2021

# Monitoring of Microscopic Traffic Behavior for Safety Applications Using Temporal Logic

Mariam Wessam Hassan Mohamed Nour
*University of Central Florida*

# MONITORING OF MICROSCOPIC TRAFFIC BEHAVIOR FOR SAFETY APPLICATIONS USING TEMPORAL LOGIC

by

MARIAM NOUR
B.Sc. German University in Cairo, 2017

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science
in the Department of Civil, Environmental and Construction Engineering
in the College of Engineering & Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2021

Major Professor: Mohamed H. Zaki

# ABSTRACT

Smart cities are revolutionizing the transportation infrastructure by the integration of technology. However, ensuring that various transportation system components are operating as expected and in a safe manner is a great challenge. One of the proposed solutions is traffic monitoring systems which collect and analyze traffic data for the safe operation and management of the overall system. Even though traffic safety analysis has been tied to crash data, surrogate safety measures (SSM) have recently emerged as a replacement. SSM can provide a convenient alternative for understanding the impact of conflicts on overall road safety. Traditionally, conflicts were studied through manual methods that are time-consuming and prone to biases. With the onset of new technologies such as automated video techniques, micro-simulation and Vehicular ad-hoc Networks researchers are adopting statistical and machine learning methods for developing reliable indicators for possible traffic crashes. However, these technologies don't provide high-level reasoning about the overall state of the traffic. In this work, we propose the use of formal methods as a means to specify and reason about the traffic network's complex properties. Formal methods provide a flexible tool to define the safe operation of the traffic network by capturing non-conforming behavior, exploring various possible states of the traffic system and detecting any inconsistencies within it. Hence, we develop specification-based monitoring for the analysis of traffic networks using the formal language, Signal Temporal Logic. We develop monitors that identify safety-related behavior such as conforming to speed limits, maintaining appropriate headway, and performing safe lane-change maneuvers. The framework is tested using a calibrated micro-simulated highway scenario and offline monitoring is applied to individual vehicle trajectories to check whether they violate or satisfy the defined safety specifications. Statistical analysis of the outputs show that our approach can differentiate violating from conforming vehicle trajectories based on the defined specifications. This work can be utilized by traffic management centers to study the traffic stream properties, identify possible hazards, and provide valuable feedback for automating the traffic monitoring systems.

## ACKNOWLEDGMENTS

First, I would like to start by expressing my deepest gratitude and appreciation towards my advisor Dr Mohamed Zaki, for his continuous guidance, advice and support.

I would also like to thank my respectable committee, Dr Samiul Hasan and Dr Shaurya Agarwal , for their valuable feedback and comments.

Finally, I would like to deeply thank my family for being a constant source of encouragement and support, and confidence in my success.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

## 1.1 Background

Smart mobility will be revolutionizing the way modern transportation network is envisioned. They hold the promise of a more sustainable, safe, and efficient future for our commutes. However, such great ambition entails considerable challenges. Ensuring that the various components of the urban road networks, such as traffic systems, are operating optimally and safely is not a trivial task. As a result, a growing research stream on traffic monitoring systems is being performed to design, analyze and implement novel approaches for safety applications. Traffic monitoring systems collect and analyze necessary data for the proper operation, evaluation, and management of the overall transportation infrastructure [1]. The collected data are invaluable in understanding the traffic dynamics, identifying issues, and bringing forth practical solutions. Current approaches to traffic monitoring systems depend on traffic detector sensors, Vehicular Adhoc Networks (VANETs), or video processing for data collection [2]. Traffic detector sensors such as roadside sensors, Bluetooth detection, or inductive loop usually capture basic traffic information such as speed, headway, or counts. VANETs, while not designed as sensors but rather as a way of communication between vehicles, can be used to transmit more detailed information like positions, traffic-related surrounding information like safety warnings. Video data also can capture many of the above information in addition to detailed trajectories information. With advances in machine learning, a detailed interpretation of the traffic scenes is now also possible. However, none of the above technologies can provide high-level reasoning about the state of the traffic. Commonly, this task is achieved through manual visualization of the captured traffic data.

Surrogate safety measures (SSM) have emerged as an indispensable tool for predicting and evaluating traffic safety. SSM are derived from non-crash events that provide a larger data-set for analyzing safety [3]. SSM use vehicle trajectories and interaction between vehicles to understand the implication of conflicts on overall road safety. Traditionally, these conflicts were studied through manual methods, which were time-consuming and prone to errors. But with the onset of new technologies, researchers are exploiting automated video techniques and micro-simulation for developing reliable indicators for possible traffic crashes. However, with the introduction of embedded technologies within the traffic environment in the form of connected vehicles and infrastructure with various levels of autonomy, the task of evaluating and improving safety and mobility is becoming more complex [4].

Formal verification is initially developed to offer a rigorous approach for analyzing computer systems, exploring the possible states that the system can visit, and detecting any inconsistencies in its operation. Inconsistencies can be in the form of incorrect implementation of the system requirements, contradicting rules that govern the system operations or even bugs in implementing the system. Formal verification requires precise specification language to describe the system requirements (expected behavior) and a model to describe the system's different states. Variants of formal methods exist, as discussed in the literature review section of this thesis. Many of those approaches have been applied successfully to complex systems such as cyber-physical systems [5] and recently to transportation systems [6], [7], [8]. The use of formal methods in transportation applications can provide a means to specify and reason about the traffic network's complex properties in a rigorous manner. This new way of traffic verification can uncover potential shortcomings in the network operators that might not be detected using the current state-of-the-art methods.

The primary goal of this thesis is to adopt specification-based monitoring for analyzing vehicle trajectories. Our approach focuses on identifying safety-related behavior in the traffic stream, such as conforming to speed limits, maintaining appropriate car-following behavior, and performing safe lane-change maneuvers.

## 1.2 Scope

The main scope of our work involves the application of specification-based monitoring to a highway scenario, to be able to observe whether the traffic flow satisfies or violates the specified traffic requirements. Our focus is on formally defining microscopic traffic parameters as well as safety-related indicators such as lane-change maneuvers and post encroachment time.

## 1.3 Objectives

This work proposes a new approach to traffic monitoring using formal methods. The approach is to develop specification-based monitors using a formal language to define safety-related microscopic traffic behavior, such as speed and headway. The next step is to extend the developed monitor to include lane-change detection and safety-related specifications. To demonstrate the value of specification-based monitoring the monitors are applied to a highway scenario to obtain robustness results. We also implement a data dissemination algorithm using inter-vehicular communication. The purpose of this algorithm is to provide another case study for the application of the proposed specifications. The output of this framework can be utilized by traffic management centers for the purposes of understanding the traffic stream properties, define hazards and produce solutions.

## 1.4 Contributions

In this work, we define formal specifications using Signal Temporal Logic to describe microscopic traffic behavior. A calibrated highway scenario is implemented using the micro-simulator, SUMO, and the developed specifications are then applied to the extracted vehicle trajectories. Finally, a vehicular communication algorithm is implemented for the online adjustment of individual vehicles acceleration to observe the impact such communication can have on safety enhancement.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Formal Methods in Intelligent Transportation Systems

The application of formal methods in the transportation field is gaining traction. The majority of the research in this area is focused on ensuring the overall safety of the transportation systems. In this section, we first discuss the use of formal methods in the field of verifying the safety of autonomous vehicles, followed by their application in transportation systems as a whole.

Arising from the need to replace classical modelling and simulation methods, researchers are shifting towards the use of formal methods for defining safety properties of autonomous vehicles. It has been proven that the use of statistical data-driven approaches for this problem is not feasible, as they require huge amounts of data to reach satisfactory results that can be accepted by the public [9]. Therefor, formal methods are often used in the verification of inter-vehicle systems for autonomous vehicles. They are employed to ensure the correct operation of the most critical components such as decision making and software control modules [10].

Various studies discuss the different aspects of autonomous vehicle safety such as; collision avoidance, lane change maneuvers and adaptive cruise control. In [9], Shalev-Shwartz *et al.* aim to exhaustively and formally define a safety model for the nominal safe operation of self-driving vehicles. This is achieved by defining different properties and scenarios that the AV might find itself in and the proper response to each scenario. They also define both the driving policy as well as the sensing system that would achieve a safe and comfortable experience. A different approach for using formal methods, was the application of runtime monitoring of a cooperative adaptive cruise control (CACC) system by Mao *et al.* [11]. They defined temporal specifications for the safe operation of CACC and the results showed that their approach can capture specification violation correctly. Pek *et al.* [12] formally defined lane change maneuvers. This was done by formalizing traffic rules then verifying the motion plans of the vehicle, ensuring that if a collision occurs during a lane change maneuvers, the autonomous vehicle will not be the blame. Formal methods can also be used to ensure that a self-driving vehicle will avoid static objects as well as dynamic obstacles on the road. This was studied and discussed by Atlhoff *et al.* [13].

On the other hand, there's research concerned with the development of formal models for the improvement of the traffic system as a whole. For example, the work by Mitsch *et al.*[4] was one of the early attempts to utilize formal verification tools in the modelling of freeway dynamics. The objective was to ensure the system correctly calculates the appropriate speed limit and communicates this information to vehicles in certain regions of interest, this would ultimately create an online variable speed limit system. Differential dynamic logic was used for the formulation and verification of the system specifications.

Coogan and Arcak [14] propsoed Linear Temporal Logic (LTL) specifications to define a ramp metering control strategy by using a Cell Transmission Model for the traffic system. CTM defines the system as a series of flows being controlled by upstream demands and downstream supplies. Multiple LTL formulae were proposed to ensure the design objectives of the controller are met. These objectives include: safety, reachability and liveness of the overall system. Building further on this work, Kim *et al.* [15] used the same concept of upstream demands and downstream supplies to build traffic controllers in sub-networks that would guarantee the soundness of the global network controller. This was achieved by creating supply and demand contracts between the sub-networks using LTL specifications. The concept of contracts was also explored by Müller *et al.* [16] to show that proving the safety of one component, ensures the safety of the overall traffic system. A directed graph was developed to model the signalized intersection and differential dynamic logic is used for the formal verification of the whole system. Other work by Coogan *et al.*[17], [18] formalized controllers for signalized intersections using LTL to achieve more efficient traffic networks. It was argued that compared to traditional control approaches, LTL offers a rich set of expressions to address more complex traffic requirements. For example, the latter developed LTL properties such as "a signal's state cannot change twice in two periods", which would ensure the correct operation of the signal controller.

Even though LTL is a powerful specification language for system description, it can be argued that it is not rich enough to capture all the complexities of the traffic dynamics. Hence, other temporal logic languages have been proposed for such purposes. The study conducted by Mehr *et al.* [19] used Signal Temporal Logic (STL) to define a predictive ramp metering system for a highway in a stochastic approach. Another study by Rashid *et al.* [20] focused on defining the macroscopic traffic parameters using High Order Logic specifications, while the work by Sadraddini *et al.* [21] used Metric Temporal Logic (MTL) for the purpose of formalizing traffic signal control.

The formalization of monitoring and management of the overall smart city applications is also worth mentioning. Ma *et al.* [7] used STL specifications to create a runtime monitoring system for the improvement of safety and performance of various smart city services. These services include: environment, transportation and emergency. Ma *et al.* [22] created a new formal verification language specifically for the use of formally define different services within a smart city. This new language extends the temporal properties from STL by adding a spatial component to it. However, none of these studies focuses on the detection and analysis of safety-related aspects in the traffic stream using formal methods. Our work defines microscopic traffic parameters, with a focus on ensuring the safe bounds of these parameters. We also use the formal language (STL) to detect lane change maneuvers and study their safety through formally defining safe time gaps during maneuvers as well as post encroachment time.

