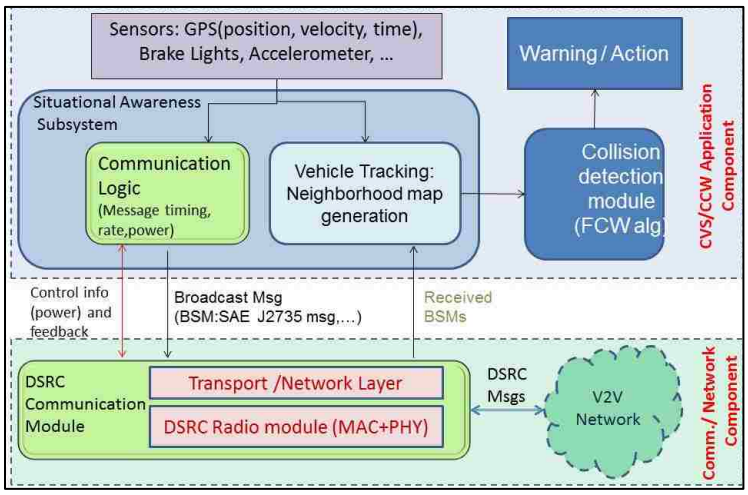


Figure 13: Architecture of inter-vehicular cooperative vehicular safety system *(Figure borrowed from [18] with permission from authors)*

3.1 Dedicated Short-Range Communication

Among the suggested communication protocols, the Dedicated Short-Range Communication (DSRC) [28] is the leading standard candidate for inter-vehicular communication. From a technical perspective, vehicle-to-vehicle (V2V) communication requires a fast, safe and interoperable connectivity that would enable cooperative collision safety systems and other non-safety services. Additionally, the V2V and Vehicle-to-Infrastructure (V2I) communication must be able to operate in any environment with a short delay through a secure and safe wireless interface. Given these requirements, i.e. time sensitivity of active safety systems and the need for low latency in their communication protocols, DSRC has been specifically designed for ITS applications as a fast-response and scalable solution. In principle, DSRC is similar to Wi-Fi technology, however, DSRC is preferred over Wi-Fi because the number of devices that use unlicensed Wi-Fi channels would make significant interferences that could hamper the safety concerns of vehicular applications. DSRC can be used in a variety of applications such as:

- Emergency warning systems for vehicles
- Cooperative cruise control
- Cooperative Forward collision warning
- Intersection Collision warning
- Electronic parking payments
- Electronic toll collection, and many more

It is also worth mentioning that mandatory deployment of DSRC devices on all vehicles is being widely debated at U.S. Department of Transportation.

3.2 Components of a Cooperative Vehicular Safety System

DSRC messages are usually sent at a fixed rate of 10Hz (there are, however, ongoing research for a dynamic transmission rate based on the scenario and congestion of the network). A communication interface may suffer from scalability issues when there are many vehicles in its vicinity. Therefore, the congested environment results in many of the safety messages being lost and the application accuracy and performance degrade significantly to the degree that it is no longer acceptable. To overcome this problem, researchers have initially attempted to design a system that would maximize the message reception probability. However, a different perspective indicates that what a safety application essentially needs is the current state of the neighboring vehicles. With such a view, the performance of the system could be described in terms of minimizing the estimation error of vehicles' positions. This perspective is closer to the reality of a safety application and is, therefore, becoming the de facto standard in viewing how CVS operates. Figure 14 demonstrates the relationship between different components of a CVS that follows this view.

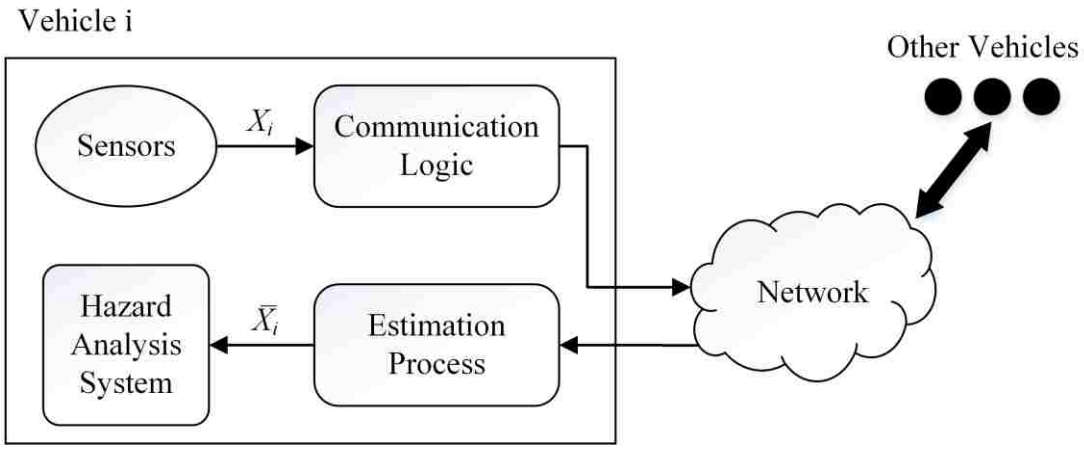


Figure 14: An abstract architecture of inter-vehicular cooperative vehicular safety system

In what follows, we describe these components and their effect on the performance of the communication network.

3.2.1 Communication Logic

This component represents the communication subsystem on the host side. It determines the transmission power and frequency and dictates how the recourses of the network should be used to broadcast the next Basic Safety Message (BSM) [2]. The choice of the communication parameters directly affects the reception probability of BSM packages within a V2V environment [29, 30]. The fundamental transmission rate in Cooperative Vehicular Safety Systems (CVS) is 10Hz [31].

3.2.2 Error-Dependent Estimators

This module deals directly with the tracking of vehicles whose information are received over the communication medium. Upon reception of BSMs by host vehicle, a local map is constructed from these communicated data which will keep track of surrounding vehicles. Various methods have been proposed to minimize the tracking error. A few notable ones include simple models such as first and second order kinematic models (constant speed and constant acceleration), to more complex ones that are based on Kalman Filter. Nevertheless, it has been shown that in a longitudinal car-following regime, a second order kinematic model can perform with high accuracy [32]. Consequently, the constant acceleration of the following form is also the model we adopt in

[2] A basic safety message holds essential vehicle's state information such as its location and kinematic data [30]

our framework. The location update based on constant acceleration is calculated based on the following formulas:

$$v_{j,next} = \check{v}_j + \check{a}_j \cdot \Delta t \tag{7}$$

$$\Delta s = \frac{v_{j,next} + \check{v}_j}{2 \cdot t} \tag{8}$$

$$\check{v}_j = v_{j,next} \tag{9}$$

$$\check{s}_{j,x} = \check{s}_{j,x} + \Delta s \cdot cos\left(\check{H}_{j,yaw}\right) \tag{10}$$

$$\check{s}_{j,y} = \check{s}_{j,y} + \Delta s \cdot sin\left(\check{H}_{j,yaw}\right) \tag{11}$$

where j is the index of the j-th nearby vehicle; $\check{v}_j, \check{a}_j, \check{H}_j -$ communicated values for the velocity, acceleration, and heading of the nearby vehicle j; $v_{j,next}$ is the updated value of the speed of the j-th vehicle based on constant acceleration model; $\Delta t = 0.1s$ is time step between each update; $\check{s}_j$ represents the coordinate locations of the vehicle j. The speed and location are the network estimates of the vehicle j that are updated every 100ms until the next BSM packet with new and updated information arrives. It must also be noted that the host vehicle who receives and decodes a BSM always has precise information about its own state (position, speed, acceleration), but only an estimate of other vehicles' states.

3.2.3 Forward Collision Warning Algorithms

The hazard analysis system in Figure 14 implements collision detection algorithms such as Forward Collision Warning (FCW). FCW systems are one of the most crucial collision prevention mechanisms. Statistical analysis of accident data shows that rear-end crashes constitute 25.2% of

all crashes occurring on highways [33]. This indicates that FCW systems have the potential of saving almost 8000 lives each year in the USA alone.

FCW algorithms are designed to alert the driver when the movement pattern of their vehicle and the leading vehicle yield an imminent crash. The timing of these alerts should take into account the perception and reaction time of the driver. However, the alert timing should not be too soon to cause false alerts or too late that at which the driver has already taken a pervasive action on his/her own. Striking such a balance is a difficult task and has been the subject of numerous studies [34, 35, 36, 37, 38, 33, 39].

Currently available FCW systems rely on local sensors such as radar, laser, or camera to detect immediate neighbors of the host vehicle, whereas in a Cooperative Collision Warning (CCW) system, this local information can be replaced or fused by information perceived via communication through BSMs from all surrounding vehicles up to a few hundred meters away (removing the line of sight requirement). Therefore, a cooperative vehicular safety systems can provide an inexpensive way of implementing FCW systems in existing vehicles. Furthermore, BSMs contain information that is usually not available through other sensors and can be used for more complex predictions (such as heading changes and brake information)

Several FCW algorithms are being tested and used by different auto manufacturers. Unfortunately, most auto manufacturers do not disclose the details of their safety algorithms, limiting us to only those that are available to academic and research communities. We analyzed the performance of a few such FCW algorithms, namely Knipling et al. algorithm [36], CAMP Logistic Regression [34], and a Driver-tuned FCW algorithm from NHTSA, designed by Brunson et al. [40]. However, the models and evaluation methodology presented here applies to any FCW algorithms,

since any of these algorithms essentially determines when a prevention action needs to be taken to avoid a collision. In what follows we will look describe each of these algorithms in more detail:

1. Knipling et al. – Algorithm developed by Knipling et al. for two scenarios of stationary and moving leading vehicle (Appendix A) [13]. The important parameter in designing this algorithm is the gap distance between the following and leading vehicles. In designing their FCW algorithms, the authors took a simplified Monte Carlo simulation approach for two scenarios of stationary or moving leading vehicle. The equations describing each of the considered scenarios are given below:

   a. Lead vehicle stationary ($\check{v}_i = 0$):

$$D_w = \frac{T_f^2}{2a_i} + TDv_i \tag{12}$$

where D_w is the dynamic warning distance; $v_i -$ velocity of the following (host) vehicle; a_i is the deceleration of the following vehicle; T_d represents total time delay of the driver of the following vehicle before a full response is executed (~2.05 seconds). This expression simply represents the required separation distance D_w that allows the driver of the following vehicle to react and decelerate to complete stop right before the stationary obstacle.

   b. Lead vehicle moving ($\check{v}_{i-1} > 0$):

$$D_w = \frac{v_i^2}{2v_i} + T_d v_i - \frac{\check{v}_{i-1}}{2\check{a}_{i-1}} \tag{13}$$

where the new two variables $\check{v}_{i-1}$ and $\check{a}_{i-1}$ represented the speed and acceleration of the leading vehicle. The hat over these variables indicates that these are estimated values of the leading vehicle available at the host vehicle's side that arrived through DSRC network. This scenario only considers a decelerating leading vehicle. Therefore, the warning distance D_w represents the necessary

separation distance such that after both vehicles decelerated to a complete stop, the following vehicle ends up right behind the lead vehicle.

2. CAMP Logistic Regression (Inverse TTC-Based Approach) – The logistic regression approach that was designed from statistical analysis of naturalistic crash and near-crash data by the research consortium of several vehicle manufacturers (CAMP) [35]. Two fundamental driver behavior parameters considered in the CAMP algorithm are driver deceleration and his/her reaction time parameters. These parameters are fed into the algorithm which determines the necessary warning range in assisting the driver in avoiding a potential crash. In statistical modeling of CAMP logistic regression, both normal and hard braking data were used to determine a binary outcome. In this algorithm, the alert timing, described as "warning range," is designed to issue an alert immediately before it becomes necessary for the driver to take action. A collision warning is issued to the driver of the following vehicle when the estimated distance between the leading vehicle and the following vehicle becomes less than the warning range. The equations from which the warning range r_w is obtained are summarized below. The algorithm uses the communicated kinematic data of the host and its leading vehicle, i.e. $v_i, a_i, \check{v}_{i-1}, \check{a}_{i-1}$. Having these information, the r_w is calculated as the sum of driver reaction and system delay range r_d and brake onset range BOR

$$r_w = r_d + BOR \tag{14}$$

with

$$r_d = \Delta v \cdot t_d + \frac{1}{2}\Delta a \cdot t_d^2 \tag{15}$$

$$\Delta v = v_i - \check{v}_{i-1}; \qquad \Delta a = a_i - \check{a}_{i-1} \tag{16}$$

where t_d is the total delay that is comprised of driver and braking system reaction delays. The general equation for brake onset range is:

$$BOR = \frac{b \cdot \left(v_{pred,i} - \check{v}_{pred,i-1}\right)}{\ln\left(\frac{1}{p^*} - 1\right) - a - c \cdot v_{pred,i}} \tag{17}$$

where $v_{pred,i} = v_i + a_i t_d$ is the predicted value of the host vehicle speed after a delay of t_d. The $\check{v}_{pred,i-1}$ is the predicted speed of the lead vehicle that obtained similarly with the communicated data from BSMs. p^* is the probability value of braking onset that can be theoretically selected by the designer. A reasonable value for p^* is considered 0.75. For the 3-tiered inverse TTC approach, the value of brake onset range differs for three different lead vehicle kinematic behavior, i.e., stationary, moving with zero or positive acceleration, and decelerating. Hence, the other parameters in equation (17) are given below

a. If lead vehicle is moving and braking ($\check{a}_{i-1} < 0$)

$$\begin{cases} a = 6.092 \\ b = -18.816 \\ c = -0.0534 \end{cases}$$

b. If lead vehicle is moving but not braking ($\check{a}_{i-1} \geq 0$)

$$\begin{cases} a = 6.092 \\ b = -12.584 \\ c = -0.0534 \end{cases}$$

c. If lead vehicle is stationary ($\check{v}_{i-1} = 0$)

$$\begin{cases} a = 9.073 \\ b = -24.225 \\ c = -0.0534 \end{cases}$$

It should be noted that actual implementation of this algorithm requires slight modification to avoid any discontinuities in transition between the above scenarios.