## 2.2 Surrogate Measures of Safety

Historically, crash data has been the prominent measure used in road safety research. However, there is a shift in the research community towards using surrogate safety measures instead. Surrogate safety measures define traffic events that are deemed as possibly hazardous or fatal to participating road users. These unsafe traffic events, also called traffic conflicts, include road users coming in close spatial or temporal proximity to each other, where if no evasive action is applied, collision is bound to happen. Therefore, it can be argued that observing and analyzing these traffic conflicts, can be an indicator for possible collision between the road-users. There are several benefits of using these quantitative measures. First, these events are more recurrent than crashes; hence it is possible to obtain more data for analysis. Second, since they are quantitative measures, it is easy to avoid subjective limitations of data obtained by observers. Finally, they have been used in multiple safety-related research therefor comparison with previous work is possible [23].

Multiple conflict indicators have been proposed in the literature such as: time-to-collision (TTC), post encroachment time (PET), gap time, deceleration rate (DR) and others. TTC and PET are one of the most popular measures used in research for measuring traffic conflicts [24]. TTC is defined as *"The time required for two vehicles to collide if they continue at their present speed and on the same path"* [25]. It can be inferred that TTC is calculated by extrapolating the vehicles trajectories, while assuming constant velocity and unchanged course of collision [3]. On the other hand, PET is defined as *"the time difference between the moment an "offending" vehicle passes out of the area of potential collision and the moment of arrival at the potential collision point by the "conflicted" vehicle possessing the right-of-way"* [26]. This behavior is illustrated in figure 2.1; where we can see the first vehicle leaving the conflict point $x_{cp}$ at time t1. At t2, the second vehicle arrives at the conflict point. According to the definition above, PET in this case is calculated as the time difference between the first vehicle leaving $x_{cp}$ and the second vehicle arriving at the same point ($PET = t2 - t1$). However, conflict will exist if the calculated PET value is below a certain threshold which would indicate the severity of the conflict if it had taken place.

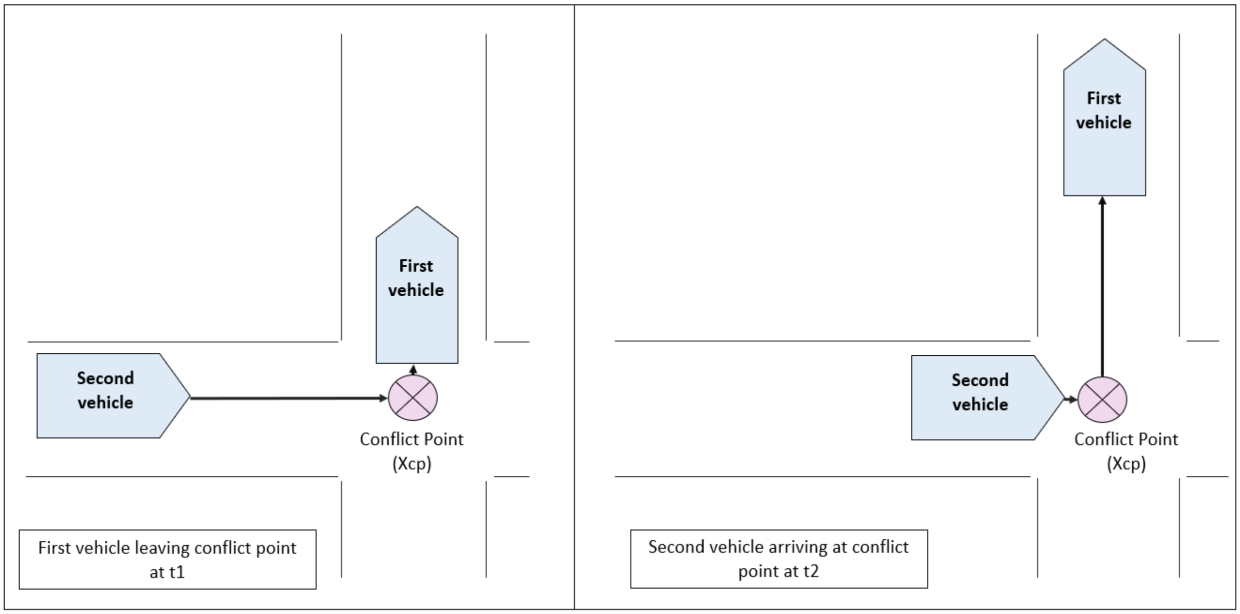


Figure 2.1: Post encroachment time (PET)

### 2.3 VANETs in Safety Applications

One of the key components of the Intelligent Transportation System is the Vehicular Adhoc Networks (VANETs). They span a wide range of applications such as infotainment, information sharing among vehicles and safety. In this section, we take a closer look at the safety-related applications for VANETs; focusing on Cooperative Awareness Applications. The main goal of these applications is the periodic dissemination of vehicle dynamics information for the enhancement of the overall road safety. Hence, beacons, which are also called Cooperative Awareness Messages (CAMs), were developed for the purpose of communicating information such as the position, velocity and acceleration of the broadcasting vehicle. On receiving this data, neighboring vehicles are then able to adjust their own dynamics to fulfil a certain application goal.

However, there are two challenges regarding the design and implementation of such algorithms, namely the congestion and awareness control. Congestion control deals with the quality of the communication channel and ensuring that its load doesn't surpass certain thresholds [27],

[28]. On the other hand, the main goal of awareness control algorithms is to adjust the rate of sending messages based on the vehicle's environment, especially safety-critical situations. For example, Aznar-Poveda *et al.* [29] proposed a vehicle control algorithm that takes into consideration the surrounding traffic condition as well as neighbouring vehicle dynamics in order to generate CAMs. Their approach uses the Time-to-collision measure in order to decide the priority of broadcasting certain messages; hence also providing better channel utilization. They also tested their algorithm on multiple scenarios such as free-way, congested roads and a realistic traffic jam, while using single and multi-hop approaches. In addition to having an efficient and reliable communication protocol to guarantee the delivery of critical messages, it is also important to properly define the application that will exploit this information to improve the road safety. As a result, Cooperative Adaptive Cruise Control (CACC) (aka, platooning) was developed to achieve this purpose. Fernandes and Nunes [30] propose five application-aware cooperative communication algorithms for CACC. The base algorithm proposed starts with leader vehicle transmitting its velocity and acceleration information to following vehicles, which then broadcast data in a cascading fashion towards the end of the platoon. The transmitted information allows each vehicle to calculate its own acceleration which would ensure a stable platoon with the appropriate time headway. The algorithm proposed by Segata *et al.* [31] use velocity and acceleration as inputs to a CACC controller. They mainly focused on enhancing the communication scheme by employing synchronized communication slots as well as adjusting transmission power.

# CHAPTER 3: PRELIMINARIES

## 3.1 Specification-based Monitoring

Cyber-physical systems generate various types of data which can be in the form of time series data, wave forms or signals [32]. One of the major tasks when designing and testing these systems, is to analyze their outputs to understand whether they are behaving as designed or not. This evaluation is realized through the monitoring of their temporal behavior; which can vary in length and can hold many variables and events, hence carrying complex information. Since this type of evaluation and testing can be time consuming and complicated, mathematical functions have been proposed for the task of mapping these complex behaviors into logical values which would indicate whether the performance of these systems are as expected. This approach is known as specification-based monitoring, which is derived from the formal verification approaches that focus on the specification of the system requirements using formal languages.

As a result, specification-based monitoring has been utilized for the purpose of ensuring that systems conform to the designer's intention. This is achieved by the rigorous specification of the system requirements using formal languages. Many formal languages have been proposed for system specifications that use temporal logic or regular expressions. However, CPS have special requirements as such systems include both physical and cyber components and producing dense temporal continuous signals. Therefor, there is a need for formal languages to address these challenges. Signal Temporal Logic (STL) has emerged as one of the new prominent formal languages in this field [33], [34]. It uses predicates on numerical values as well as atomic propositions. In the next section, we discuss the syntax and applications of STL.

### *3.1.1 Signal Temporal Logic*

Signal Temporal Logic (STL) is a formal specification language that is used to describe systems with real-valued, continuous signals. It was first proposed by Maler and Nickovic [33] as an extension to Metric Temporal Logic (MTL) to formalize the properties of cyber-physical systems (CPS). It defines predicates that enable reasoning over real-time properties of systems that exhibit both continuous and discrete dynamics.

STL specifications are defined using a combination of atomic predicates, boolean and temporal operators [32]. The boolean operators include negation $\neg$, conjunctions $\vee$ and disjunctions $\wedge$, while temporal operators include (Always) $\Box$, (Until) $\mathcal{U}$ and (Eventually) $\Diamond$. There are two types of temporal operators, namely future and past operators. In our work, we focus on using the future operators where the satisfaction of such operators at position $t$ is dependant on the signal's value from $t$ onward. For example, $\Box p$ is true if and only if $p$ holds every $t' > t$. It is also worth mentioning here that the temporal operators can be true either over a certain time range or infinitely. For example, $\Box_{[a,b]} p$ is true if p holds during $t' \in [t + a, t + b]$. This is particularly useful when used with signals as it allows events to occur anywhere within this interval where, unlike discrete time, it is independent of sampling points or clock ticks. In addition to the previous specifications, the definition of instantaneous events can also be useful in defining more complicated specifications, as they enable the specification of rising or falling edges in signals.

Below is the grammar used to define STL specifications:

$$\varphi ::= \mu \mid \neg\mu \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \Box_{[a,b]}\varphi \mid \varphi\, \mathcal{U}_{[a,b]}\, \psi \mid \Diamond_{[a,b]}\varphi \tag{3.1}$$

The STL semantics are defined as the satisfaction relation $(\xi, t) \models \mu$, which means that the signal $\xi$ satisfies the specification $\mu$ at some point $t$ if and only if, there exists a real-valued function of the signal such that $\mu(\xi(t))$.

Furthermore, the satisfaction of STL formulae is denoted by the below specifications:

$$(\xi, t) \models \mu \leftrightarrow \mu(\xi(t)) > 0$$

$$(\xi, t) \models \neg\mu \leftrightarrow \neg((\xi, t) \models \mu)$$

$$(\xi, t) \models \varphi \wedge \psi \leftrightarrow (\xi, t) \models \varphi \wedge (\xi, t) \models \psi$$

$$(\xi, t) \models \varphi \vee \psi \leftrightarrow (\xi, t) \models \varphi \vee (\xi, t) \models \psi$$

$$(\xi, t) \models \Box_{[a,b]}\varphi \leftrightarrow \forall t' \in [t + a, t + b], (\xi, t') \models \varphi$$

$$(\xi, t) \models \Diamond_{[a,b]}\varphi \leftrightarrow \exists t' \in [t + a, t + b], (\xi, t') \models \varphi$$

$$(\xi, t) \models \varphi\, \mathcal{U}_{[a,b]}\, \psi \leftrightarrow \exists t' \in [t + a, t + b], (\xi, t') \models \psi \wedge \forall t'' \in [t, t'], (\xi, t') \models \varphi$$

Since STL provides powerful expressive capabilities, its use in transportation is gaining traction. Many studies are using it to describe traffic systems since such systems are becoming more integrated by combing both cyber and physical aspects [7], [19], [32], [35]. We observe the vehicle trajectories as continuous real-valued signals, therefore we use STL to formalize the specifications that can then be used for the monitoring of traffic behavior.

## 3.2 Implementation tools

Our proposed framework consists of multiple simulation tools. In this section, we go over these tools to explain how they were employed in our approach.