3. NHTSA – A driver-tuned FCW method designed by Brunson et al. [20]. In this method, the components of judgment are range and closing speed. Based on the assumption that lead vehicle's acceleration is constant, the algorithm predicts the host and lead vehicles' range, speed, and acceleration, and then uses this information to compute a projected miss distance. Next, the projected miss-distance is compared against a miss-distance threshold to decide whether a collision warning should be issued. The algorithm can generate three level of cautionary warning that are dependent on the braking capability of the driver. The imminent level corresponds to the maximum braking level while the earlier levels are set for reduced braking levels. This warning sensitivity is a driver-tuned parameter. In the calculation of the projected miss-distance, the NHTSA algorithm considers three different cases: stationary lead vehicle, decelerating lead vehicle come to a complete stop before the host vehicle, and lastly the host vehicle stops while the lead is still moving. The miss-distance varies for each of these cases. To determine the case, one first needs to find the time it takes each host and lead vehicle to come to a complete stop. Similar to previous equations, we index the host (following vehicle) as i and the lead vehicle as $i-1$.

$$T_{i-1} = -\frac{\check{v}_{i-1}}{\check{a}_{i-1}} \tag{18}$$

$$T_i = \begin{cases} -\dfrac{v_i}{a_i}, & if\ \ v_i + a_i T_R < 0 \\ T_R - \dfrac{v_i + a_i T_R}{a_{i,max}}, & if\ v_i + a_i T_R > 0 \end{cases} \tag{19}$$

where $T_R = 1.6s$ represents the combined driver reaction and system delay times that it takes for the reaction (braking) occur after a warning is generated. $a_{i,max}$ is the assumed host vehicle braking

capability. This is a driver-tuned parameter that can be set for 3 different warning levels early (0.32g), intermediate (0.4g), and imminent (0.55g). Consequently, for the case when $T_{i-1} < T_R$:

$$D_{miss} = R + \frac{T_R^2}{2}(a_i - a_{i,max}) - \frac{\breve{a}_{i-1}T_{i-1}^2}{2} - T_R T_i(a_i - a_{i,max}) + \dot{R}T_i + \breve{a}_{i-1}T_{i-1}T_i - \frac{a_{i,max}T_i^2}{2} \quad (20)$$

where R indicates the range (distance) between host and leading vehicles and $\dot{R}$ is the range rate, i.e. the, rate at which the distance between two vehicles changes.

For the case of a stationary lead vehicle or when the host vehicle stops before a moving lead, the miss distance happens when the range rate is zero.

$$D_{miss} = R + \dot{R}T_M + \frac{T_M^2}{2}(\breve{a}_{i-1} - a_{i,max}) - T_M T_R(a_i - a_{i,max}) + \frac{T_R^2}{2}(a_i - a_{i,max}) \quad (21)$$

with

$$T_M = T_R + \frac{\dot{R} + T_R(\breve{a}_{i-1} - a_i)}{a_{i,max} - \breve{a}_{i-1}} \quad (22)$$

which represents the time when miss-distance occurs. In the implementation of the NHTSA algorithm, one should take precautions to avoid division by zero.

Interested readers can find a detailed explanation of each of these algorithms in their respective publications.

The performance of each of these FCW algorithms for a scenario where lead vehicle suddenly brakes to full stop is demonstrated below. In this scenario, the two vehicles drive at a speed of 45 mph with the driver of the following vehicle being distracted (Appendix A). The packet

loss in this communication medium equals 30% (more on this in section 3.3). Having an Error-Dependent communication policy improves the separation distance approximation to the ground truth value, which is a key factor in cooperative vehicle safety systems. Additionally, the response of the driver of the following vehicle to a collision warning is in the form of hard braking.

Figure 15 illustrates the warning range and the effectiveness of CAMP Logistic Regression in preventing a collision in a near crash scenario. Although we prefer time-to-collision as a safety metric for the evaluation of FCW algorithms (see section 5.1), the algorithms above use the concept of a safe gap distance as a mean for warning generation. Hence, to illustrate the changes in this warning range alongside the separation distance of the vehicles, we look at the distance plot between the two vehicles. As was mentioned earlier, the CAMP algorithm generates a warning when the warning range r_w exceeds the estimated distance between the two vehicles. The plot on the left in Figure 15 marks the start of the deceleration of the lead vehicle (dotted vertical green line) which is followed by issuance of a warning 1.26 seconds later (dotted vertical red line). The driver of the following vehicle reacts to this warning after an initial delay of 1.6 seconds which is most visible in the plot of throttle changes (right side plot in Figure 15). Consequently, the following vehicle comes to a complete stop with a final safe gap of 8.52 meters.

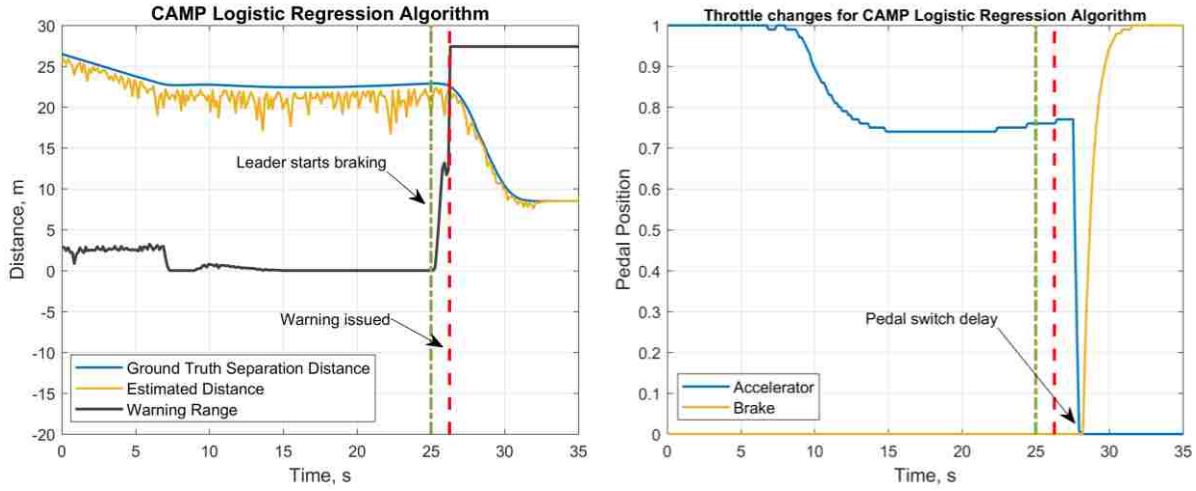


Figure 15: Performance of CAMP Logistic Regression FCW algorithm in a near-crash scenario involving distracted driving

Following figures show the performance of the driver-tuned NHTSA FCW algorithm for three different sensitivity settings, i.e. e,arly (-0.32g), intermediate (-0.40g), and imminent (-0.55g). For the early setting (Figure 16), the warning is generated 1.24 seconds after the start of braking of the lead vehicle. The scenario is concluded with a safe gap of 6.30 meters between the two vehicles. The missing part of the projected miss-distance plot indicates that no value exists for that moment's kinematic parameters.

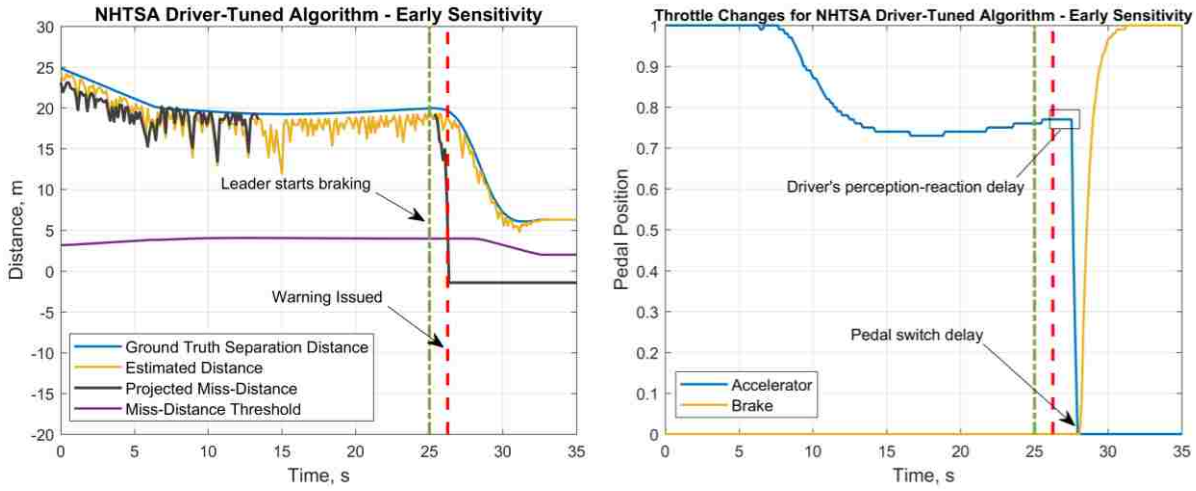


Figure 16: Performance of Driver-tuned NHTSA FCW algorithm with Early sensitivity in a near-crash scenario involving distracted driving

With the intermediate settings, the warning is generated after 1.25 seconds of braking. The following vehicle comes to a complete stop with a safe gap of 2.75 meters.

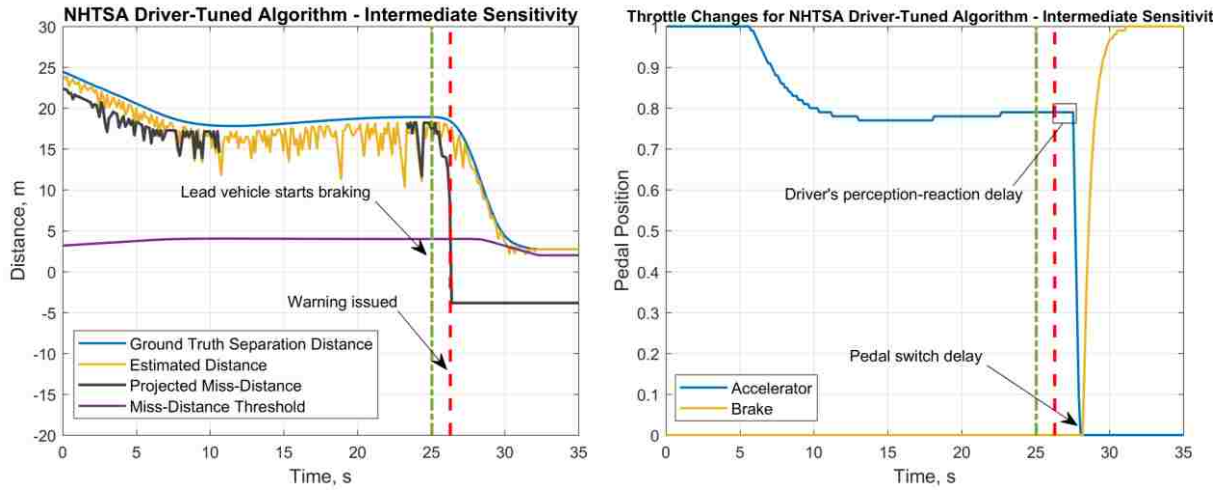


Figure 17: Performance of Driver-tuned NHTSA FCW algorithm with Intermediate sensitivity in a near-crash scenario involving distracted driving

Unlike the above scenarios, for this particular driver of the following vehicle, the NHTSA imminent generates a late warning (after 2.17 seconds) which fails to prevent a collision. Even with a hard-braking response, the vehicles do not have enough space cushion between them and hence, the scenario results in an accident. This illustrates the significance of tuning an FCW algorithm for that driver's specific driving condition.

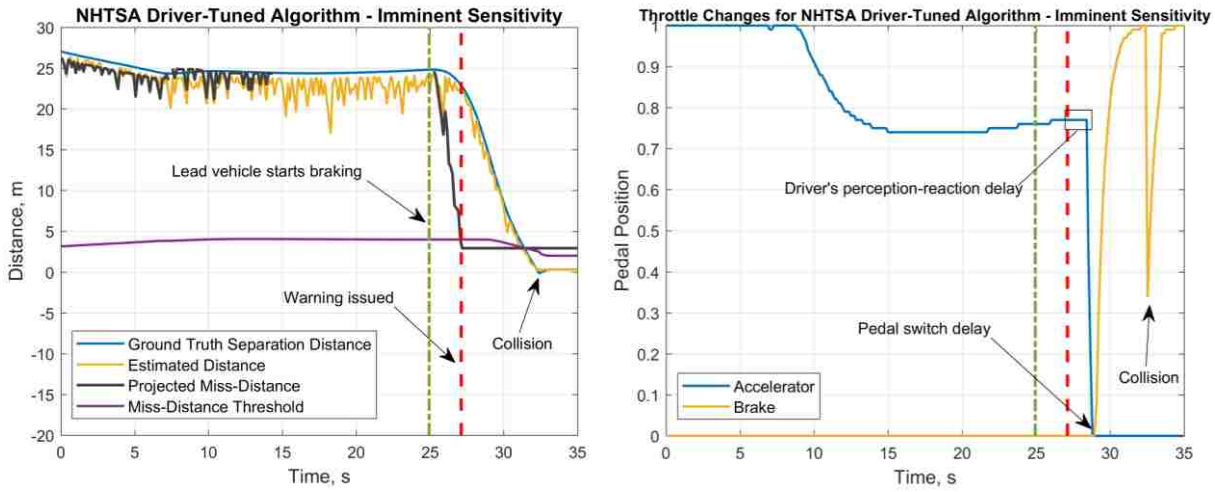


Figure 18: Performance of Driver-tuned NHTSA FCW algorithm with Imminent sensitivity in a near-crash scenario involving distracted driving with a small time-headway

To summarize, we described three different FCW algorithms that are available for public use. The performance of two of these algorithms was shown for a scenario of a decelerating lead vehicle. We saw the importance of driver behavior consideration in design and tuning of FCW algorithms as the NHTSA with imminent sensitivity could not help in avoiding a collision. In the above discussion, we skipped the Knipling et al. FCW algorithm as this does not show a satisfactory result in the early prototype (see section 5.1) of the simulator and therefore was not implemented in the final iteration of the simulator that is built with UE4.

3.2.3.1 Collision Warning Performance Metric

As we have learned, the performance of safety systems relies on the accuracy of sensory information that is supplied from different sources. We have also learned that in a cooperative vehicular safety system, these sensory data are provided through V2V based DSRC network. In this section, we provide a reasonable metric for measuring the effect of different system parameters on the performance of FCW systems. While there is no universally agreed upon metric for this purpose, the work of [39] provides a practical metric for quantifying the performance of FCW. In their paper, authors use a confusion matrix to calculate different metrics such as accuracy, precision (P), true positive (TP), false positive (FP), true negative (TN), false negative (FN), as well as geometric mean of TP and P. The formulation of the confusion matrix as well as the expressions to calculate the above metrics are presented in the Table 1 and 2.

Table 1: Confusion matrix

		Ground Truth	
Number of predictions		Negative (safe)	Positive (threatening)
Prediction	Negative (safe)	a	c
	Positive (threatening)	b	d

Table 2: Metrics calculated using the confusion matrix

Metric	**Expression**
Accuracy	$\frac{a+d}{a+b+c+d}$
Precision (P)	$\frac{d}{b+d}$
True Positive (TP)	$\frac{d}{c+d}$
False Negative (FN)	$\frac{c}{c+d}$
True Negative (TN)	$\frac{a}{a+b}$
False Positive (FP)	$\frac{b}{a+b}$
Geometric Mean	$\sqrt{TP \cdot P}$

In Table 1, the variable a and d refer to the number of correctly predicted safe and threatening (hazardous) instants, while b and c indicate the number of incorrectly predicted safe and hazardous instants (misidentified actual safe scenarios). *Accuracy* can provide us with the overall measure of correct decisions in issuing (or not issuing) a warning. However, accuracy is considered a good performance index, if the population of safe and dangerous data sets are similar. This, however, may not be always as significant portion of most driving datasets is composed of safe driving scenarios and only a small interval of some of the scenarios may indicate a dangerous situation. This is the case even for those scenario that involve an accident, where the initial part of the driving pattern before entering the hazard zone would be considered safe by any warning system

which, in most cases, would dominate the hazardous interval. Therefore, a suggested performance index that better quantifies such unbalanced situations is the geometrical mean of TP and P which is also adopted by the authors of [39].

3.3 Effect of Communication Network on Performance of Vehicular Safety Systems

In a communication network model, the application layer perceives the effect of lower layers on communicated data as a pattern of random losses and delays imposed on the stream of packets received by the host vehicle. Since V2V based DSRC network uses a single hop broadcast network, any unreliability in communication would be originated from MAC and PHY layers. When network transmits information with maximum power, any packet error and message losses within 200 meters of transmission source would be mostly due to MAC layer issues, as fading is not strong enough in this short range [41]. Fading becomes a key factor in network modeling when messages are sent to higher distances especially if the line of sight is blocked. In a single hop DSRC network, transmission delay is insignificant as EDCA (Enhanced Distributed Channel Access) protocol ensures that each packet is sent out in less than contention window size, even in a fully utilized network and despite the high risk of packet collision. Therefore, knowing that the transmission delay in a single hop DSRC network is insignificant and that the losses in a CSMA/CA broadcast network are well-randomized [29], we can describe the effect of the communication network in an abstract mathematical form using a random loss rate during transmission [18]. Such a simplified representation of communication network will help us better investigate and discern the effect of the communication network on the performance of active safety applications.

As a first step in analyzing the mutual effect of coupling of communication and active safety applications, we implemented the baseline design of CCW in which Periodic Beaconing (PB) policy with a transmission rate of 10Hz is used. We can then investigate how the amount of delivered information affects the application performance by varying the transmission rate from 1Hz to 10Hz and Packet Error Rate (PER) from 0 to 0.9. Figure 19-22 report the values of accuracy, precision, true positive and geometric mean (section 3.2.3.1) for all given combinations of sampling rate and PER for several independent simulated crash and near-crash scenarios involving two vehicles (Appendix A). Following performance metrics are considered for two different cases in all the simulated scenarios:

- Left side graphs: periodic beaconing policy without an error-dependent model
- Right side graphs: periodic beaconing policy with constant acceleration model as the error-dependent model

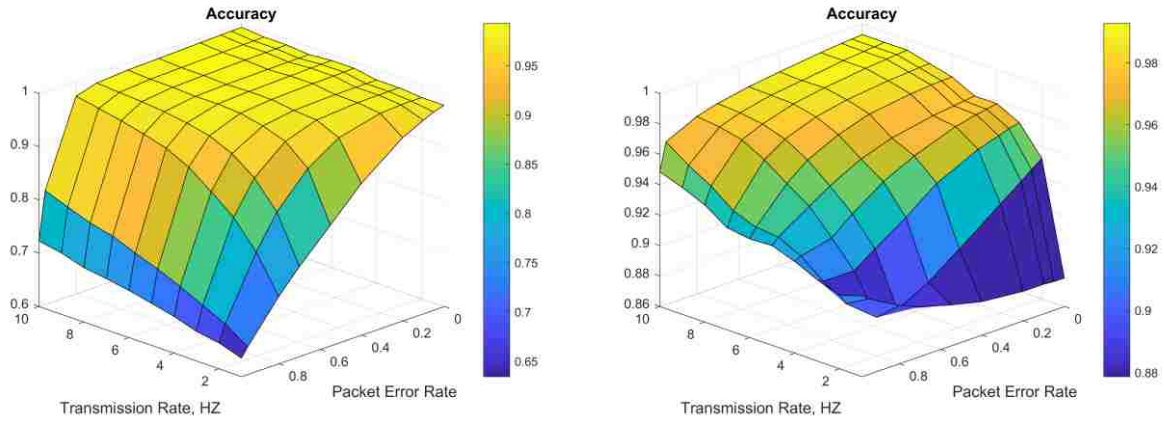


Figure 19: Accuracy of FCW system for different values of PER and transmission rates with PB policy; (left) no error-dependent model; (right) with constant acceleration model

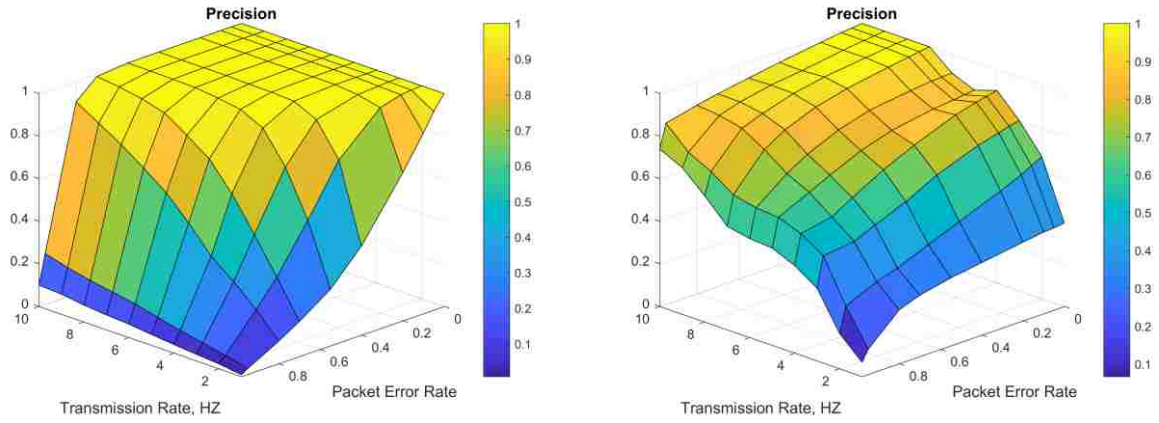


Figure 20: Precision of FCW system for different values of PER and transmission rates with PB policy; (left) no error-dependent model; (right) with constant acceleration model

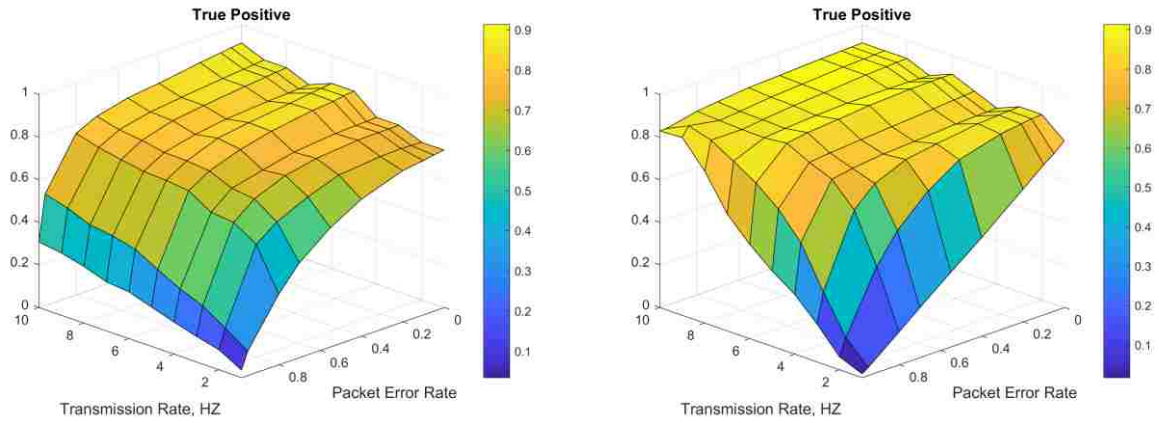


Figure 21: True Positive of FCW system for different values of PER and transmission rates with PB policy; (left) no error-dependent model; (right) with constant acceleration model

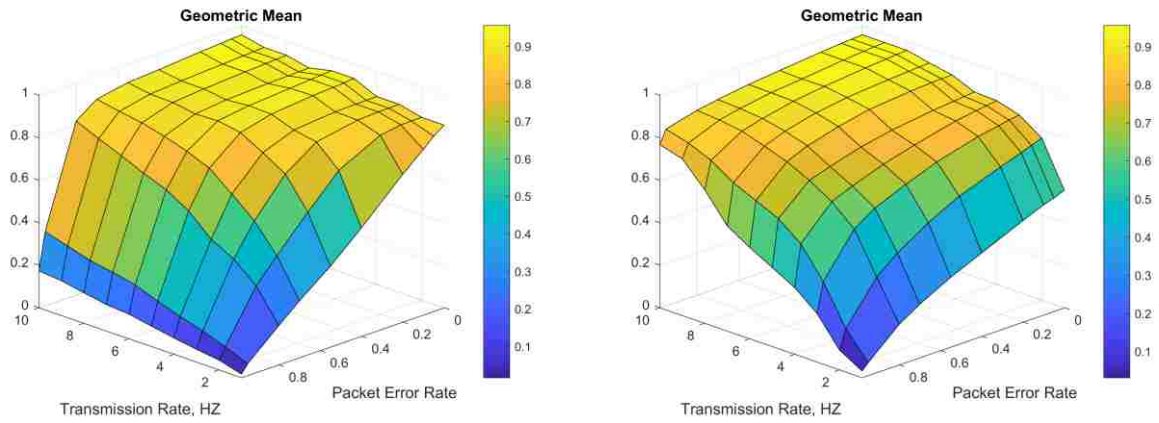


Figure 22: Geometric Mean of FCW system for different values of PER and transmission rates with PB policy; (left) no error-dependent model; (right) with constant acceleration model

We start the discussion of above figures by first looking at the right-side plots where an error-dependent policy is present. From the Accuracy figures, we can observe that a sampling rate above 5 Hz with a PER below 55% result in accuracy above 96%. The peak value of 99% accuracy corresponds to the transmission rate of 10Hz with no PER. The 10Hz is the rate at which FCW system operates and is set up to provide the ground truth. Since simulated data involved predominantly safe sample, even for the worst-case scenario of 1Hz transmission rate and over 90% PER, the accuracy graph enjoys a high value of over 88% in the right-side graph with constant-acceleration error-dependent model and hence cannot be a suitable performance indication. On the other hand, the other three figures, i.e., precision, true positive and geometric mean, help us better analyze these unbalanced situations. In general, higher precision is an indication of fewer false alarms and higher true-positive shows better capability of detecting hazardous situations. Unlike the accuracy plot, the sampling rate of 5Hz with PER of 55% only correspond to 77% precision and

a geometric mean of 79%. We can also more clearly see that as the sampling rate decreases and PER increases, all these plots fell quickly to less than 3%.

The scenarios with no error-dependent model perform nearly as good as the scenarios with an error-dependent model (constant acceleration for this case) only for rates above 5Hz and PER less than ~50%. The advantage of having an error-dependent model is evident at the high PER values where the precision and other metrics in all of the left side plots (no error-dependent model) fell instantly to significantly lower values than in comparison to the same PERs but with an error-dependent model. This indicates the important role of an error-dependent model in improving the performance of CVS. The other missing metrics that were discussed in section 3.2.3.1 (true negative, false positive, etc.) showed a similar trend to precision, and true positive plots and hence were skipped.

Next, we narrow our study of the effect of this coupling of safety and communication components and specifically look at a potential crash scenario that can be avoided with the help of a FCW system. As previous, we look at the effect of network-specific parameters such as transmission rate and the network congestion on the performance of this CCW system. Similar to all previous examples, the scenario discussed here also involves two vehicles that are in a dangerous collision situation (Appendix A). Without an FCW system, the follower vehicle would collide with the leading vehicle. In the case where vehicles are equipped with FCW, a collision warning would be issued for the following vehicle, which triggers a reaction from its driver whose driving state would be changed to the emergency braking state after a short reaction delay (Figure 8). As was discussed in previous chapters, the primary form of a human reaction to a collision warning is braking.

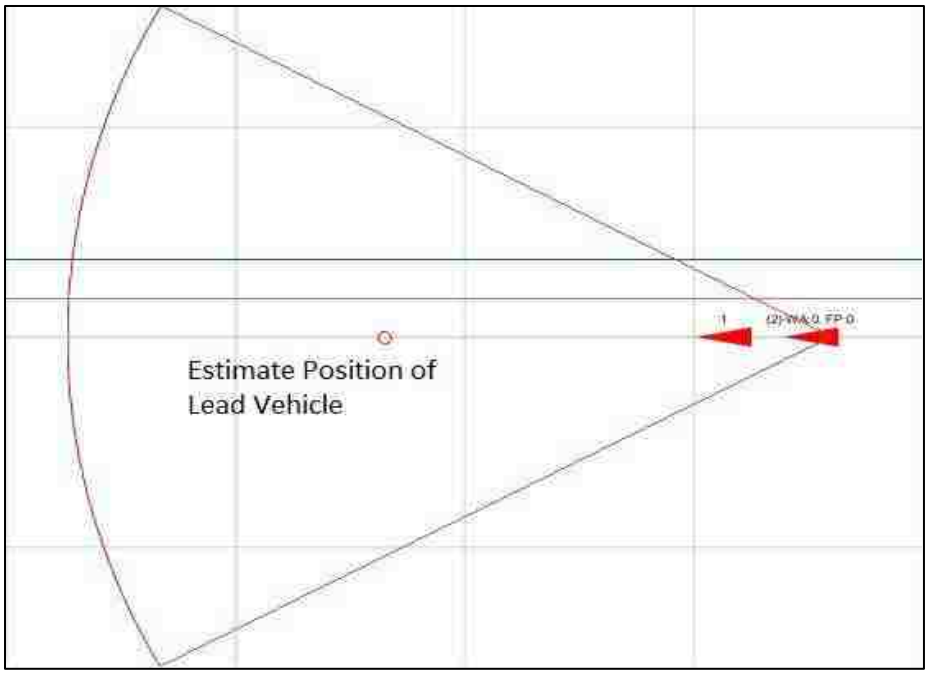


Figure 23: A simplified view of the scenario involving two vehicles with a high packet error rate and inaccurate location estimation

The effect of network congestion can be simulated by setting a high value for the network's PER. We set the PER of the communication network to a value of 70%. Figure 24 shows the response of a distracted driver to a collision warning who is approaching quickly to the vehicle in front. Earlier in section 3.2.3, we saw the response of the following driver and the effectiveness of FCW algorithms in preventing rear-end collisions. However, in this scenario, the high packet loss has raised the problem of position tracking error accumulation, which leads to the failure of safety algorithms to issue an on-time warning. We can observe how the accumulated tracking error leads to ~1.7sec delay in issuing a warning and consequently causing an accident.