### *3.2.1 SUMO*

Our approach uses micro-simulation in order to retrieve the various traffic parameters and use them as input to the monitor. Micro-simulation is a valuable tool for understanding the traffic dynamics and conditions for the accurate design and implementation of traffic solutions, especially traffic safety problems. SUMO [36] is an open-source traffic simulator, which provides various tools and packages for every step of traffic network simulation. It comes with ready to use and adjustable car-following model which allows for a flexible calibration and validation process. It also provides SSM devices which can be attached to vehicles to record conflicts and interactions between vehicles and calculate safety surrogate measures such as time-to-collision (TTC), Post Encroachment Time (PET), DRAC (Deceleration to avoid crash) and others. In order to start the simulation, XML files are automatically or manually generated to describe different aspects of the simulation. This includes the network itself, the traffic demand, and additional traffic infrastructure elements such as traffic lights or loop detectors. After the simulation is run, different XML or CSV output files are generated which contain all simulation results such as speeds, positions, accelerations of the vehicles and much more.

SUMO also has an implementation for the TraCI protocol [37]. TraCI is a communication protocol that allows the online access to the simulation for the retrieval of simulation parameters and the online manipulation of their behavior. It also provides a variety of vehicle-related information encapsulated in messages that are easily communicated between different platforms. This allows for the easy integration of other simulators with SUMO as TraCI offers a flexible interface for the online and direct interaction with the vehicles while the simulation is running.

### *3.2.2 OMNeT++ and VEINs*

One of the platforms that are able to interface with SUMO through TraCI is OMNeT++ [38], [39]. OMNeT++ is a discrete event-based communication network simulation tool written in C++. It first became available in 1997 and has been heavily used by the research community as it is both a powerful and reliable tool as well as being open source. The basic OMNeT++ model consists of modules that communicate with each other through exchanging messages. These modules can be further grouped together to create more complex modules known as compound modules. The main strength of OMNeT++ is in its modularity which facilitated the creation of the VEINs plug-in. VEINS [40] is an OMNET++ project that provides a channel between SUMO and OMNeT++. It communicates with SUMO using the TraCI protocol; to retrieve different road and vehicle related parameters. On the other hand, it also passes this information to OMNeT++ to simulate the vehicles as mobile communicating nodes. In this case, SUMO is acting as a server, while VEINs is acting a client that sends and receives traffic parameter information from SUMO and communication related parameters from OMNeT++. VEINs provides all underlying communication layer implementation which makes it straightforward for focusing on the implementation of vehicular communication applications.

#### *3.2.3 Breach*

Breach is a Matlab toolbox that provides simulation-based techniques for analyzing hybrid dynamical systems [41]. It was initially implemented to use the Metric Interval Temporal Logic (MITL), and was further extended to support a library for defining specifications using Signal Temporal Logic (STL) formal language. Breach's main strength is that it can be readily used for specifying scalable hybrid non-linear systems. This provides the capability to be used for synthesizing runtime monitors for investigating the properties of large sets of trajectories and their satisfaction robustness. Breach provides both online and offline monitoring capabilities which can be used to study the trajectories produced by systems while they are running or afterwards to check for satisfaction robustness. This tool has been employed for the computation and investigation of the properties of large sets of trajectories. It has been used in several research from simulation-based design and evaluation of CPS systems [42], to verifying artificial intelligence (AI) systems to ensure the correctness of their behavior through formally specified requirements [43].

# CHAPTER 4: METHODOLOGY

## 4.1 Proposed Framework

In this chapter, we discuss the methodology and framework that was followed in our work. The methodology is split into two tasks; first is the traffic network simulation and the second is the formalization of the traffic behavior. As shown in figure 4.1, the first task starts by traffic data collection and preparation from detector data, followed by the creation of the network using the micro-simulator. The traffic network is then calibrated and the simulation is run. The output of the traffic simulation is the set of vehicle trajectories that are then used as an input to the developed specification-based monitor. As for the second task, we first started by collecting various traffic-related rules with a focus on safety; such as speed limits, headways, and safe lane-change maneuvers. Then, these requirements were formalized using the formal verification language: Signal Temporal Logic (STL). The different formalized properties are then aggregated into a runtime monitor that takes as an input the vehicle trajectories that were extracted from the first phase. Finally, the monitor is applied to these trajectories and their conformity/violation to the specifications is then studied.

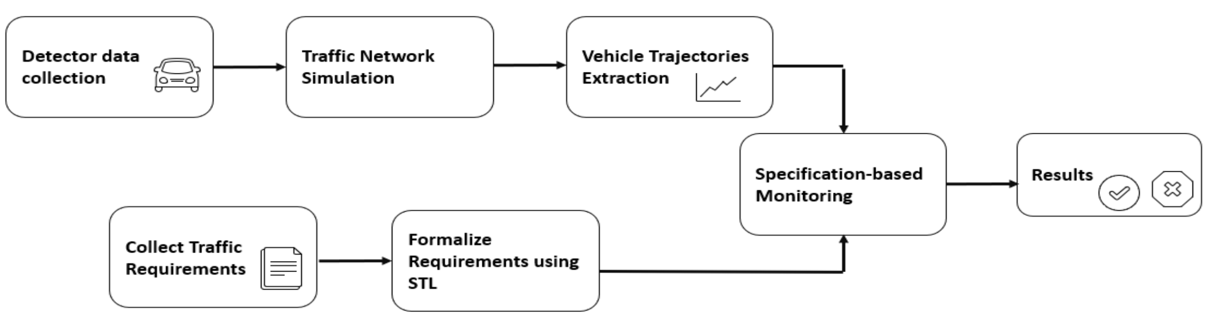



Figure 4.1: Methodology

In order to implement this approach, various tools were employed as shown in figure 4.2. This framework explains the relationship between the different tools and components of our approach. First, the obtained traffic data is used as the input to the road traffic simulation using SUMO. After the trajectories are extracted, they are then used as an input to the runtime monitoring tool, Breach, which analyzes the conformity of these trajectories to the proposed STL specifications. Another goal of this work is to study the impact inter-vehicle communication can have on the safety of the road-users. Therefor, we use the VEINs framework to realize the simulation of connected vehicles through the integration of both the road traffic simulator SUMO and the communication network simulator OMNeT++. We also propose an application layer vehicular communication algorithm that disseminates safety-critical information among vehicles. This algorithm is implemented within VEINs and is further discussed in a later section.

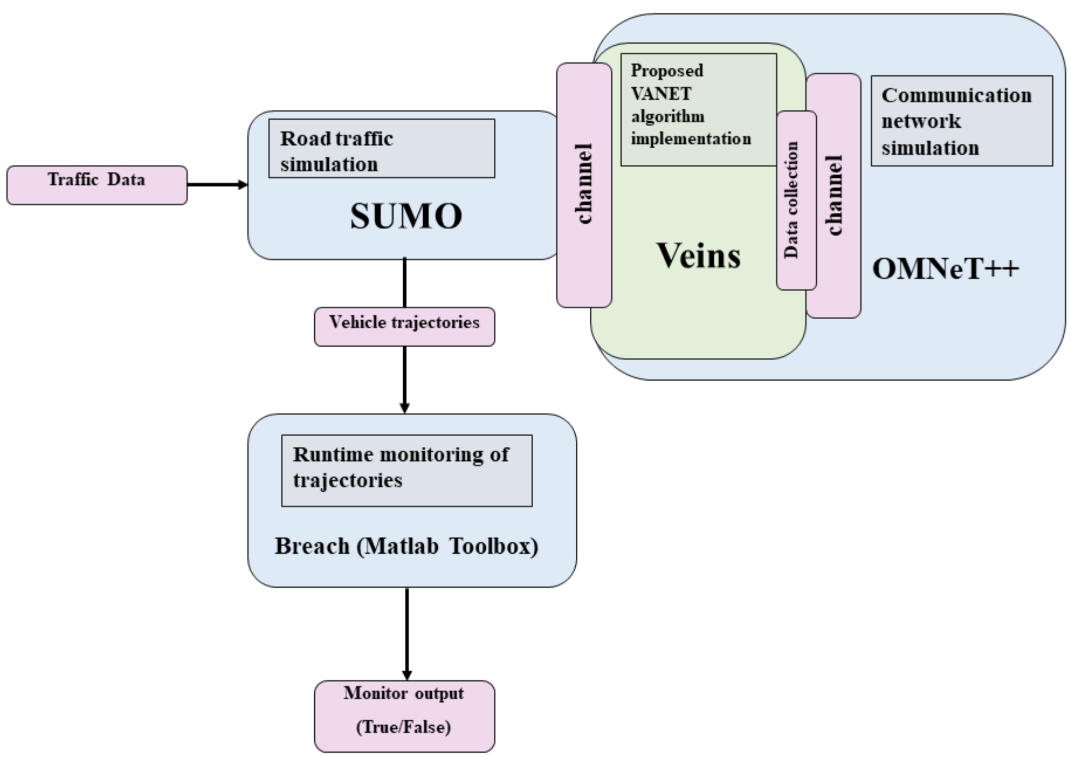



Figure 4.2: Proposed framework

## 4.2 Monitoring of Microscopic Traffic Parameters

Inspired by the work of Bartocci *et al.* [32], we use Signal Temporal Logic (STL) to formally define microscopic traffic parameters. In each of the following subsections, we start by defining these parameters in plain language and then continue to propose STL formulae to formally describe them.

### *4.2.1 Speed-related Specifications*

On a highway, vehicles must abide by certain speed-related rules. Vehicles must not accelerate in such a way that would exceed the maximum speed limit. Also, vehicles must not decelerate below the minimum speed limit as this can also pose a risk for surrounding vehicles and the traffic flow as a whole [4]. Below is the formal definition of these speed requirements using STL language:

$$\begin{aligned}\square[(V_{min} \leq V_s \leq V_{max}) \vee ((V_s > (V_{max} + V_{err}))& \\ \implies \Diamond_{[0,t]}(V_s \leq V_{max}) \vee ((V_s < V_{min})& \\ \implies \Diamond_{[0,t]}(V_s \geq V_{min}))]&\end{aligned} \tag{4.1}$$

where:

$V_{min}$: minimum speed limit

$V_{max}$: maximum speed limit

$V_s$: subject vehicle's velocity

$V_{err}$: allowed speed error

Equation 4.1 states that *"The vehicle speed is always within the maximum/minimum allowed limits. However, if the vehicle speed increases/decreases with respect to the allowed speed limits (while allowing for some error), it eventually (during a predefined duration $t$) has to decrease/increase to be within the allowed speed limits"*. This formulation is then applied to individual vehicle trajectories to ensure that every vehicle on the highway will be abiding to the speed-related requirements.

Another consideration when studying speed profiles of vehicles is the deceleration behavior of vehicles and whether it is safe or not. Jerk, which is the rate of change of acceleration, has proven to be a powerful indicator of unsafe deceleration [44], [45]. Jerk can be used to differentiate between normal braking and sudden braking that is due to unsafe situations. Therefor, as a compliment measure to acceleration, jerk values can be used to accurately capture conflict situations. A threshold level of $-9.9m/s^3$ is used for the jerk as an indicator of safety-critical driving behaviour. Vehicles must also avoid decelerating abruptly as this will negatively impact the traffic flow. Hence, to define speed specification for decelerating vehicles, we need to take into consideration the safe deceleration of vehicles by including the jerk in the formulation.