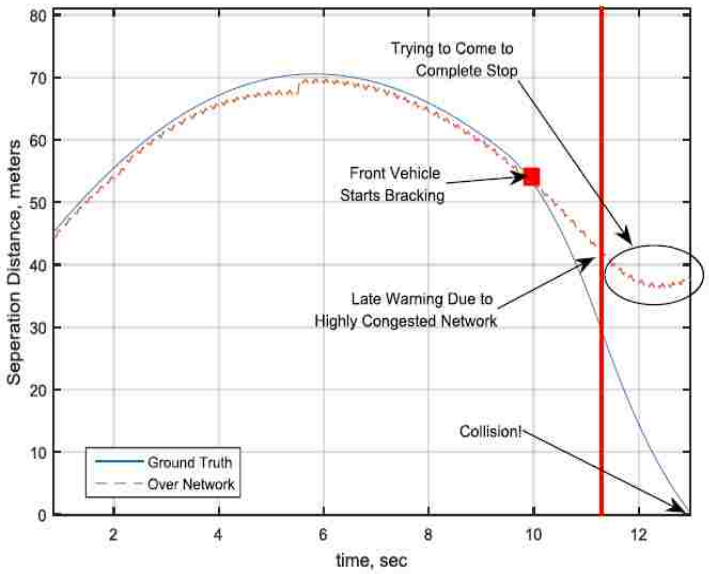


Figure 24: Crash scenario due to network congestion and late warning issuance

The negative effect of position tracking error accumulation is an important issue which needs further studying and analysis. There are ongoing studies on the design of communication logic, as well as the use of different communication medium. The method used in this study is an Error-Dependent policy. There are improved models of this policy such as Error-Dependent Network-Aware policy [42, 43] that uses a simplified network simulator that inflicts losses experienced by the receiver. Interested readers are encouraged to refer to studies [18, 42, 43] that further investigate this issue.

CHAPTER 4: ANALYSIS AND CLASSIFICATION OF HUMAN DRIVING CHARACTERISTICS

Human factors play a significant role in any cyber-physical system that includes humans as part of its physical plant, such as drivers and pedestrians in an Intelligent Transportation System (ITS). In the first section of this chapter, we will form an understanding of human driving characteristics and risky behaviors that indicate a high risk of collision. Using the knowledge of the first section, we will try to analysis a driving dataset and classify different classes of drivers that will help us in extracting driving parameters suitable for the driver model proposed in chapter 2.

4.1 Understanding Human Driving Characteristics

Motor vehicle crashes are showing a steady increase in several types of human choice crash situations [44]. One of such studies by NHTSA shows a total of 37,461 lives lost in 2016 alone, of which 3,450 cases involved distracted driving [44]. Another summary of driving statistics shows that as many as 660,000 drivers use cell phones while driving during daylight hours, exposing themselves to be caught in an accident [45]. These statistics also state that approximately 80% of recorded crashes were the results of drivers' distraction, making it the primary cause of rear-end crashes. However, driver's inattention does not necessarily imply the carelessness of that driver [36]. A driver may avert his/her attention from the forward path for both driving and non-driving reasons. Examples of driving-related distractions may include watching for a pedestrian, looking at signs and landmarks, looking at side-mirrors, or turning the head during a lane change.

Despite the importance and the need for its detailed study, distracted driving does not necessarily describe a specific driver's driving personalities. Instead, it defines a state in which the driver is no longer paying attention to the driving task. In other words, one cannot bind distracted driving to their driving characteristics (as this does not relate to how the kinematics of vehicle are affected). On the hand, there exists a category of dangerous on-the-road driving behavior that is labeled "aggressive driving." NHTSA defines aggressive driving as "the operation of a motor vehicle in a manner that endangers or is likely to endanger persons or property" [46]. An aggressive driver may exhibit behaviors such as speeding, failure to obey traffic signals, making frequent lane changes, and most importantly tailgating which is considered one of the significant cause of crashes that may result in severe injury or death. Recent NHTSA reports [44] show that speeding, which is a characteristic of an aggressive driver, has been the cause of approximately 10,111 deaths, accounting for more than 27% of all traffic fatalities in 2016 alone.

These alarming statistics urge the implementation of active vehicular safety systems, which once again brings us to imply the goal of this work: introducing a simulation platform that would allow the evaluation and verification of vehicular active safety systems in different traffic scenarios. The simulation architecture described in this work allows the introduction of two types of drivers, i.e., cautious and distracted, with three different categories of driving behaviors, i.e., normal, conservative, and aggressive. This segregation results in a total of six different states that a driver may be in at any time.

We use the visual perception of a driver as a deciding factor of the cautious and distracted states. We define a cautious driver as one who pays particular attention to the road ahead and what is in front of him/her, whereas representing a distracted driver as one, whose vision range is severely

restricted, resulting in him/her being unaware of the surroundings. In such a system, a distracted driver has a very high chance of colliding with a decelerating Leading Vehicle (LV) in a congested traffic scenario or a traffic light.

The driving characteristics of a driver can be extracted from a congested traffic scenario, which is considered a major contributing factor to aggressive driving [45]. The frustration raised from on-the-road delays caused by either a high traffic volume during rush hour or a collision can yield in impatient drivers responding aggressively by either following too closely or changing lane frequently. For this purpose, we use the Next Generation Simulation (NGSIM) datasets [47] to find appropriate parameters for IDM car-following model that will represent the discussed categories of drivers. The NGSIM dataset includes the vehicle trajectory of more than 2000 drivers during an afternoon rush hour. This makes the dataset an ideal candidate for the task in mind. The details of the NGSIM dataset along with its statistical analysis are discussed in the next chapter.

4.2 Dataset Analysis: Parameter Extraction and Driver Classification

The Next Generation SIMulation (NGSIM) program was originated by Federal Highway Administration of the U.S. Department of Transportation (USDOT) to "improve the quality and performance of simulation tools, promote the use of simulation for research and applications, and achieve wider acceptance of validated simulation results" [47]. The NGSIM dataset provides high resolution (10Hz) positional and kinematic data of 5648 vehicles, which were collected during afternoon rush hour in three 15-minute intervals (4:00pm to 4:15pm, 5:00pm to 5:15pm, and 5:15pm to 5:30pm). These vehicle trajectory data were collected on a sub-second basis in US

Interstate 80 in California on April 2005, using seven stationary cameras mounted on a 30-story building. The dataset is regarded to contain the most useful features for the development and validation of microscopic traffic models. The NGSIM dataset was published in 2007 and is freely available for download [47].

Before starting our analysis, it is worth mentioning that the NGSIM dataset is provided in imperial units which were converted to the metric system for the results reported here. Additionally, the trajectory and kinematic data of vehicles exhibited noise artifacts that were filtered using a moving average filter with half a second window. The comparison of a filtered and unfiltered sample acceleration data for three different drivers are shown in Figure 25.

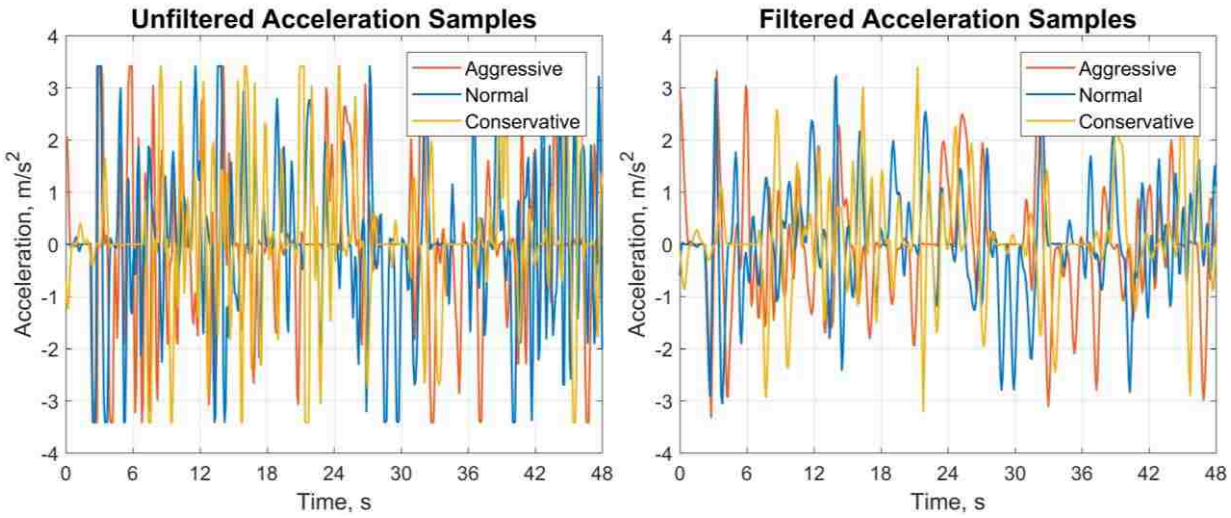


Figure 25: Comparison of filtered and unfiltered acceleration data

Given the specification of an aggressive driver, i.e., speeding and tailgating, two main kinematic features that can best describe these aggressive characteristics are acceleration and the time headways of vehicles. Time headway is a measurement of the time it would take the following vehicle to reach the leading vehicle if the leader remains stationary

$$\mathrm{T} = \frac{s_i}{v_i} = \frac{x_{i-1} - x_i - l_{(i-1)}}{v_i}, \quad \forall v_i > 0 \tag{23}$$

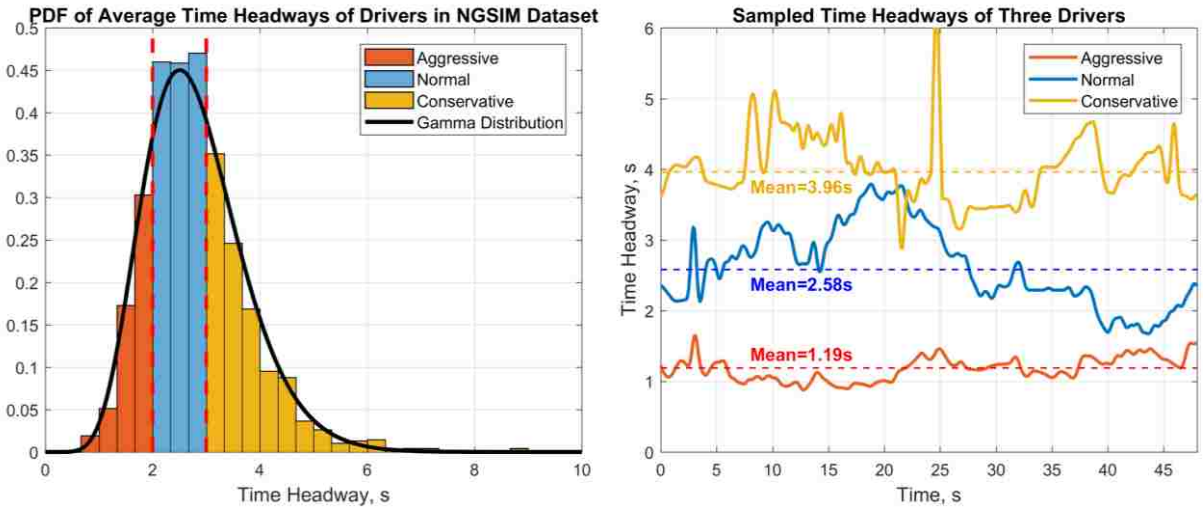
It has been shown that drivers tend to keep a constant time headway from their leading vehicle during a car-following driving regime [48]. The probability distribution function of this feature for all drivers within the dataset is represented in Figure 26a.

Driving courses including NHTSA give several rules of thumbs and recommendations for a safe time headway. For a scenario such as the one in the NGSIM dataset (dry and clean pavement), and for the average speed of 35mph (~55km/h), the recommended following distance rule is 2-3 seconds. This can also be observed from Figure 26a with the mode of the histogram sitting around 2.5 s. Obviously, many drivers follow this rule in an attempt to keep a safe driving experience. However, we can see that a fraction of drivers have kept a much smaller time headway to their leading vehicles while the other group of drivers have chosen to keep larger time headways. Using the recommended time headway rule, we classify these drivers into 3 clusters: aggressive drivers with ($\mathrm{T} < 2.0s$), normal drivers ($2.0s \leq \mathrm{T} \leq 3.0s$), and conservative drivers ($3.0s < \mathrm{T}$). The changes in time headway of three randomly sampled drivers from all bins, along with the mean of their time headways are represented in Figure 26b. We can see that these drivers attempt to keep a relatively constant time headway all the time.

Having clustered these drivers into three categories, next, we try to extract the IDM parameters through the statistical analysis of the time headway and acceleration data. The PDF of the average time headways presents sufficient information to find the distribution ratio of the drivers and their minimum desired time headway T (Eq. 3). We use a gamma probability distribution function of the following form to fit this data:

$$f_{\Gamma}(x|\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \tag{24}$$

Thus, representing the time headway T of each driver as a gamma-distributed random variable x with shape parameter $\alpha = 9.15$ and rate parameter $\beta = 0.31$. Based on the above classification, and the fitted gamma PDF, the ratio of each category of the driver can be obtained using the inverse percentile at the edge points 2 and 3 seconds. These ratios are presented in Table 3. A simulated driver will be assigned a category based on bins' thresholds and the randomly gamma-distributed time headway.



(a) (b)

Figure 26: (a) Distribution and clustering of mean time-headways of all drivers in the NGSIM dataset; (b) Changes in time headway of three sampled drivers

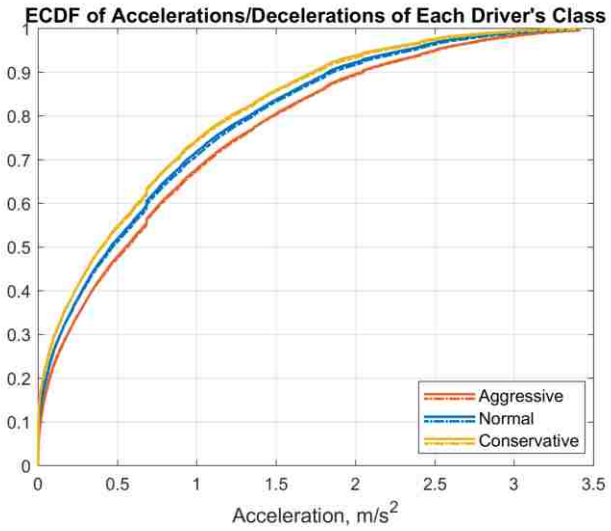


Figure 27: distribution of acceleration values of all drivers of each category

Next, we analyze the changes in the acceleration profiles to find the remaining parameters of IDM. Figure 27 represents the Empirical Cumulative Distribution Function (ECDF) of all the acceleration values of each driver's category. We can see that the aggressive driver has kept a higher acceleration profile than the other two drivers. As we move toward the conservative drivers, the resistance in accelerating/decelerating becomes more apparent. From this observation, the comfortable accelerations α and braking decelerations b (Eq. (2)) were represented as a uniformly distributed random variable within the range of 70^{th} to 90^{th} percentile of accelerations and decelerations respectively. The final extracted ranges for each category of drivers are shown in Table 3.