Below is the STL formula for speed requirements while considering safe braking behavior:

$$\square[(A_s > -7.) \wedge (J_s > -9.9)] \tag{4.2}$$

where:

$A_s$: subject vehicle acceleration

$J_s$: subject vehicle jerk

which states that *"At all times, both the jerk and acceleration profiles of the vehicle must be below certain threshold, which would indicate comfortable and safe deceleration."*

In addition to defining the deceleration behavior in general, we can also study a special case in details to define the vehicle's speed profile during that time, which is when a vehicle is approaching an off-ramp. Vehicles intending to use an off-ramp might approach it with a higher speed (which might match the maximum speed limit of the main road). Therefor, these vehicles need to continue deceleration until their speed matches that of the off-ramp. It's noted here that this specification is applied only to vehicle trajectories that are merging onto an off-ramp. We can define this behavior using the following specification:

$$\begin{aligned}\square[(V_s \leq V_{sl})\vee\\ &((V_s > V_{sl}) \implies ((A_s > -7.7) \wedge (J_s > -9.9)\\ &\mathcal{U}\,(V_s \leq V_{sl}))]\end{aligned} \tag{4.3}$$

where:

$V_s$: subject vehicle speed

$V_{sl}$: maximum speed limit on an off-ramp

This formula states that *"The speed of the vehicle has to always be less than the speed limit, however, if it increases to be more than the speed limit, then the vehicle must continue to decelerate comfortably until its speed is less than or equal to the maximum speed limit allowed."*

Even though using high sampling rate can produce better resolution for acceleration and jerk, it can also be prone to high fluctuations and high noise [45]. Therefor, we use the weighted exponential smoothing method in Matlab to be able to smooth the data and minimize the noise.

### *4.2.2 Headway Specifications*

Vehicle headway is one of the fundamental microscopic traffic parameters, where it describes the temporal relation between consecutive vehicles. It can be defined as the time difference between a leader vehicle arrival at a designated point and the following vehicle reaching the same point.

According to [46] , it is recommended to keep a minimum following distance of 4 seconds during normal weather and traffic condition at all times. Therefore, we use the below STL formula to define the headway specification:

$$\square[(h \geq 4) \wedge ((h < 4) \Longrightarrow (\Diamond_{[0,t]}(h \geq 4)))] \tag{4.4}$$

where:

$h$: headway between every two vehicles following one another

Equation 4.4 states *that the headway should always be greater than or equal to 4 seconds. However, we allow for the headway to fall below this value but only for some time* $t$*, if it doesn't increase back to 4 seconds or more, then the specification is violated.* This definition ensures that the individual vehicles are always abiding by the headway specifications for a safe operation of the overall traffic flow.

## 4.3 Lane-change Maneuvers Specifications

This section aims to propose STL monitors for different lane change maneuver aspects including its definition and safety. We first define what constitutes a lane change; how to infer from a vehicle's trajectory that a lane change has taken place, how long was the maneuver and the interactions between the subject vehicle and the neighboring vehicles in current and target lanes. We then discuss three approaches that aim to differentiate between safe and unsafe lane change maneuvers.

### *4.3.1 Lane-change Detection*

Historically, lane change maneuvers were observed and recorded manually using traffic camera data. However, there has been an interest in detecting these maneuvers by using vehicle trajectories extracted from detector or GPS data [47]. Inferring a lane change maneuver from a vehicle's trajectory is no straightforward task. A vehicle's lateral position can be disturbed easily by either unintentional deviations from the lane's center line or by noise in the data collection devices.

Therefor, the vehicle's lateral trajectory is observed as an offset from the center line of the lane, such that it can be compared to an ideal lane change trajectory. In figure 4.3, the y-axis represents the lateral offset position of the vehicle within the lane. The maximum and minimum values of such trajectory are bound by the 1/2 of the lane width. The vehicle is considered to be moving straight as long as its lateral shift is within $\pm 1m$ from the lane's centre line [48]. Once the vehicle initiates a lane change, its lateral offset increases until it reaches the highest bound (represented by 1/2 the lane width) and then it instantly drops to the lowest bound once it crosses over to the next lane. According to Das *et al.* [48], a lane change maneuver can last up to $15s$ and its end can be determined when the vehicle's lateral position settles at $\pm 4cm$ from the lane's centre line.

We first define an STL formula that describes when a vehicle is going in a straight line, and use this definition to further specify the lane change behavior:

$$straight \equiv 0 \leq |S_{lateral}| \leq 0.04 \tag{4.5}$$

$$lanechange \equiv (|S_{lateral}| \geq 1.0) \wedge \Diamond_{[0,15]}(straight) \tag{4.6}$$

where:

$S_{lateral}$: is the lateral position offset of the subject vehicle

Formula 4.5 states that *if a vehicle's lateral position offset is within the values* $0m$ *and* $0.04m$*, then it follows that the vehicle is going in a straight line with no lateral deviations.* On the other hand, the formula 4.6 states that *if a vehicle's lateral position offset increases or decreases more than the threshold of* $1m$*, then it means that the vehicle has started a lane change maneuver and will eventually (within* $16s$*) go back to be in a laterally straight position.* It's noticed here that the equivalence symbol is used. This is because STL allows the definition of events that define the start of certain behavior (in this case, the start of a lane change) which can be further used in more complex specifications, as will be seen in the next sections.

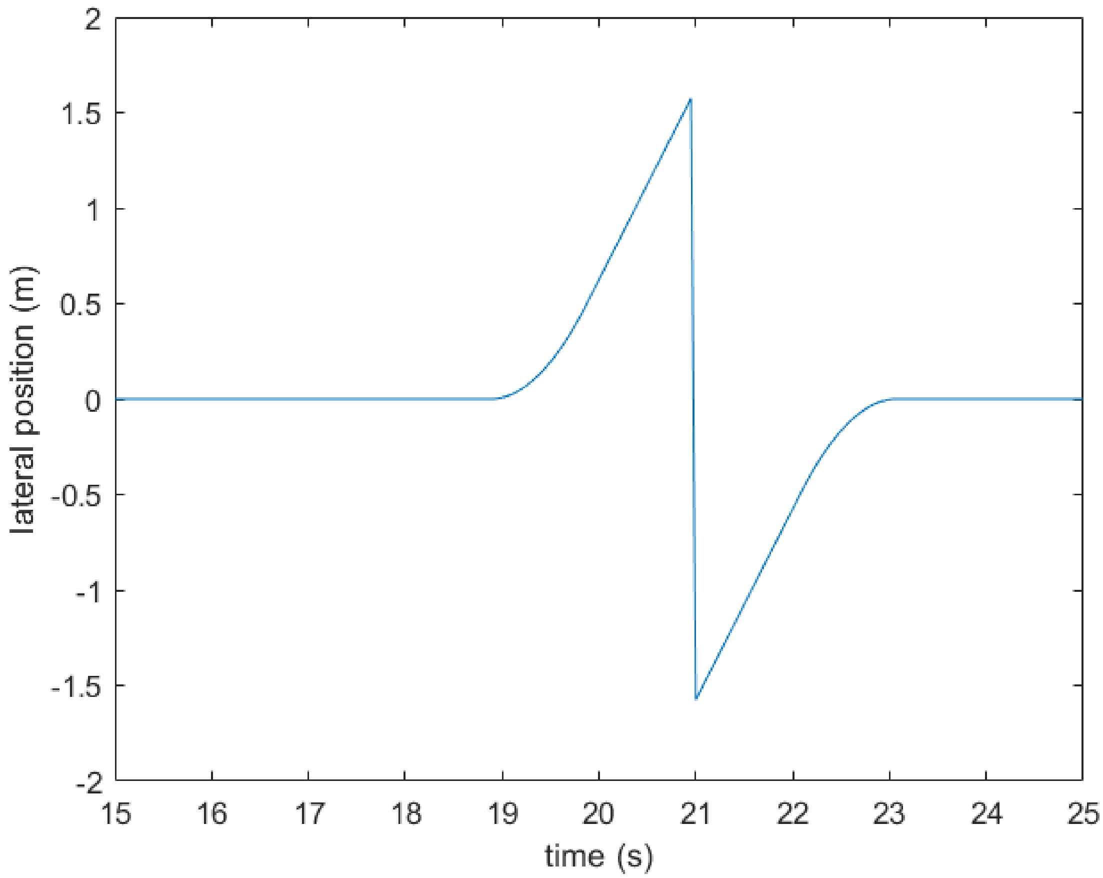


Figure 4.3: Lane Change Trajectory using lateral position offset

### *4.3.2 Lane-change Safety Monitoring*

Lane change maneuvers are an integral part of the traffic flow which can greatly impact its efficiency and safety. According to the NHTSA [49], lane changing maneuvers accounted for 4.6% of crashes in the United States in 2015. Therefor, a lot of research is being done to study the different aspects that drivers consider when initiating lane changes that could affect the maneuver's safety [50], [48], [51], [52]. Speed, lateral position, visibility and available gap in the target lane are some of the parameters that impact the driver's decision to start a lane change maneuver. For example, an unexpected lane change can cause surrounding vehicles to either brake hard or accelerate unnecessary, which would impact the traffic flow and might result in conflict situations [50]. There are two types of lane changes; namely mandatory lane changes (MLC) and discretionary lane changes (DLC) which have different gap acceptance behavior due to the different motivation behind performing either [53]. The motivation to initiate an MLC is dictated by the need to take an exit or a turn or to avoid an obstacle or workzones. On the other hand, DLC's motivation is the enhancement of the driving experience, where drivers change lanes for to gain speed or avoid a slow moving vehicle.

In this section, we propose the STL formulation of three different approaches for the detection of unsafe lane change maneuvers. Each of these approaches studies the lane change safety problem from a different perspective. First, we study the gap acceptance behavior of vehicles. Second, inspired by the work of Park *et al.* [54], we propose an STL formulation of the lane change risk index. Finally, we define and formalize the Post Encroachment Time (PET) surrogate safety indicator to further understand conflicts that can arise during lane change maneuvers.

#### *4.3.2.1 Method 1: Gap Acceptance Behavior*

At the onset of a lane change maneuvers, the driver has to consider neighboring vehicles as they play an important role in deciding how, why and when the maneuver can take place. In figure 4.4, the subject vehicle is performing a lane change maneuver by moving from the current lane to the target lane. In the most general scenario, it is assumed that there are four vehicles surrounding the subject vehicle; where there are two in the current lane and two in the target lane. Therefor, gap acceptance in the target lane is a key factor for the decision to start the maneuver. Based on the observed gap between the lag and lead vehicles in the target lane, the driver decides whether it is safe to initiate the lane change maneuver or not; hence accepting or rejecting the gap [50]. It is clear that the value of the accepted gap impacts the safety of the traffic environment greatly, as a result, there's a focus on studying the gap acceptance behavior. This can be directly correlated to the safety implications of lane changes as well as building accurate micro-simulation models [48].

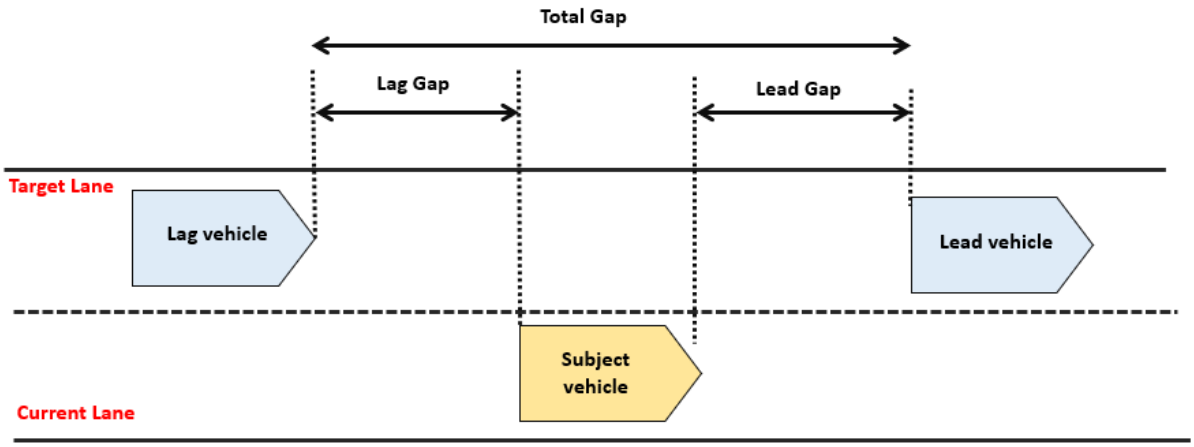



Figure 4.4: Lane-Change Maneuver

According to Yang *et al.* [50], the mean of the accepted lead gap on highways is $1.48s$, while the mean for the accepted lag gap is $1.37s$, which sums up to a total critical gap of $2.85s$. These values are used for the specification of an STL formula that will be further employed for the monitoring of gaps between vehicles during lane changes. It is also assumed that every vehicle has information about the distances between itself and neighboring vehicles using vehicular communication that will be discussed in a later section.

$$(lanechange) \implies \Box[(t_{lag} \geq 1.37) \wedge (t_{lead} \geq 1.48))] \tag{4.7}$$

where:

$t_{lag}$: lag gap between subject vehicle and lag vehicle in the target lane

$t_{lead}$: lead gap between subject vehicle and lead vehicle in the target lane

This formula states that *if there is a detected lane change, then both the lead and lag gap values have to always fall within the safe bounds defined in the formula.* It noted also that this can be calculated through merging both values and comparing with the total critical gap.

#### 4.3.2.2 Method 2: Stopping Distance Index

Building on the gap acceptance approach in the previous section, we define the Stopping Distance Index which was first proposed by Park *et al.* [54]. In this paper, various measures are calculated for the assessment of crash risk during lane change maneuvers based on improper interactions. This is done by continuously evaluating the risk of lane change events. They first define the Stopping Sight Distance (SSD) which is the worst-case distance a vehicle driver needs to be able to see in order to have room to stop before colliding with something in the roadway.

$$SSD = 0.278V_s t_r + \frac{0.039V_s^2}{a_s} \tag{4.8}$$

where:

$V_s$: subject vehicle velocity

$t_r$: perception reaction time ($= 2.5s$)

$a_s$: subject vehicle acceleration

The SSD is then used to calculate the Stopping Distance Index (SDI) which is continuously calculated during a lane change maneuver and then determines the read-end collision risk. For example, an SDI greater than $0$ means that the subject vehicle is able to stop completely and safely when the lead vehicle stops suddenly. However, a value less than $0$ presents a situation where the lead vehicle makes a sudden stop and the subject vehicle doesn't perform an appropriate evasive maneuver that would prevent a conflict between the two vehicles. Below is the SDI definition for the subject vehicle compared to the lead vehicle:

$$SDI_{subject,lead} = S_0(t) + SSD_{lead}(t) - SSD_{subject}(t) - L_{lead} \tag{4.9}$$

where:

$S_0(t)$: spacing between subject vehicle and lead vehicle

$SSD_{lead}(t)$: SSD for lead vehicle

$SSD_{subject}(t)$: SSD for lag vehicle

$L_{lead}$: length of lead vehicle

Based on equations 4.8 and 4.9, we propose the below STL formulation for the formal specification of the SSD:

$$\square[lanechange \implies (SDI_{subject,lead} > 0 \wedge SDI_{lag,subject} > 0)] \tag{4.10}$$

where:

$SDI_{subject,lead}$: SDI for the subject and lead vehicles interaction during a lane change maneuver

$SDI_{lag,subject}$: SDI for the lag and subject vehicles interaction during a lane change maneuver

which states that: *"If there is a lane change, then at all time steps during the lane change, the SDI between the lag and front vehicle and the SDI between the front and lead vehicle must be greater than 0."*

#### *4.3.2.3 Method 3: Post Encroachment Time*

Zheng *et al.* [51] proposed the Post Encroachment Time (PET) as an appropriate indicator for understanding the safety of lane-change maneuvers. For calculating PET in general, there needs to be a clear definition of the encroachment line or conflict area and this is different in the case of a lane change since there is no clear area where the trajectories of the two vehicles overlap. Hence, the definition of PET is adjusted to take into consideration the unique properties of a lane-change behavior. Therefor, the encroachment line is considered to be at the point where the subject vehicle crosses the lane line dividing the origin and target lanes. In this case, PET can be defined as the time between the subject vehicle leaving the encroachment line and the lag vehicle reaching it. Figure 4.5 illustrates PET calculation for a lane-change maneuver. At $t1$, the subject vehicle initiates a lane-change maneuver and the encroachment line is defined as the line that crosses both lanes where the subject vehicle's front bumper crosses from current to target lane. At $t2$, the lag vehicle arrives at the encroachment line. Therefor, PET in this case is defined as $PET = t2 - t1$.

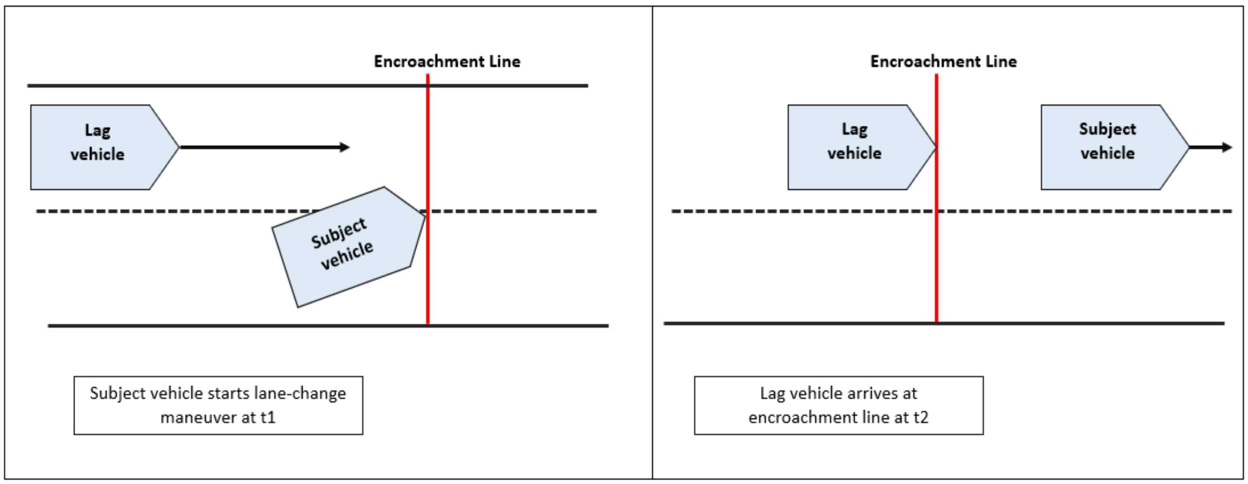



Figure 4.5: PET in a lane change scenario

To formally define this behavior, we propose the formula (4.11). It states that i*f there is a lane-change detected according to the formula 4.6, then for a PET to exist, the lag vehicle has to change its position from the initial x (defined as $x_{lc}$) and reach the conflict point (defined as $x_{cp}$) within* $2.25s$. This value was chosen according to the research on PET values during lane-change done by Qi *et al.* [55], which showed that if the calculated PET is less than or equal to $2.25s$ during a lane-change, then it follows that there is a serious to potential conflict between the participating vehicles.

$$(lanechange) \implies (\Diamond_{[0,2.25]} x_{lag} = x_{cp}) \tag{4.11}$$

where:

$x_{lag}$: position of the lag vehicle

$x_{cp}$: conflict point, where the subject vehicle initially crossed the encroachment line.

According to this specification, the input signal will satisfy it if the position of the lag vehicle is close enough to the conflict point so as to cause a conflict, and will violate the specification if the position of the lag vehicle is too far at the time the subject vehicle performs the lane-change.

## 4.4 Data Dissemination VANET Algorithm for Traffic Safety Improvement

In order to enhance the overall traffic safety, we propose the use of Vehicular Adhoc Networks for the dissemination of safety-critical information to vehicles. Our approach focuses on calculating appropriate accelerations for each vehicle to maintain safe distance to its preceding vehicle. In this section, we demonstrate a case scenario where using the proposed STL monitors can be applied to understand the traffic behavior. An example of the headway trajectories before and after applying the communication algorithm is explained and the comparison is done through using our proposed monitors.

In general, a follower vehicle is tasked with maintaining an appropriate headway to its leader vehicle at all times by regularly actuating its acceleration [30]. The calculated acceleration is based on: the follower vehicle's velocity, the leader's relative velocity and the spacing between both vehicles [56]. Vehicular communication can be utilized as a convenient method in conveying such information among the road-users.

The task of each follower vehicle is to maintain a certain ”desired minimum gap”, which is defined by the below equation:

$$s(v_\alpha, \Delta v_\alpha) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}} \tag{4.12}$$

where $s_0$ is the jam distance, reasonable ranges are from $1m - 5m$ so a value of $2m$ is used in our approach.

Equation (4.12) is the desired minimum gap between two vehicles according to the Intelligent Driver Model (IDM) [56]. The term $vT$ plays an important role in non-stationary scenarios, as it ensures a constant time gap regardless of the speed. It is also noticed that the desired minimum gap is directly proportional to the speed of the vehicles, which means that it increases with the increase of the speed. If maintained, this formulation of the desired gap guarantees collision-free behavior.

Building further on this concept, equation 4.13 defines the IDM acceleration function which takes advantage of the safety properties of the desired minimum gap equation (4.12). Each follower vehicle utilizes the equation 4.13 in order to calculate the appropriate acceleration to be applied in order to always maintain an appropriate inter-vehicle headway of $4s$.

$$\dot{v}_\alpha(s_\alpha, v_\alpha, \Delta v_\alpha) = a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right] \tag{4.13}$$

where:

$v_\alpha$: velocity of the follower vehicle

$\Delta v_\alpha$: velocity difference between follower and leader vehicles

$a$: maximum acceleration (a value of $1.4m/s^2$ is used)

$v_0$: the desired velocity

$T$: Safety time gap (a value of $4s$ is used)

$b$: Desired deceleration (a value of $2.0m/s^2$ is used)

In our proposed approach, we adopt a vehicular communication algorithm based on the work by Fernandes and Nunes [30]. Their approach focuses on minimizing the inter-vehicle gaps in platoons for better road utilization, while our approach aims at maintaining safe time gap between vehicles at all times. For every update cycle, vehicles broadcast their position and speed to their direct neighbors. It is assumed that each vehicle is able to calculate the distance to its leader, so if a vehicle is detected as a leader vehicle, the follower vehicle sends its position and speed information to it. Each follower vehicle maintains a record about its leader by constantly listening to beacons from the the leader that contains position and speed information. Both leader and follower vehicles update records according to vehicles entering and leaving the traffic stream through periodic beacons. Vehicles can be both a leader and a follower based on its position in the traffic stream.

The pseudo-code for leader and follower vehicles are explained in Algorithm 1 and Algorithm 2 consecutively.

**Algorithm 1:** Vehicle information updating algorithm: Leader

```
for all vehicles do
    Broadcast x_leader and v_leader;
    if receives message from follower then
        store follower information;
    end
end
```

**Algorithm 2:** Vehicle information updating algorithm: Follower

1 **for** *all vehicles* **do**
2     **if** *receives message* **then**
3         calculate $s_{leader}$ to leader vehicle;
4         **if** $(s_{leader}/v_{follower}) < 4s$ **then**
5             declare vehicle as leader;
6             send $x_{follower}$ and $v_{follower}$ to leader;
7             calculate new acceleration to be applied next time-step;
8         **end**
9     **end**
10 **end**

Initially, a vehicle is unaware whether it has the role of a leader or a follower or both. Therefor, all vehicles start by broadcasting their position and speed information. The following chain of events decides the role of the vehicle. If it detects another vehicle preceding it, then it is a follower. If it receives messages that encapsulate its own ID in addition to speed and position information, then it is assumed that this information was sent by a follower and it is assigned the role of a leader.