Table 3: Parameters of Intelligent Driver Model for three categories of drivers

Driver Type	Aggressive	Normal	Conservative
Comfortable Acceleration, m/s^2	$\alpha \in [1.53, 2.75]$	$\alpha \in [1.43, 2.59]$	$\alpha \in [1.30, 2.41]$
Comfortable Deceleration, m/s^2	$b \in [1.52, 2.73]$	$b \in [1.43, 2.59]$	$b \in [1.27, 2.41]$
Desired Time headway, s	$f_\Gamma(x\|\alpha, \beta) < 2$	$2 \leq f_\Gamma(x\|\alpha, \beta) \leq 3$	$3 < f_\Gamma(x\|\alpha, \beta)$
Ratio	19%	43%	38%

CHAPTER 5: PERFORMANCE EVALUATION AND SIMULATION RESULTS

This chapter is concerned with the evaluation of the simulation framework proposed in earlier chapters. In the first subsection, we look at the performance of the previously mentioned forward collision warning algorithms in a large traffic scenario. This analysis is conducted using our early prototype 2D simulator which simulates the movement of the vehicle using a pure car-following model. In section 5.1 we take an in-depth look at the car-following model used by our early prototype that was taken from the popular SUMO simulator and show its limitation and unrealistic response in simulating vehicles' movements. We demonstrate that a simple CFM alone is not sufficient to evaluate active vehicular safety systems. In the final subsection (5.3) we repeat our analysis of a large-scale traffic scenario but in the comprehensive 3D environment with full vehicle dynamics consideration.

5.1 Large-Scale FCW Performance Evaluation within the 2D Co-Simulator

In this section, we study the performance of different FCW algorithms in a large-scale traffic scenario. We created a dense traffic setting, involving 200 cars. We then analyzed the performance of each algorithm and the overall traffic system by studying the results obtained from simulation equivalent to one hour of driving. To evaluate the performance of FCW algorithms in a large-scale traffic scenario, we created a dense traffic setting, involving 200 cars. We then analyzed the performance of each algorithm and the overall traffic system by studying the results obtained from simulation equivalent to one hour of driving. This section is followed by a discussion on the

limitation imposed on simulating and analyzing active vehicular safety systems when vehicle dynamics is significantly simplified in the form of a pure car-following model. We specifically look at the famous Krauss car-following model that is the main driving component of the SUMO [8]. The Krauss model, named after its creator, was designed in 1998 and is the main driving component of SUMO. The original model is formulated as follows:

$$v_{safe}(t) = v_i(t) + \frac{g(t) - g_{des}(t)}{\tau_b + \tau} \tag{25}$$

$$v_{des}(t) = \min\left[v_{max}, v_i(t) + a_i(t)\Delta t, v_{safe}(t)\right] \tag{26}$$

$$v(t + \Delta t) = \max[0, v_{des}(t) - \eta] \tag{27}$$

$$x(t + \Delta t) = x_i(t) + v_i \Delta t \tag{28}$$

where $x_i(t), v_i(t), a_i(t)$ represent the position, velocity and acceleration of the Host Vehicle (HV) at time t; $v_{max}(t)$ is the maximum possible velocity of HV; $g(t)$ is the gap distance between vehicles at time t; $g_{des}(t)$ is the model parameter that represents the desired gap between vehicles; $\tau -$ driver reaction time; $\tau_b = \frac{v_{i-1} - v_i}{b}$ is determined by desired deceleration b that HV driver would use; $\eta > 0$ is a random perturbation to deviate from optimal driving; and Δt represents the simulation time step. We implemented the updated version of the Krauss model which can be found within SUMO's source code.

The main simulation parameters are shown in Table 4. Since the main concern of every collision avoidance algorithm is the perception of the braking time and the level of braking needed, we addressed this by studying the effect of different braking levels (0.85g, 0.675g, and 0.5g) as a hard-braking reaction to an imminent warning.

Table 4: Simulation Parameters

Simulation Time, hour	1.0
Number of Vehicles	200
Car-Following Model	Krauss
Reaction Type	Hard Braking $(0.85g, 0.675g, 0.5g)$
Reaction Time, s	1.6
Packet Error Rate (PER)	0.3
Transmission Rate, Hz	10
FCW Algorithms	CAMP Logistic Regression Knipling NHTSA (Early, Immediate, Imminent)

The recommended safety metric used in our evaluation is Time-To-Collision (TTC) which is frequently used in the literature. A driver reacts and adjusts speed in regards to what is seen during driving. The study by Horst [49] suggests that a decision made by a driver to brake is based on the perception of TTC. Hence, TTC can be regarded as a descriptor of how a driver recognizes stimuli and how dangerous a situation has become. TTC can be calculated by the formula in [50]:

$$TTC_i = \frac{x_i - x_{i-1}}{v_i - v_{i-1}}, \quad \forall v_i > v_{i-1} \tag{29}$$

where indices i and $i-1$ represent host and the leading vehicles respectively.

As a first step, we started a simulation run with no warning algorithm enabled. After multiple runs, a factor of 3% distracted drivers resulted in an average of 245 crashes for 1 hour of simulation. Figure 28 represents the Empirical Cumulative Distribution Function (ECDF) of TTCs for the whole duration of the simulation. We can see that TTCs contributed only to crash data frequently have lower values than the overall driving data.

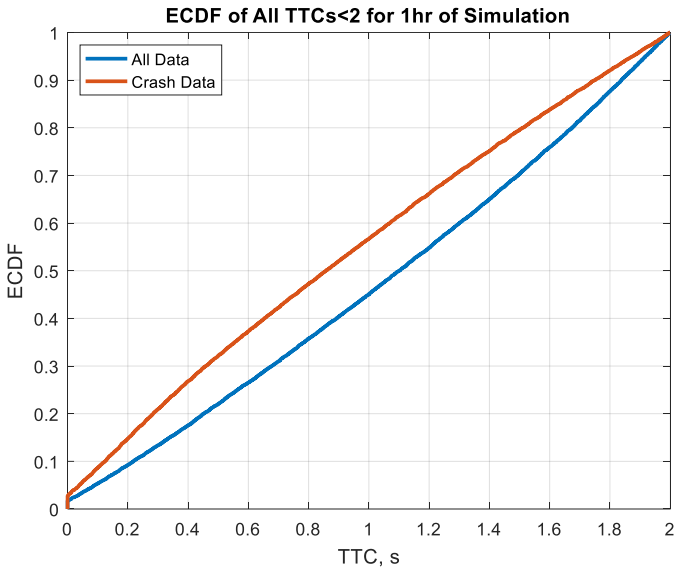


Figure 28: ECDF of all recorded TTCs versus TTCs of crash data for 1 hour of simulation with no warning system

Next, the same scenario was repeated several times for each individual warning algorithm and a corresponding driver braking level. Table 5 shows the recorded number of collisions for each algorithm. Although some collisions were inevitable, it can be observed that all the algorithms were effective in reducing the total number of rear-end crashes significantly. Amongst the three tested algorithms, NHTSA (imminent level warning) resulted in the lowest number of accidents, albeit it generated warnings more frequently compared to the other two algorithms. As we expected, Knipling resulted in more collisions than the other methods since this algorithm considers only two scenarios: Leading Vehicle (LV) stationary, and LV decelerating.

Table 5: Total Number of Crashes for 1 Hour of Simulation

Braking Level After Receiving a Warning		$\mathbf{0.5g}$	$\mathbf{0.675g}$	$\mathbf{0.85g}$
Warning Algorithm	CAMP Logistic Regression	27	15	2
	Knipling	39	37	25
	NHTSA Early ($0.32g$)	**11**	5	10
	NHTSA Intermediate ($0.40g$)	12	6	4
	NHTSA Imminent ($0.55g$)	64	**4**	**2**
	No Warning Algorithm	**245**		

Much to our surprise, it turned out that except for NHTSA early warnings, a braking reaction with a maximum deceleration of 0.85g resulted in the lowest number of crashes. This observation implies that if all vehicles were equipped with FCW systems, even a hard-braking reaction with a maximum deceleration rate in the middle of a roadway would still be effective in lowering the number of rear-end collisions.

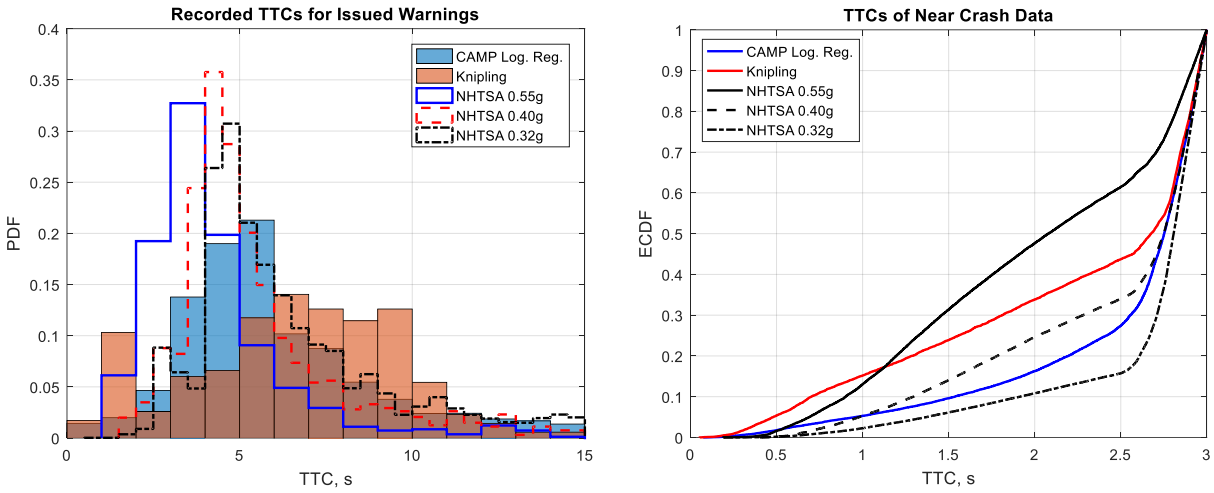
Table 6: Total Number of Near-Crash Scenarios

Braking Level After Receiving a Warning		$\mathbf{0.5g}$	$\mathbf{0.675g}$	$\mathbf{0.85g}$
Warning Algorithm	CAMP Logistic Regression	1243	1564	1311
	Knipling	**244**	**216**	**266**
	NHTSA Early ($0.32g$)	8170	8887	9280
	NHTSA Intermediate ($0.40g$)	1988	3000	3067
	NHTSA Imminent ($0.55g$)	1856	749	791

When a vehicle displays a sudden reaction to a warning by hard braking, it forces its following vehicles to slow down or potentially hard brake too, causing more hazards and near-crash

scenarios. Since all vehicles in our simulation are equipped with FCW systems, such sudden reaction does not necessarily result in more collisions, rather more warnings are generated. We consider a warning that successfully averted a crash scenario into a near-crash as a positive warning. The total number of near-crash scenarios in all the test configurations are reported in Table 6. We observe that NHTSA early and intermediate had the highest number of warning generation (not including the false warnings) which complies with the assertion made in [37]. Knipling, on the other hand, had the lowest number of warning generation and hence, the highest number of collisions as was shown in Table 6. In the long run, NHTSA imminent warning level proved once again to be the most effective of the three, maintaining a good balance between collision reduction and the frequency of warning generation.

The overall sensitivity of each FCW algorithm is represented as the Probability Distribution Function (PDF) of TTCs when a warning was issued (Figure 29a). Given the limited volume of the thesis and that different braking levels showed a similar tendency, we only represent the TTCs for highest braking level, i.e., 0.85g. As can be seen in the figure, PDF of the NHTSA imminent (0.55g) is stretched more towards lower values than other PDFs with a peak TTC value corresponding to 3.5 s. This observation satisfies the nature of the algorithm, in which the warning generation is tuned for aggressive driving. Accordingly, this trend is followed by NHTSA intermediate (4.25 s), NHTSA early (4.75 s), and Camp Logistic Regression (5.5 s). Note that all TTCs shown in this work are calculated via communicated data that arrive at host vehicle through BSM messages. These are also the perceived time by CCW algorithms.



(a) (b)

Figure 29: (a) Distribution of TTCs that were recorded when a positive warning was generated (b) ECDF of near-crash data for all recorded TTCs

In the final analysis (Figure 29b), the TTCs of all vehicles in near-crash plots were examined. We see that unlike other FCW methods, NHTSA imminent (0.55g) algorithm had a higher percentage of time that vehicles were exposed to shorter TTCs. This implies that in the long run, even after exhibiting a hard-braking reaction due to a warning, almost 50th percentile of the vehicles were getting as close as 2 s to their respective leading vehicle. Nevertheless, the closest TTC that many vehicles experienced was 1 s for 15% of the time and 0.5 s in some rare cases. Early response to other algorithms is evident as ECDF drops significantly within a period of 2.5 – 3.0 s.

5.2 Limitation of a Pure Car-Following-Based Mobility Model in Evaluation of Active Vehicular Safety Systems

In the previous section, we showed the performance of FCW systems using the approach of traditional simulators in simulating the vehicles' movements, i.e., using a CFM as a simplified

model of vehicles' dynamical, as well as driver-specific, characteristics. We specifically looked at the CFM of the popular open-source traffic simulator SUMO, i.e., the Krauss model. However, this model (and essentially any CFM) poses a significant limitation in analyzing the performance of active vehicular safety systems on a nanoscopic scale. Ideally, we wanted this section to appear in chapter 2.1 as part of our reasoning for selection of a more comprehensive simulation platform. However, this topic had to be discussed here after the reader has obtained an understanding of the simulation environment as well as the cooperative vehicular safety systems, especially the forward collision warning algorithms that are used in this work. If we take a closer look at the FCW algorithms discussed in section 3.2.3, we can see that they make extensive use of kinematical parameters of both the following and leading vehicles as the only input they can obtain from BSM sent through DSRC network or read from other sensors. These parameters play a significant role in the judgment made by these algorithms and hence, they must be simulated with the highest accuracy possible.

Unfortunately, car-following models such as Krauss make significant simplifications to the vehicles' dynamics, to a level that a detailed looked at their performance shows an unnatural response in velocity and acceleration of vehicles. This unnatural response is especially visible in a scenario where the lead vehicle is stationary which appears regularly in the urban areas where vehicles have to make frequent stops such as stopping at a traffic signal. Figure 30 shows the changes in velocity and acceleration of the following vehicle for a scenario where following vehicle approaches and stops behind a stationary leader with a safe gap (Appendix A). Figure 31 demonstrates the changes in the separation distance between these two vehicles as well as the changes in warning range of CAMP Logistic Regression FCW algorithm.

The violation in physical laws is obvious in the above figure, especially in acceleration plot. Unless acted by a large impulse, no physical object can instantly change its acceleration by such a large value. It is also worth mentioning that from the simulation and driving perspective, this scenario is safe and collision-free as the following vehicle stops with a large and safe gap distance behind the leading vehicle. However, the FCW algorithm generates a false warning at 5.9s. This problem is explained by the way Krauss model is originally formulated. For the first part of this scenario (before the abrupt change), this CFM calculates the safe speed using only the kinematic formula of constant acceleration $v_i(t) + a_i(t)\Delta t$ from equation (26), which assumes the driver drives with the maximum acceleration of the vehicle, i.e. $a_i(t) = a_{i,max}$. This is also a reason for many unnecessary false warnings in our earlier evaluations.

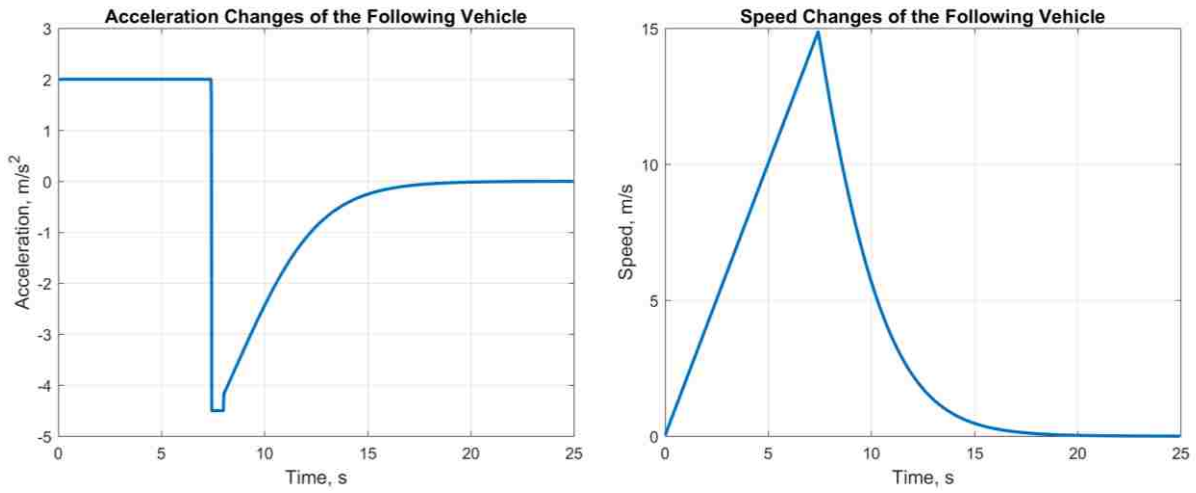


Figure 30: Changes in velocity and acceleration of the following vehicle obtained from Krauss car-following model for a safe driving scenario with a stopped leader 100m away

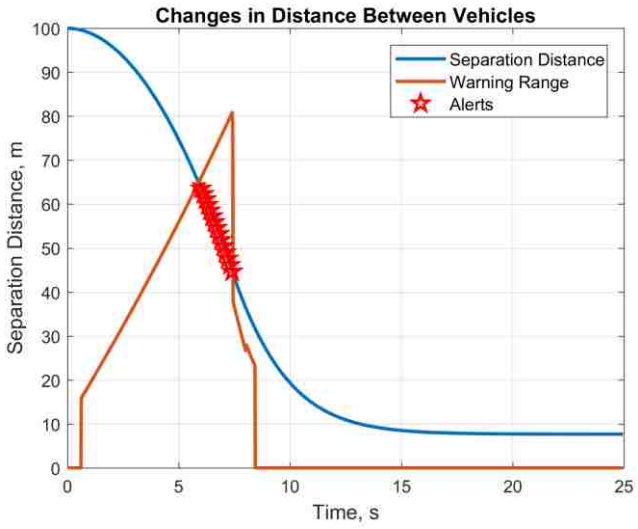


Figure 31: Changes in the separation distance between two vehicles in a safe driving regime with a stopped leader 100m away simulated using only Krauss car-following model

The unrealistic kinematical response of CFMs mostly taints those models that are based on calculating the next speed value. Models such as IDM which try to calculate the next acceleration value typically do not show such abrupt changes in the kinematic response of the vehicle. Since CFMs are representative of the way a human-driver drives and follows a leading vehicle with a safe distance, albeit in a simplified manner, we can look at the speed/acceleration values generated by CFMs as a reference value that the driver would like to reach in order to keep a safe distance from the leading vehicle. Hence, we transferred our development environment into the Unreal Engine 4 which, thanks to its physics engine alongside NVIDIA PhysX Vehicle, allowed us to separate the vehicle's dynamics from its driver model and represent the driver as a feedback controller who attempts to maintain a safe speed and distance from the leading vehicle.

5.3 Large-Scale FCW Performance Evaluation with Near-Realistic Physics

In this section, we re-analyze the FCW algorithms in large-scale traffic scenario within the Unreal Engine-based 3D co-simulator with a near-realistic physical engine and vehicles' dynamics. This evaluation is based on the parameters obtained from chapter 4, for which we designed a dense highway traffic scenario, similar to NGSIM, involving 150 gamma-distributed drivers. The highway scenario was built to be 2km long with vehicles looping back to the start once they have reached the end. This way we ensured a consistent congested scenario.

We integrated the six classes of drivers introduced earlier and analyzed their driving response in two different situations, one where no vehicle is equipped with any CCW system, and the other where all vehicles have CCW. To study the effectiveness of implemented FCW algorithms in preventing near-crash and crash scenarios, we specified the behavior of the emergency state (Figure 8) to be a hard braking reaction by the driver when a warning is issued. This emergency state reaction is simulated by pressing the brake pedal all the way down (i.e., the throttle of -1) after an initial perception-reaction delay of 1.3s. The maximum braking torque of all vehicles is 1500Nm. The work of Malaterre et al. [3] shows that braking was the primary form of the reaction of many drivers to an FCW alert. In fact, it is the primary objective of any collision avoidance algorithm to determine the right time for braking that would avoid a dangerous scenario. The main parameters of the simulation are presented below.

Table 7: Simulation Parameters

Duration, minutes	90
Number of Vehicles	150
Exact Number of Drivers Aggressive/Normal/Conservative	27/66/57
Network's Packet Error Rate	30%
Implemented FCW Algorithms	CAMP Logistic Regression; NHTSA Driver-Tuned Models
Error-dependent Estimator	Constant Acceleration

Similar to the previous section, we used Time-To-Collision (TTC) in addition to Time Headway (T) to evaluate the performance of the two FCW algorithms (CAMP Logistic Regression and NHTSA Driver-Tuned Models). We start our first simulation analysis with no warning algorithm enabled. After multiple runs, a factor of 3% distracted drivers resulted in an average of 92 individual vehicles being involved in collisions, 74% of which were resulted from distraction, 15% from aggressive driving characteristics and the other 11% were caught in a pileup. Of these 92 counted collisions, 19.5% involved aggressive drivers, 52.2% normal drivers, and 28.3% conservative drivers. It is worth mentioning that the vehicles that were involved in an accident continued blocking the road for a period of 5 to 10s before being removed from and respawned back into the simulator at a later time.

For a comparison with NGSIM data, the count density distribution of time headways of different classes of drivers are shown below (Figure 32). The following results include both cautious and distracted drivers for each behavior characteristic. We can observe that the combined

distribution with a mode of ~2.5s exhibits similar characteristic as the analysis shown in section 4.2.

Next, the same scenario was repeated several times but with the vehicles equipped with FCW systems and the drivers able to react to a warning by hard braking. We ran a series of analysis with a communication network's PER of 30%. Table 8 shows the collision statistics obtained for each class of driver and the respective FCW algorithm.

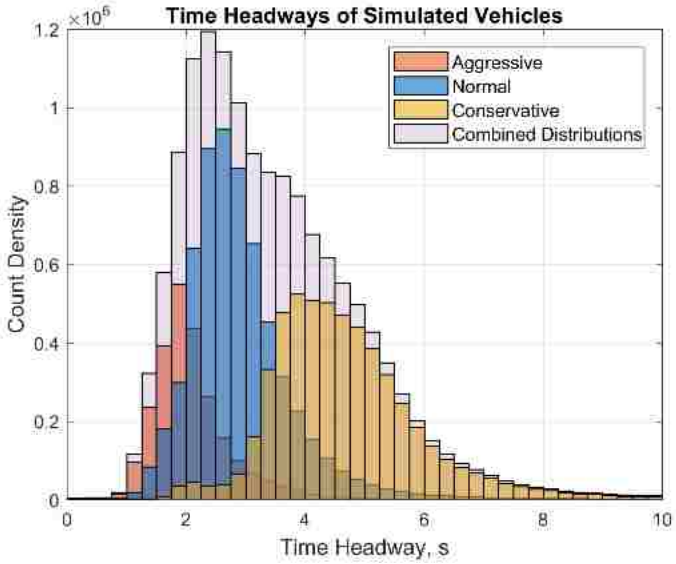


Figure 32: Count density distribution of time headways of simulated vehicles for each class of drivers

While some collisions were inevitable, all the FCW algorithms were effective in significantly reducing the number of collisions. Upon the occurrence of a collision, our algorithms analyzed the scenario to determine who was at fault. The "Total" column for each driver's class in Table 8 refers to the total number of crashes that type of driver was at fault. We can see that for all the analyzed scenarios, the conservative drivers were involved in considerably fewer accidents,

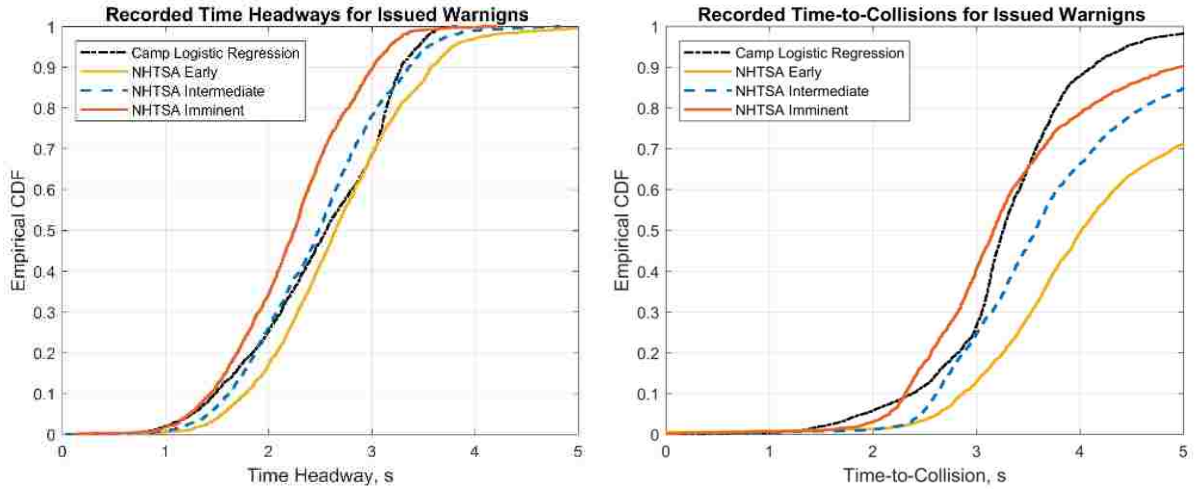
whereas the aggressive drivers, despite their smaller numbers (18%), were most at fault. Interestingly, for all accidents where a normal or a conservative driver was at fault, it was due to them being distracted. Hence, implying that not all the tested FCW algorithms were effective in preventing the distracted-related collisions. Recall that all drivers had a 1.3s perception-reaction delay. This implies that the implemented algorithms may still fail at complex circumstances, e.g., the vehicle traveling at higher speeds with its driver being distracted while the leader experiences an emergency state and suddenly brakes. Amongst the two FCW algorithms, CAMP Logistic Regression performed worse than all three settings of NHTSA as this algorithm is not a fully-adaptive algorithm with less flexibility compared to the driver-tuned NHTSA ones.

When a driver displays a sudden reaction to a warning by hard braking, it forces its following vehicles to slow down or potentially hard brake too, causing more hazards and near-crash scenarios. For the simulation configuration above, we observed that an accident might result from either a distracted driver or a sudden slowdown of a leading vehicle, or combination of both. Therefore, we consider a warning that successfully averted such a crash scenario into a near-crash as a positive warning. Table 8 shows the statistics with regards to the number of warnings generated. For an aggressive driver, we can see that the NHTSA Early and Intermediate had the highest number of warning generation, albeit most of these were false warnings. This is also true for the other two algorithms and implies that for the aggressive driving behavior, none of these algorithms are suitable options. Such a significant fraction of false warnings will result in the frustration of the driver and ultimately in him/her shutting down the system. A similar trend follows for normal drivers whereas the NHTSA with Early alert settings appeared to assist the conservative drivers exceptionally good with 87% positive warnings and zero accidents at fault.