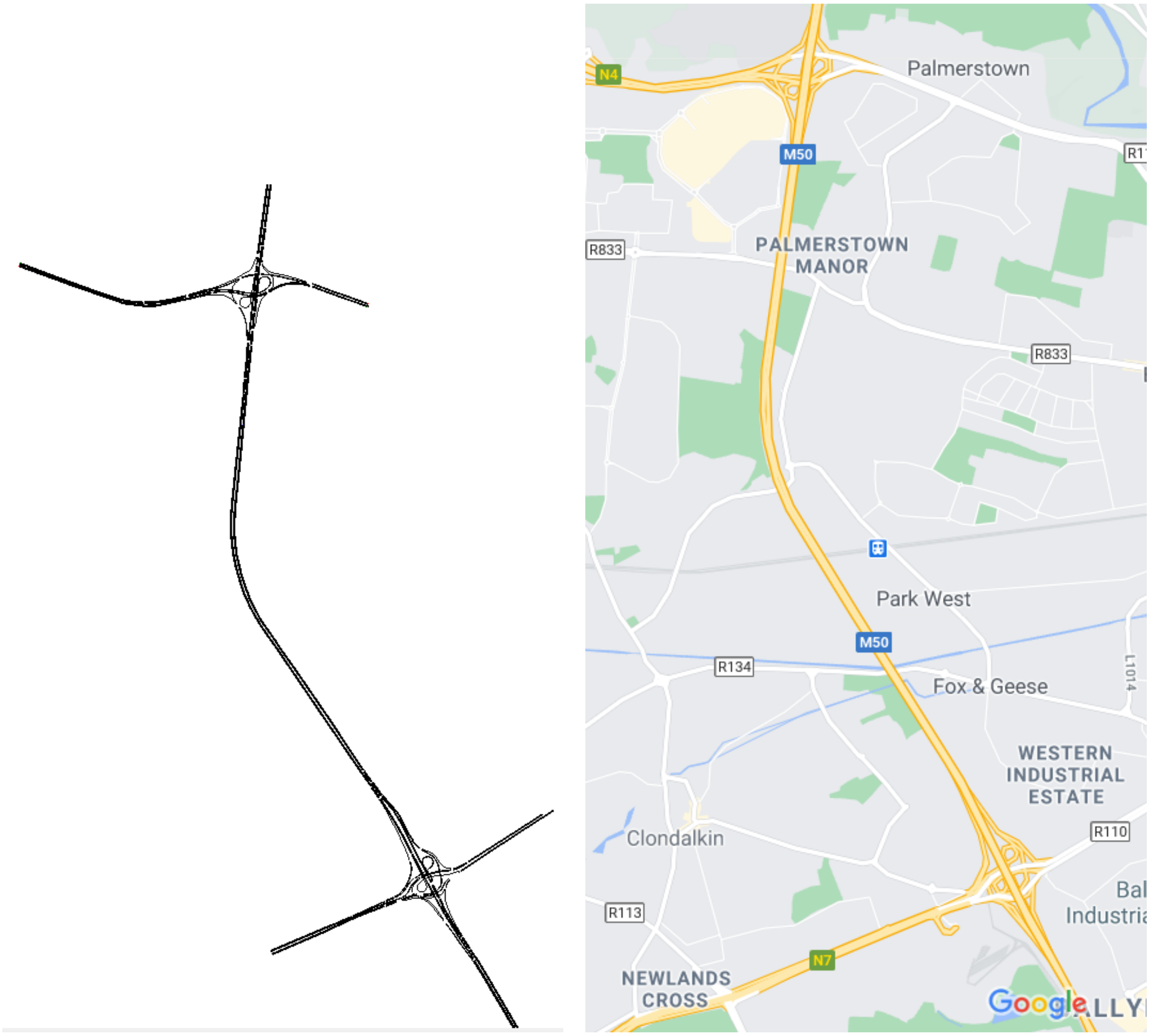
Algorithm (1) defines the behavior of a vehicle with the leader role. The vehicle starts by broadcasting its position ($x_{leader}$) and speed ($v_{leader}$). Lines 3-5 indicate that the vehicle has received a message from its follower and stores the corresponding information. On the other hand, algorithm (2) explains the behavior of the vehicle if it is assigned the follower role. Lines 2-3 indicate that the follower vehicle has received a message, that contains speed and position information. The distance to the vehicle ($s_{leader}$) is then calculated based on this information. If it is decided that the received information was from the preceding vehicle, then this vehicle is defined as the current leader. Then, the time headway to the preceding vehicle is calculated in lines 4-8 and if it is below a threshold of $4s$, then the follower calculates the appropriate acceleration to be applied according to equation (4.13).

# CHAPTER 5: CASE STUDY

In this section, we demonstrate a highway scenario to show the effectiveness of the proposed STL-specified traffic monitor to capture changes in the traffic flow parameters. The input to the monitor is each vehicle's trajectory and the output is the robustness measure of the overall satisfaction of each formula, as well as a plot showing where the trajectory was satisfying or violating the specification. After that, we apply the vehicular communication algorithm to the scenario, use the proposed monitors for further trajectory evaluation, and observe how the communication impacts the conformity of the vehicle's trajectory to the headway specification.

## 5.1 M50 Highway

We apply the proposed STL traffic monitor to a highway scenario obtained from [57], which was implemented and calibrated using the micro-simulator SUMO. The traffic network model simulated the M50 highway in Dublin, Ireland which is a 7KM 4-lane highway that includes two major interchanges (Figure 5.1). The traffic demand used in the model was generated using loop detectors from the Transport Infrastructure Ireland, during the morning peak hours (7-8am) with 5-minute aggregated traffic flows. The simulation is run using SUMO with a time-step of $50ms$ and vehicle-specific information are extracted. This information include: speed, longitudinal and lateral dynamics. The study [57] have adjusted several simulation parameters such as headway, minimum gap values and driver imperfection factors in order to simulate the actual driving behavior as closely as possible. They have used various types of vehicles such as human-driven vehicles and level 2 and level 4 connected autonomous vehicles. However, for the purposes of our approach, we will be using the scenario with the human-driven vehicles.




(a) M50 highway in SUMO

(b) M50 Highway from Google maps

Figure 5.1: M50 Highway

### *5.1.1 Speed-related Specifications Results*

The speed specification previously proposed was applied to the M50 highway traffic scenario. The simulation is run for $100s$ and speed data was recorded for individual vehicles. In figure 5.2, the speed of a single vehicle is plotted across the simulation time. By observing the speed plot, it can be noticed that this vehicle speed trace doesn't conform to the speed specifications as it is below the minimum speed limit allowed on the highway. We expect that the robustness satisfaction value for this trace would be negative; which means that the input trace doesn't satisfy the specification.

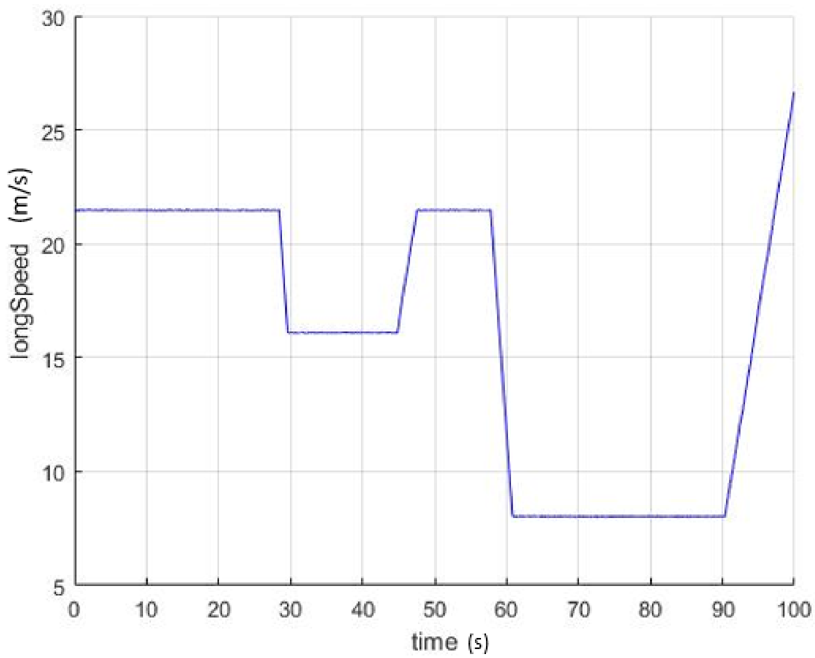



Figure 5.2: Speed Trace

Figure 5.3 shows the output of the speed monitor. It can be seen that the specification output is false throughout the trajectory of the vehicle, except at $97s$, where the vehicle adjusts its speed in such a way that it conforms to the proposed speed specification. After that time, the vehicle's speed is within accepted levels. The same approach can be applied automatically to all other vehicle trajectories in the highway scenario. The output would indicate whether vehicle trajectories are satisfying or violating the speed specifications on the highway.