Table 8: Crash and FCW warning statistics for a simulated 90 minutes long scenario

Driver Type (Number of drivers)		**Aggressive (27)**			**Normal (66)**			**Conservative (57)**		
Collision Statistics	Collision Information	Total	Distracted	Leader Hard Braking	Total	Distracted	Leader Hard Braking	Total	Distracted	Leader Hard Braking
	No Warning Algorithm	14	7	0	**21**	**21**	**0**	7	6	0
	CAMP Logistic Regression	6	2	4	**8**	**8**	**1**	5	5	0
	NHTSA Early ($0.32g$)	**6**	**6**	**0**	5	5	0	0	0	0
	NHTSA Intermediate ($0.4g$)	**6**	**3**	**2**	5	5	0	3	3	0
	NHTSA Imminent ($0.55g$)	**11**	**6**	**7**	7	7	2	1	1	0
Warning Statistics	Warning Information	Total	Positive	Ratio	Total	Positive	Ratio	Total	Positive	Ratio
	CAMP Logistic Regression	843	156	18.5%	**1116**	**411**	**36.8%**	309	38	12.3%
	NHTSA Early ($0.32g$)	1105	117	10.5%	1063	361	34.0%	**193**	**168**	**87.0%**
	NHTSA Intermediate ($0.4g$)	1290	70	5.4%	958	402	42.0%	**208**	**143**	**68.7%**
	NHTSA Imminent ($0.55g$)	989	122	12.3%	**442**	**208**	**47.1%**	149	57	38.3%

The overall sensitivity of each FCW algorithm is represented as the PDF of TTCs when a warning was issued (Figure 33a). While the time headway distributions show a similar characteristic and are close to each other, we can see a noticeable difference in the distribution of time-to-collision. Figure 33b explains the poor performance and higher number of accidents for CAMP Logistic Regression algorithm as the 90th percentile of the warnings being generated when host vehicles were less than 4s away from colliding with their leading vehicles. We can conclude from this observation that the algorithm was tuned for very aggressive driving characteristics. However, given the perception-reaction delay of a driver, the timings of its warnings were considerably late even for the aggressive drivers to effectively avoid the collision. Accordingly, this trend is followed by NHTSA Imminent, Intermediate and with Early having a 90th percentile of 8s (not shown in the figure).



(a) (b)

Figure 33: (a) Sensitivity of each forward collision warning algorithm represented through empirical distribution functions of time headway; (b) Sensitivity of each forward collision warning algorithm represented through empirical distribution functions of time-to-collision

CHAPTER 6: FUTURE WORK

The derivation of realistic behavioral models, as well as the development and verification of the adaptive safety mechanisms, is a challenging problem, which requires rich and extensive data for specific scenarios. The datasets that are currently available cannot be efficiently used for such tasks, as they are often limited to either a few scenarios or a few drivers, and lack many important features. These datasets are also often cluttered, and cannot be understood well enough to work with. For example, the 100 Car Naturalistic Dataset is essentially a collection of unrelated scenarios, each representing recorded data of a single car, missing important factors such as leading and following vehicle information (This dataset provides distance to five closets objects, however, it is unclear which one of these objects is the leader or following vehicle). It is also especially difficult to model behaviors such as driver distractions with such datasets as it is unclear when and whether the driver was distracted or not. Furthermore, the values for throttle and brake pedals positions provided are of no use as it is not clear where the baseline is. The NGSIM dataset is an example of a dataset that contains surrounding vehicles information but is limited to kinematic and positional data of each vehicle, missing the driver and vehicle specific data such as throttle and brake pedal positions, steering angle, driver glaze. To address these challenges, we suggest a number of novel methods for simulating hazardous scenarios and designing automated collision prevention mechanisms.

6.1 Data Collection and Driver Modeling using Virtual Reality

Driving data collection phase can be very costly and time-consuming if approached through real-world measurement campaigns. In addition, some of the most critical test scenarios can be very dangerous and risky as they may expose real drivers to hazardous situations. For instance, to design an adaptive FCW algorithm, a thorough analysis of driver behavior during a collision scenario is required. A collision scene, which could be potentially very dangerous, should be designed and executed in several trials. This difficulty in data collection can be extended to other scenarios such as behaviors representing states of distraction, emergency braking behaviors or aggressive driving. Real-world data acquisition of such scenarios, especially the hazardous ones, is not attainable for many researchers as they are very costly and time-consuming. Alternatively, these data can be collected in a simulated environment. This option is now more feasible than ever before due to the exponential growth in the computational power, and the availability of game development technologies such as Unreal Engine 4, which comes at free with NVIDIA PhysX vehicle dynamic model, is cable of delivering an epic and realistic experience in real-time. Moreover, the advent of virtual reality technology, as well as hand tracking devices such as Oculus touch controllers and leap motion and their compliance with the Unreal Engine, has now allowed software developers to immerse their users into a virtual world, giving them a feeling of physical presence. The feeling of immersion that virtual reality provide is astonishing, to the extent that even those who do not play or enjoy video games have been amazed by the potentials of this technology. Figure 34 shows the virtual reality platform that we have set up in Networked Systems Lab at the University of Central Florida. Although most technical aspects of it have already been completed, the platform still

requires comprehensive digital assets including a complete 3D environment with proper lighting and sound effect for it to provide a fully immersive and near-realistic experience.

Figure 34: (Left) Virtual reality platform set up in Networked Systems Lab at the University of Central Florida. (Right) The view of the co-simulator environment through virtual reality lens

In a virtual world, there can hardly be a limitation to scenario generation. As an example, distraction can be introduced to a driver by placing a distractive object such as a peculiar billboard, building, or people in the virtual world within the driver's vicinity and record all data related to the vehicle and its driver, i.e. throttle and brake position, vehicle's kinematic and positioning data, head movements, etc. Using virtual reality, not only such collected data can resemble a realistic behavior of the driver, but also serve our need in modeling, as they are scenario specific, noise-free, and contain all information of the environment and the surrounding objects (i.e., vehicles).

6.2 Development of a Vehicle Safety System Through Reinforcement Learning

In this subsection, we suggest another novel method for simulating hazardous scenarios and designing automated collision prevention mechanisms. In particular, we recommend a policy

gradient-based reinforcement learning approach to allow the automated vehicle to train itself and learn the optimal way of minimizing the damage of an accident. Reinforcement learning is a promising approach that lets an agent learn by interaction without requiring any knowledge of how to complete that task. These approaches have achieved extraordinary results over the past few years by solving complex problems that had exceedingly large solution spaces and were once unsolvable. The vehicle agent can learn the optimal behavior in a dangerous road traffic scenario by interacting with a virtual environment that is made within the traffic simulation software presented in this work. This approach has the potential of filling the gap in the crash and near-crash data collection scenarios by the inclusion of human perception and reaction factors. The data that is collected can then be used as a base for designing other advanced driver assistant systems.

CHAPTER 7: CONCLUSION

With the rapid growth of advanced driver assistant and automated systems, the simulation tools have become an integral part of the development process of these systems. Not only the simulators can boost the development efficiency, but they also offer opportunities in analyzing scenarios that otherwise would have been extremely costly, time-consuming, and on some occasions dangerous to the human test subjects. In this work, we extended the co-simulation framework proposed in [14] and deployed it on the state-of-the-art game engine technology Unreal Engine 4 (UE4). With the help of NVIDIA PhysX and the UE4 physics engine, we were able to represent a large-scale traffic scenario with detailed 3D physics and visualization. We described the individual components of our platform in detail and proposed a simple human interpretable extension to the traditional car-following models. The extended model related a human driver to a feedback controller and employed the Intelligent Driver Model to supply a reference acceleration while incorporating a Fuzzy-PD system to compensate for the error. In this extended approach, we integrate the perception-reaction delay time of the driver as an individual component, along with descriptive driving characteristics through the parameters of the Intelligent Driver Model and fuzzy membership functions. Furthermore, we surveyed different driving characteristic and looked into a traffic dataset of a congested scenario to extract these various driving behaviors. We used the time headway and acceleration of the drivers as descriptive features of their driving characteristic. Using this information, we classified the drivers into three main classes, i.e., aggressive, normal, and conservative. We found that a gamma-distribution function can best describe the dispersal of the drivers within the environment. Finally, we incorporated all the extracted parameters into our

simulator and equipped the vehicles with forward collision warning systems. We then analyzed the performance of these systems for each class of driver with a hard-braking response as driver reaction to a collision warning. Of the tested safety algorithms, driver-tuned NHTSA with Early warning configuration performed remarkably good for conservative drivers with 87% positive warnings generation and prevented all rear-end collisions. Conversely, none of the four configurations of forward collision warning systems showed adequate performance for the aggressive driving characteristics.

APPENDIX A: A SCENARIO OF TWO VEHICLES ON A STRAIGHT ROADWAY

In this first appendix, we would like to briefly explain a typical driving situation that we frequently use throughout our evaluations and explanations of different topics. This scenario consists of two vehicles on a one-lane straight roadway with one vehicle following the other. We refer to the vehicle in back as the following (host) vehicle and the vehicle in front as the leader or leading vehicle. Many equations that appear in this work make use of information from the following and leading vehicle. In these equations, the following vehicle is index by i and the leader is identified by $i-1$. Throughout our work, we use many variations of this specific two vehicle situation that best help us convey our message and understanding of the topic. Using our simulation platform that is discussed in chapter 2, we created different scenarios such as slow moving leader, decelerating leader, stopped leader, fast approaching follower, and distracted follower. The simplified illustration of the two vehicle scenario alongside variables of interest is shown in the Figure 35.

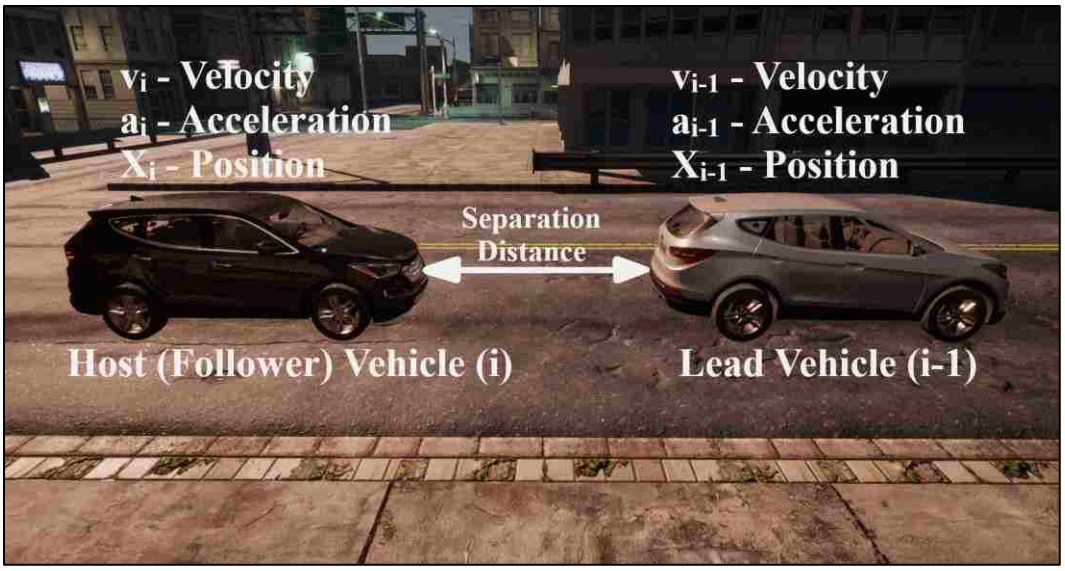


Figure 35: A scenario of two vehicles with variables of interest

APPENDIX B: VEHICLE PARAMETERS

In this appendix, we would like to provide the dynamical parameter of the considered vehicle. Throughout our simulation, we dealt with only one type of vehicle for which we had sufficient information. The vehicle we used throughout our simulations was a 2013-2014 Hyundai Santa Fe Sport. The 3D model of this vehicle was manually edited and rigged in Autodesk 3ds Max which was then imported into Unreal Engine 4 and was set up to be used with UE4 vehicle wrapper and NVIDIA PhysX Vehicle. A similar model of this vehicle can be downloaded for free at [51].

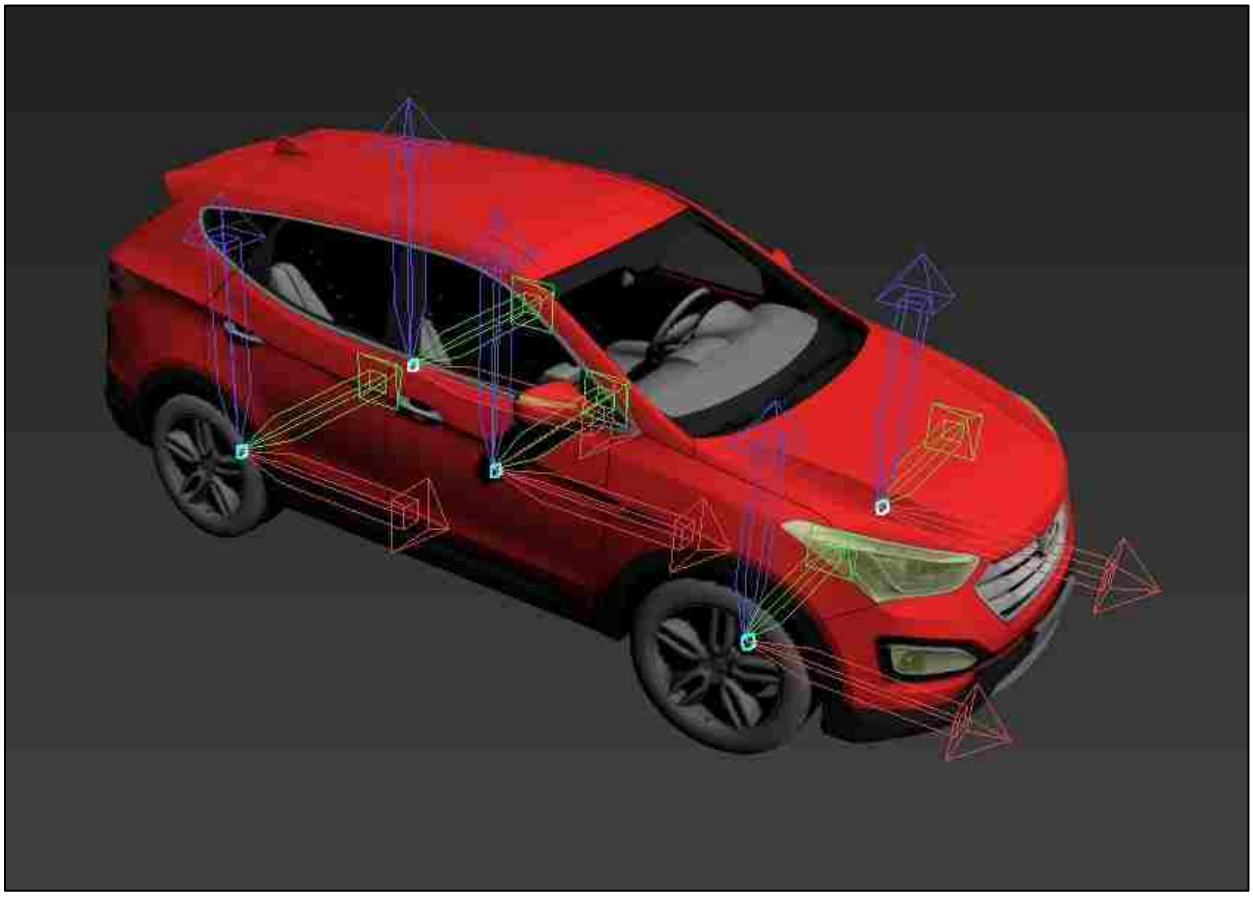


Figure 36: A visual representation of the rigged 3D model of the Hyundai Santa Fe

The characteristic of the torque curve as well as some of the important mechanical and dynamical information of this vehicle that are used within our simulations are provided below.

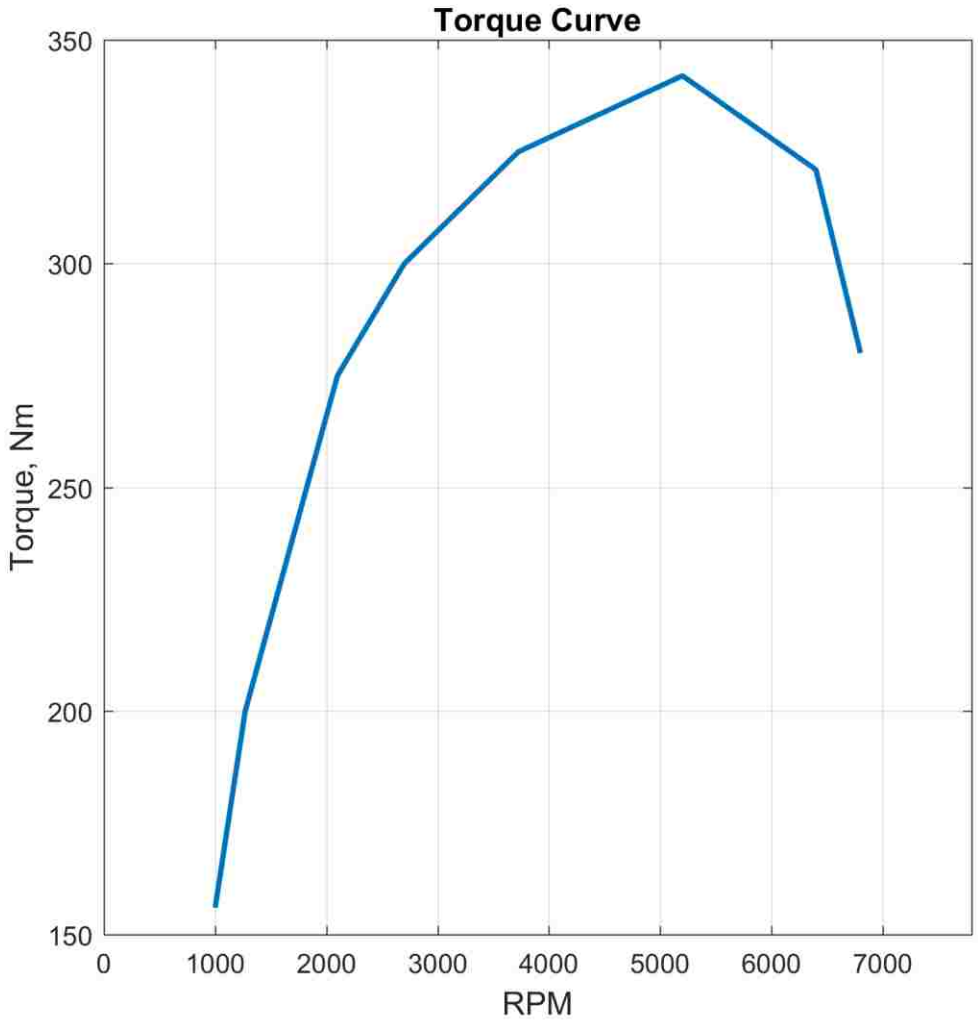


Figure 37: Torque curve for a 2017 Hyundai Santa Fe *(data source: www.automobile-catalog.com)*

Table 9: General mechanical parameters of Hyundai Santa Fe

Mass, kg	2510
Chassis width, cm	188
Chassis height, cm	155
Max engine RPM	6800
Moment of inertia of the engine, kgm^2	0.2
Differential type	Open Front Drive

Table 10: Transmission setup data

Gear Switch Time	0.2
Gear auto box latency	3.0
Final ratio	3.51

Table 11: Gears setup data

Gear Setup	**Gear Ratio**
Gear 1	4.651
Gear 2	2.831
Gear 3	1.842
Gear 4	1.386
Gear 5	1.0
Gear 6	0.772
Reverse Gear Ratio	-3.393

LIST OF REFERENCES

[1] E. K. Sauber-Schatz, D. J. Ederer, A. M. Dellinger and G. T. Baldwin, "Vital signs: motor vehicle injury prevention - United States and 19 comparison countries," MMWR Morb Mortal Wkly Rep 2016;65.

[2] C. Webb, "Motor vehicle traffic crashes as a leading cause of death in the United States," Traffic Safety Facts Crash-Stats. Report No. DOT HS 812 297, Washington, DC: National Highway Traffic Safety Administration, 2012-2017.

[3] G. Malaterre, F. Ferrandez, D. Fleury and D. Lechner, "Decision making in emergency situations," *Ergonomics,* vol. 31, no. 4, pp. 643-655, 1998.

[4] R. Sengupta, S. Rezaei, S. E. Shladover, D. Cody, S. Dickey and H. Krishnan, "Cooperative collision warning systems: concept definition and experimental implementation," *Journal of Intelligent Transporation Systems,* vol. 11, no. 3, pp. 143-155, 2007.

[5] E. Moradi-Pari, H. Nourkhiz Mahjoub, Y. P. Fallah and A. Tahmasbi-Sarvestani, "Utilizing Model-Based Communication and Control for Cooperative Automated Vehicle Applications," *IEEE Transactions on Intelligent Vehicles,* vol. 2, no. 1, pp. 38-51, March 2017.

[6] A. Tahmasbi-Sarvestani, H. Nourkhiz Mahjoub, Y. P. Fallah and E. Moradi-Pari, "Implementation and Evaluation of a Cooperative Vehicle-to-Pedestration Safety Application," *IEEE Intelligent Transportation Systems Magazine,* 2017.

[7] C. Sommer, R. German and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing,* vol. 10, no. 1, pp. 3-15, Jan. 2011.

[8] D. Krajzewicz, J. Erdmann, M. Behrisch and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements,* vol. 5, pp. 128-138, December 2012.

[9] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *Proc. European Simulation Multiconf. (ESM '01)*, 2001.

[10] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom and A. Sumner, *Torcs, the open racing car simulator*, vol. 4, 2000.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, "CARLA: An Open Urban Driving Simulator," *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1-16, 2017.

[12] S. Shah, D. Dey, C. Lovett and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," arXiv:1705.05065, 2017.

[13] "Unreal Engine," Epic Games, 1998-2018. [Online]. Available: https://www.unrealengine.com/.

[14] A. Jamialahmadi and Y. P. Fallah, "Analysis of the Impact of Driver Behavior Models on Performance of Forward Collision Warning Systems," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Orlando, FL, USA, 2017.

[15] "NVIDIA PhysX Vehicle SDK," Nvidia Corporation, 2008-2017. [Online]. Available: http://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Vehicles.html.

[16] "PTV Vissim," PTV Group, [Online]. Available: http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/.

[17] M. Treiber and A. Kesting, "An Open-Source Microscopic Traffic Simulator," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 3, pp. 6-13, 2010.

[18] Y. P. Fallah and M. K. Khandani, "Analysis of the coupling of communication network and safety application in cooperative collision warning systems," in *ICCPS '15 Proc. of the ACM/IEEE Sixth Intl. Conf. on Cyber-Physical Sys.*, Seattle, Washington, 2015.

[19] Y. Q., *A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems,*

Department of Civil and Environmental Engineering, Massachusetts Institute of Technology: 193, 1997.

[20] VIRES Simulationstechnologie GmbH, "OpenDrive," [Online]. Available: http://www.opendrive.org/.

[21] D. Filev, J. Lu, F. Tseng and K. Prakah-Asante, "Real-time driver characterization during car following using stochastic evolving models," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2011.

[22] G. Burnham, J. Seo and G. Bekey, "Identification of human driver models in car following," *Control, IEEE Transactions on Automatic*, vol. 19, no. 6, pp. 911-915, 1974.

[23] P. Zheng and M. McDonald, "Application of Fuzzy Systems in the Car-Following Behaviour Analysis," *Fuzzy Systems and Knowledge Discovery*, vol. 3613, pp. 782-791, 2005.

[24] M. Treiber, A. Hennecke and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805-1824, August 2000.

[25] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 314-325, 2005.

[26] L. Xu, J. Hu, H. Jiang and W. Meng, "Establishing Style-Oriented Driver Models by Imitating Human Driving Behaviors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2522-2530, 2015.

[27] P. L. Olson and M. Sivak, "Perception-response time to unexpected roadway hazards," *Human Factors*, vol. 28, no. 1, pp. 91-96, 1986.

[28] J. B. Kenney, "Dedicated short-range communication (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162-1182, 2011.

[29] Y. P. Fallah, C. L. Huang, R. Sengupta and H. Krishnan, "Analysis of Information Dissemination in Vehicular Ad-Hoc Networks with Application to Cooperative Vehicle Safety Systems," *IEEE Transaction on Vehicular Technology*, vol. 60, no. 1, pp. 233-247, 2011.

[30] "Dedicated Short Range Communications (DSRC) Message Set Dictionary™ J2735_200911," SAE International, 2009.

[31] "Vehicle Safety Communication Projects. Report No. DOT HS 810 591," U.S. Department of Transportation National Highway Traffic Safety Administration, Washington, DC, 2006.

[32] M. Kalantari-Khandani, W. Mikhael, Y. P. Fallah and K. Goseva-Popstojanova, "Data-Based Analysis of Sampling and Estimation Methods for Vehicle Tracking Over Wireless Networks," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Orlando, FL, USA, 2017.

[33] W. G. Najm, M. S. Mironer, J. Koziol, J. Wang and R. R. Knipling, "Synthesis report: examination of target vehicular crashes and potential ITS countermeasures. DOT-HS-808-263," National Highway Traffic Safety Administration, Washington, DC, 1995.

[34] R. J. Kiefer, M. T. Cassar, C. A. Flannagan, D. J. LeBlanc, M. D. Palmer, R. K. Deering and M. A. Shulman, "Forward collision warning requirements project: refining the CAMP crash alert timing approach by examining "last-second" braking and lane change maneuvers under various kinematic conditions (Report No. DOT HS-809-574)," National Highway Traffic Safety Administration, Washington, DC, 2003.

[35] R. Kiefer, D. LeBlanc, M. Palmer, J. Salinger, R. Deering and M. Shulman, "Forward Collision Warning Systems: Development and Validation of Functional Definitions and Evaluation Procedures for Collision Warning/Avoidance Systems," Report No. DOT-HS808-964, Washington, DC: National Highway Traffic Safety Administration, 1999.

[36] R. R. Knipling, M. Mironer, D. L. Hendricks, L. Tijerina, J. Everson, J. C. Allen and C. Wilson, "Assessment of IVHS countermeasures for collision avoidance: rear-end crashes (DOT HS-807-995)," National Highway Traffic Safety Administration, Washington, DC, 1993.

[37] S. B. McLaughlin, J. M. Hankey, T. A. Dingus and S. G. Klauer, "Developing of an FCW

algorithm evaluation methodology with evaluation of three alert algorithms," National Highway Traffic Safety Administration, Tech. Rep, Washington, DC, 2009.

[38] G. J. Forkenbrock and B. C. O'Harra, "A Forward Collision Warning (FCW) Performance Evaluation," National Highway Traffic Safety Administration, Washington DC.

[39] K. Lee and H. Peng, "Evaluation of Automotive Forward Collision Warning and Collision Avoidance Algorithms," *Vehicle System Dynamics,* vol. 43, no. 10, 2005.

[40] S. J. Brunson, E. M. Kyle, N. C. Phamdo and G. R. Preziotti, "Alert Algorithm Development Program - NHTSA rear-end collision alert algorithm - Final report," National Highway Traffic Safety Administration, Washington, DC, 2002.

[41] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai and M. P., "Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication Frequency Band," *IEEE Transaction on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 645-658, 2011.

[42] C. Huang, Y. P. Fallah, R. Sengupta and H. Krishnan, "Adaptive InterVehicle Communication Control for Cooperative Safety Systems," *IEEE Network,* vol. 24, no. 1, pp. 6-13, 2010.

[43] Y. P. Fallah, C. L. Huang, R. Sengupta and H. Krishnan, "Design of Cooperative Vehicle Safety Systems based on Tight Coupling of Communication, Computing and Physical Vehicle Dynamics," in *Proc. of ACM/IEEE Int. Conf. on Cyber-Physical Systems ICCPS*, 2010.

[44] National Center for Statistics and Analysis, "2016 fatal motor vehicle crashes: Overview (Traffic Safety Facts Research Note. Report No. DOT HS 812 456)," National Highway Traffic Safety Administration, Washington, DC, 2017.

[45] National Center for Statistics and Analysis, "2011 Distracted Driving (Traffic Safety Facts Research Note. Report No. DOT HS 811 737)," National Highway Traffic Safety Administration, Washington, DC, 2013.

[46] J. Stuster, "Aggressive Driving Enforcement: Evaluation of Two Demonstration Programs,"

National Highway Traffic Safety Administration; Report No. DOT HS 809 707, Washington, DC, 2004.

[47] U.S. Department of Transportation, "NGSIM - Next Generation Simulation," 2007. [Online]. Available: https://data.transportation.gov/d/8ect-6jqj. [Accessed 5 May 2007].

[48] D. Helbing, "Traffic and Related Self-Driven Many-Particle Systems," *arXiv:cond-mat/0012229v2 [cond-mat.stat-mech] for this version,* 2001.

[49] R. Horst, *Time-to-collision as a cue for decision-making in braking,* Vision in Vehicles—III, 1991.

[50] G. Lu, B. Cheng, Q. Lin and Y. Wang, "Quantitative indicator of homeostatic risk perception in car following," *Safety Science,* vol. 50, no. 9, pp. 1898-1905, 2012.

[51] bardiaesm, "3D Model Hyundai Santa Fe Sport 2013 - 2014," 2014. [Online]. Available: www.thingiverse.com/thing:308814.

[52] Y. P. Fallah and M. K. Khandani, "Context and Network Aware Communication for Connected VEhicle Safety Applications," *IEEE Intelligent Transportation Systems Magazine,* 2016.

[53] "Co-Simulator Demonstration," [Online]. Available: https://vimeo.com/252441087.

[54] DSRC (Dedicated Short Range Communication) Tech Cmte, "On-Board System Requirements for V2V Safety Communications J2945/1_201603," SAE International, 2016.

[55] Y. P. Fallah and M. K. Khandani, "Context and Network Aware Communication for Connected Vehicle Safety Applications," *IEEE Intelligent Transportation Systems Magazine,* 2016.