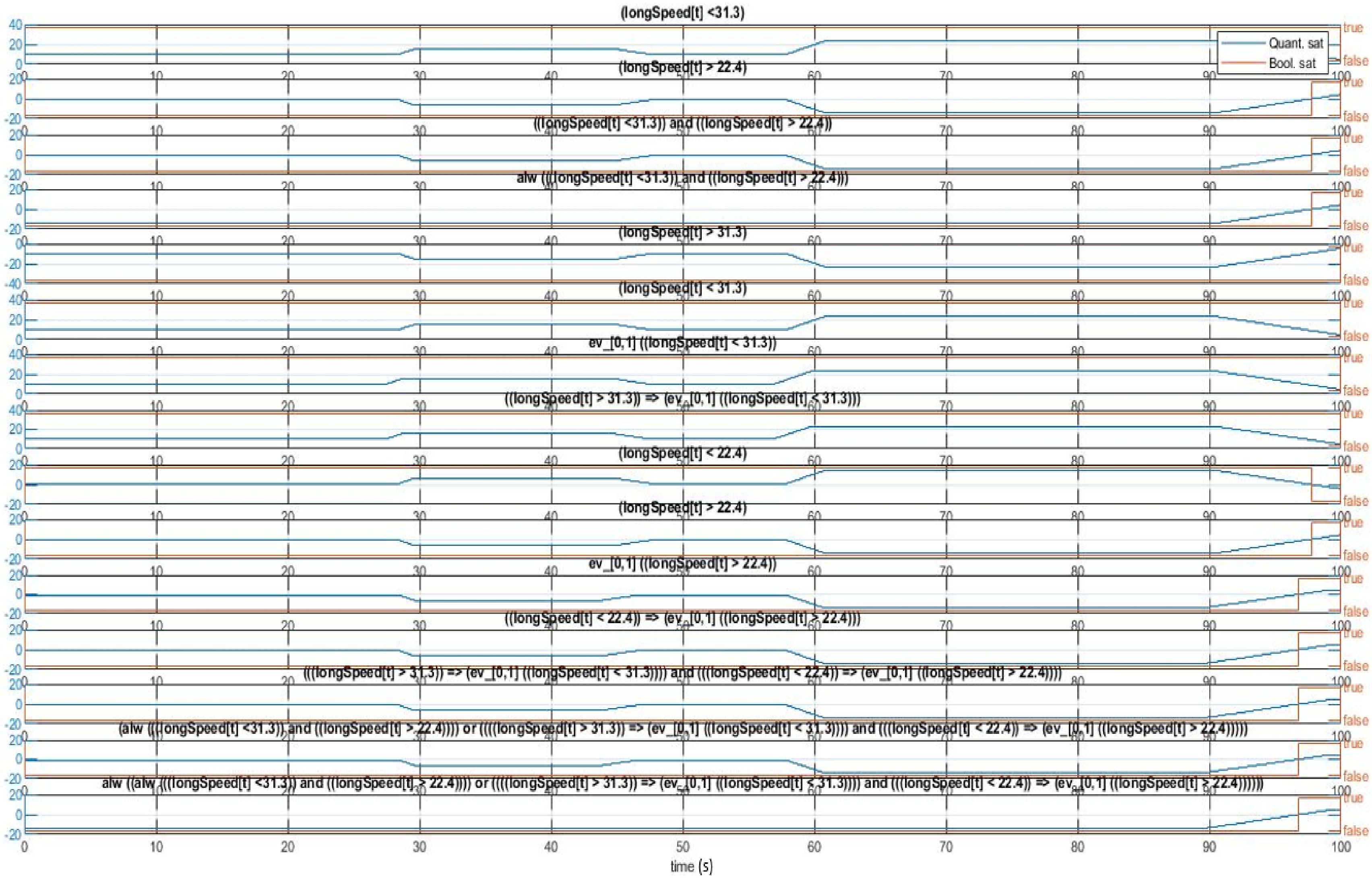


Figure 5.3: Speed Specification output

Speed is further investigated by applying the specifications defined by the STL formula 4.2 to individual speed trajectories. The main goal of this specification is to understand the braking behavior of vehicles in conflict situation. For safe and comfortable braking scenarios, vehicles jerk and deceleration values should be bounded within values expressed by the STL formula. The specification is applied to the individual vehicle trajectories and an individual trace is used for illustrating the results. By observing trace in figure 5.4, it can be noted that there is a decline in speed at $95s$. We would like to apply the specification to this trace in order to understand whether the deceleration taking place at that time was safe or not. It is worth mentioning here that the acceleration and jerk profiles of the vehicle were first subjected to an exponential smoothing method using a Matlab function. This was done to reduce the smooth the data and minimize the noise.

Figure 5.5 depicts the output of the monitor, which shows that the trace is indeed conforming to the specification. It is concluded that throughout the simulation, this vehicle was exhibiting safe speed and braking behavior.

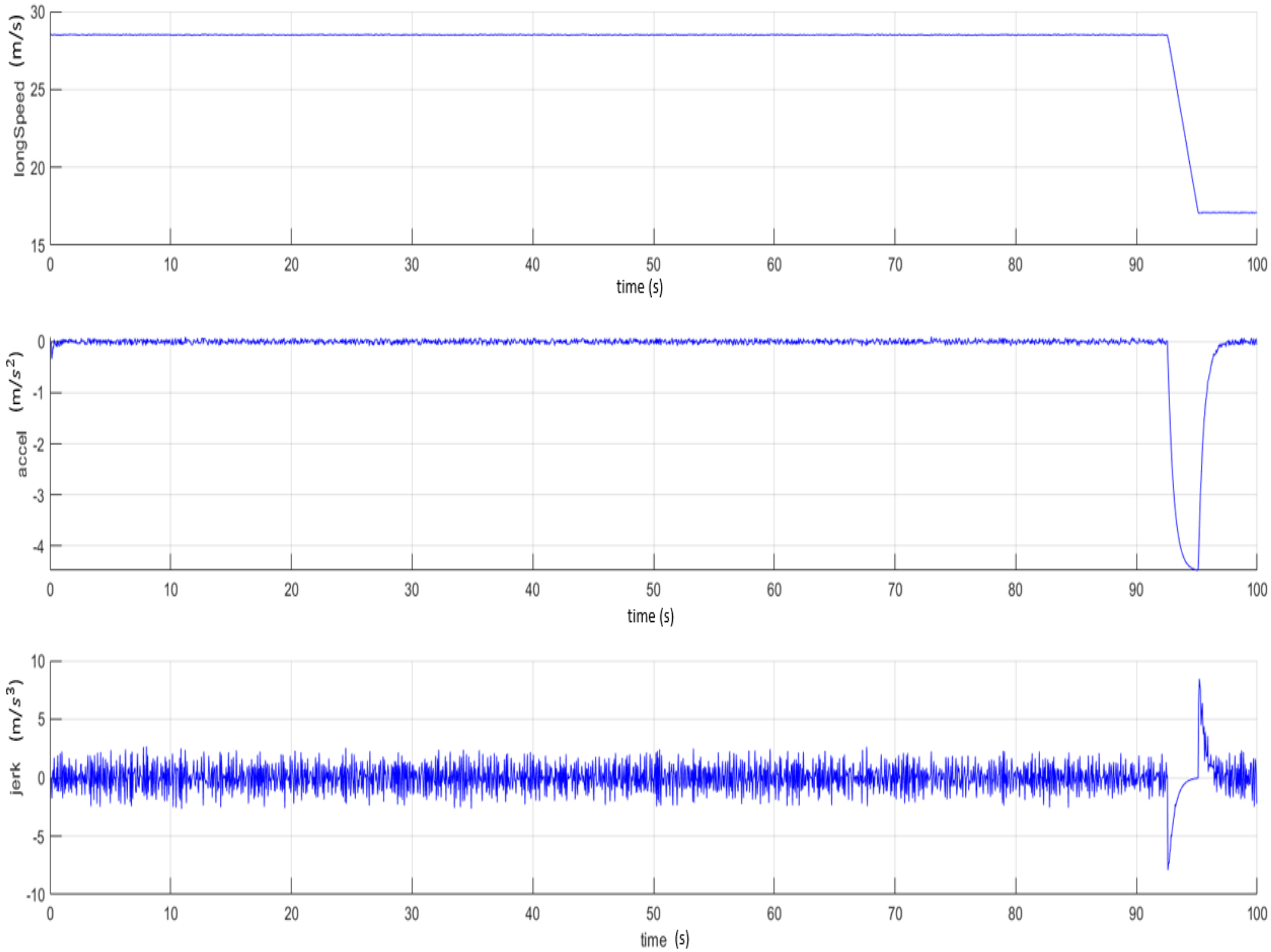


Figure 5.4: Speed Specification for safe braking trace

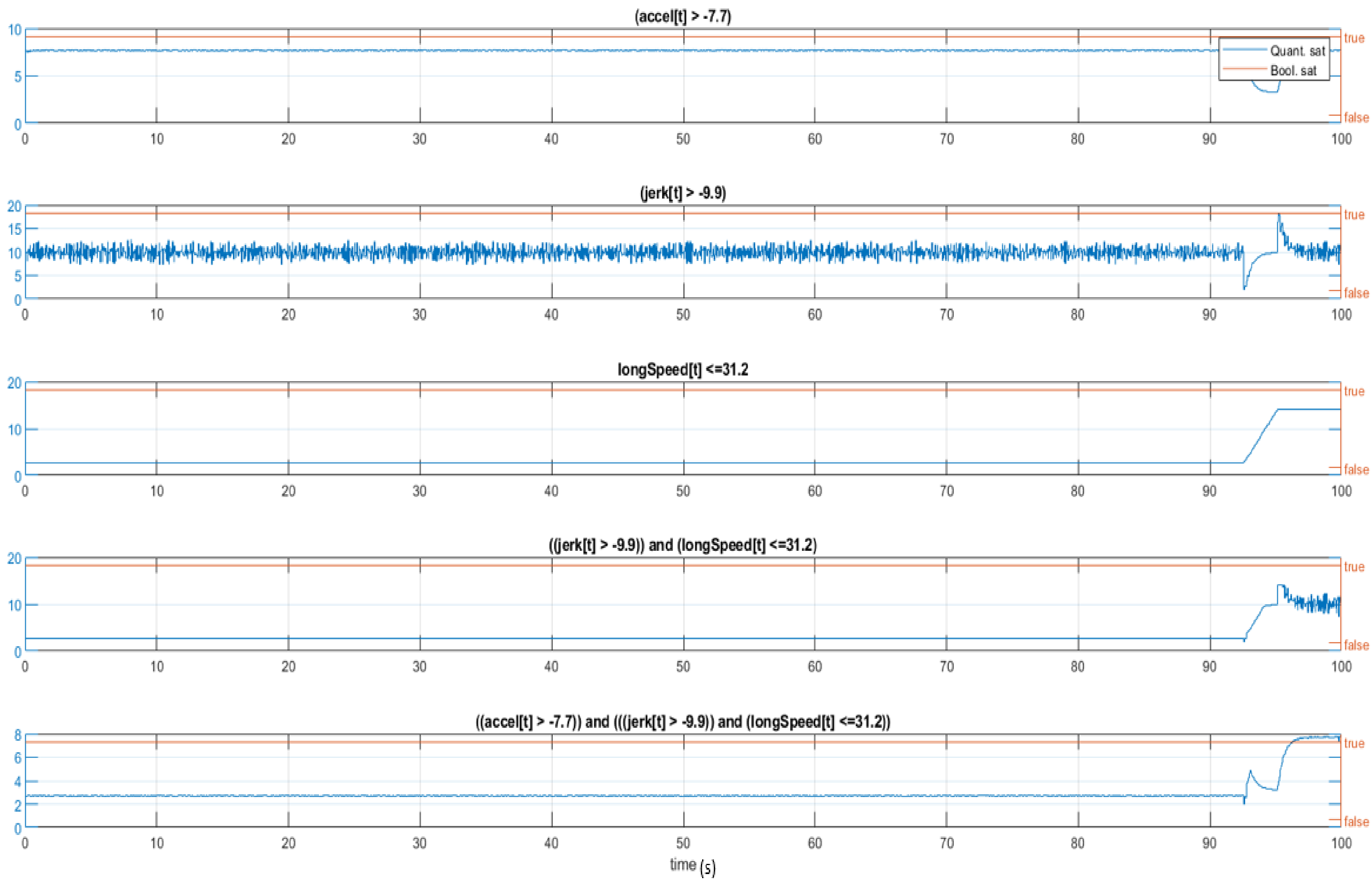



Figure 5.5: Speed Specification for safe braking output

It is noted here that this way of defining speed specification with a focus on comfort using the jerk indicator can be applied in special cases on the road segment such as off-ramps. Off-ramps usually have lower speed constraints than the main highway. Typically, vehicles on highways drive close to the highway speed limit, which would require the drivers to decelerate when approaching an off-ramp. During that time, it would be worth observing the behavior by which drivers brake in order to take the exit ramp. In this situation, it can be useful to also use our proposed speed specification 4.3 to study such behavior. We use the same vehicle trajectory studied in the previous section, but the difference here is that we add the condition that the vehicle is using an off-ramp. We observe how the jerk and acceleration variations in order to understand the underlying safety aspects. As seen in figure 5.6, the duration of time where the vehicle is on the off-ramp is indicated by the first sub-plot, while the longitudinal acceleration and jerk are in the second two sub-plots. After applying the STL specification, the output can be observed in figure 5.7. From this output, it is observed that the jerk value is greatly fluctuating during the last portion of the trace. However, our monitor registers these fluctuations as conforming behavior which indicates that during that time, this vehicle was braking safely.

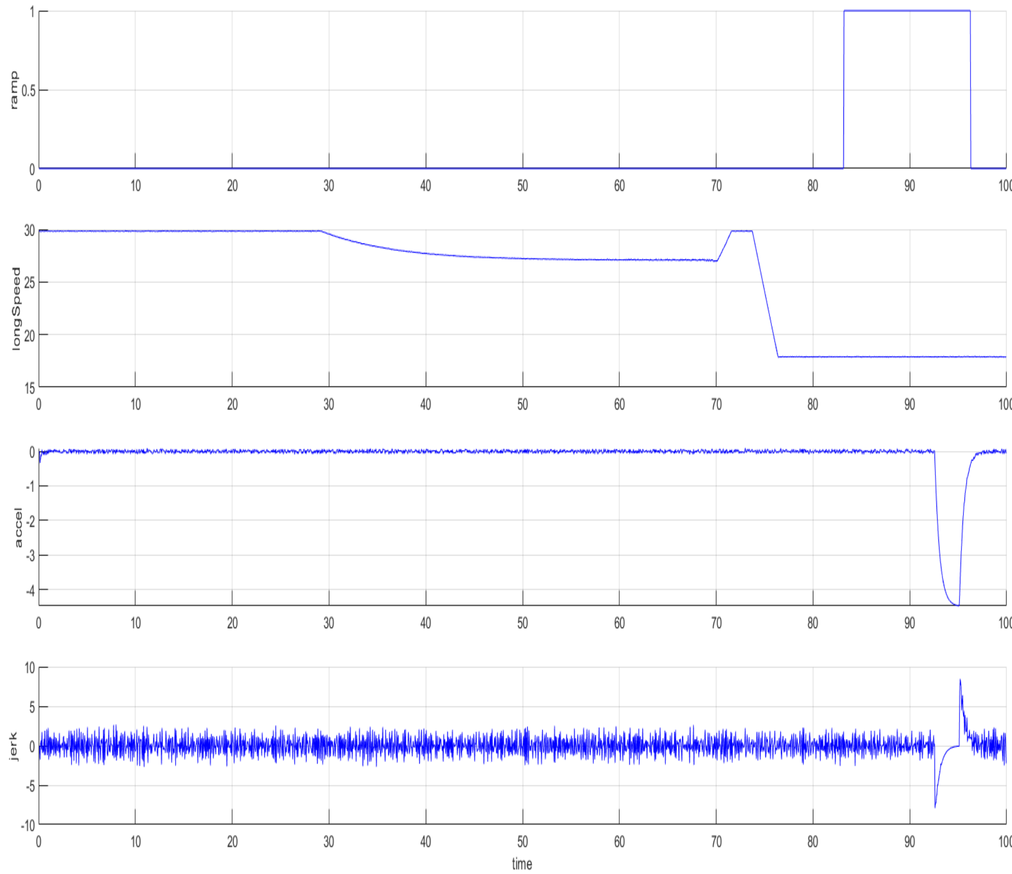



Figure 5.6: Speed Trace for vehicle using an offramp

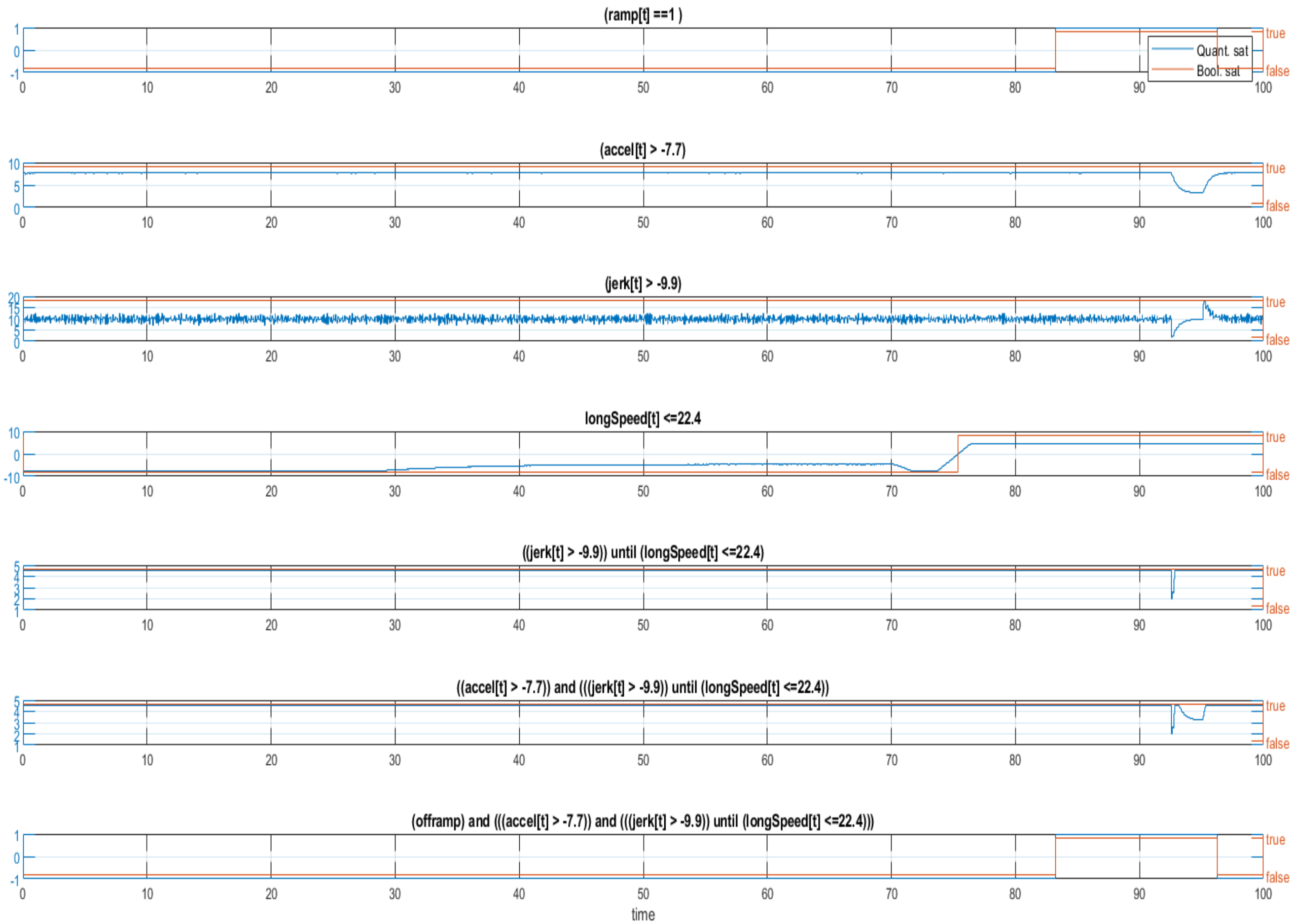



Figure 5.7: Speed Specification output for vehicle using off-ramp

To conclude, results from this section show that our speed monitor is able to capture changes in the speed, acceleration and jerk of individual vehicle trajectories. This output can be useful in understanding the safety aspects of microscopic traffic parameters. Using the speed as a safety indicator in our monitor shows when the vehicles are violating the imposed speed limits. While using indicators such as jerk help in understanding the underlying safety implications of vehicle braking behaviors.

### *5.1.2 Headway Specification Results*

In this section, we discuss the results of applying the headway specification 4.4 on the M50 highway scenario. We use vehicle headway information to test the output of our proposed monitor. It is important to note here that the trace contains some negative values, these mean that during that time, the vehicle didn't have a leader vehicle.

The headway trace is shown in figure 5.8, where it can be seen that there is a decrease in the headway at $85s$. We would like to observe whether the vehicle is closely following the lead vehicle for a prolonged period of time that would result in a conflict. However, the monitor output in figure 5.9 confirms that during that time, the vehicle recovers its spacing compared to its leader within the $2s$, hence the specification is satisfied. We can conclude that this vehicle is exhibiting safe car-following behavior.

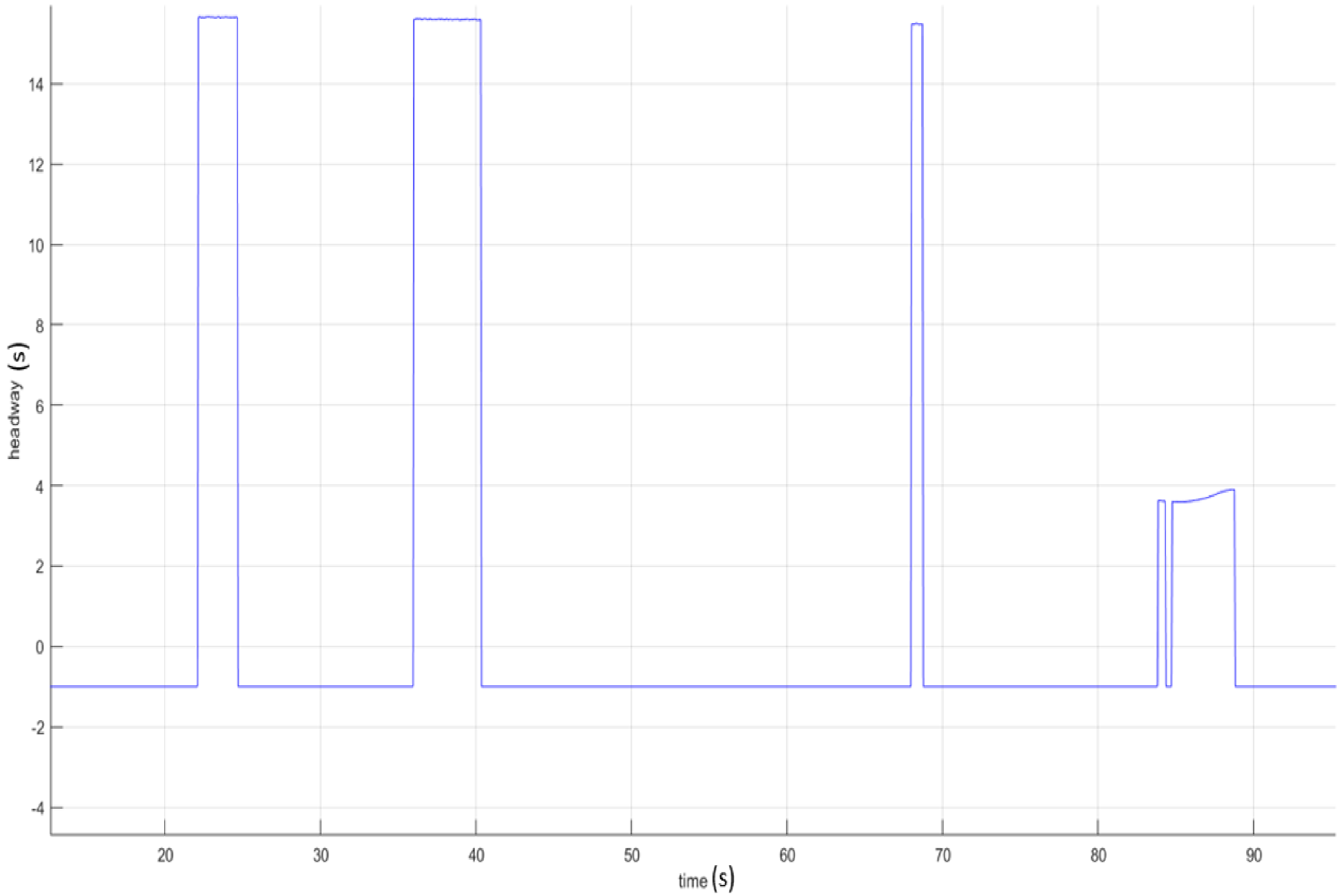


Figure 5.8: Headway Trace for vehicle

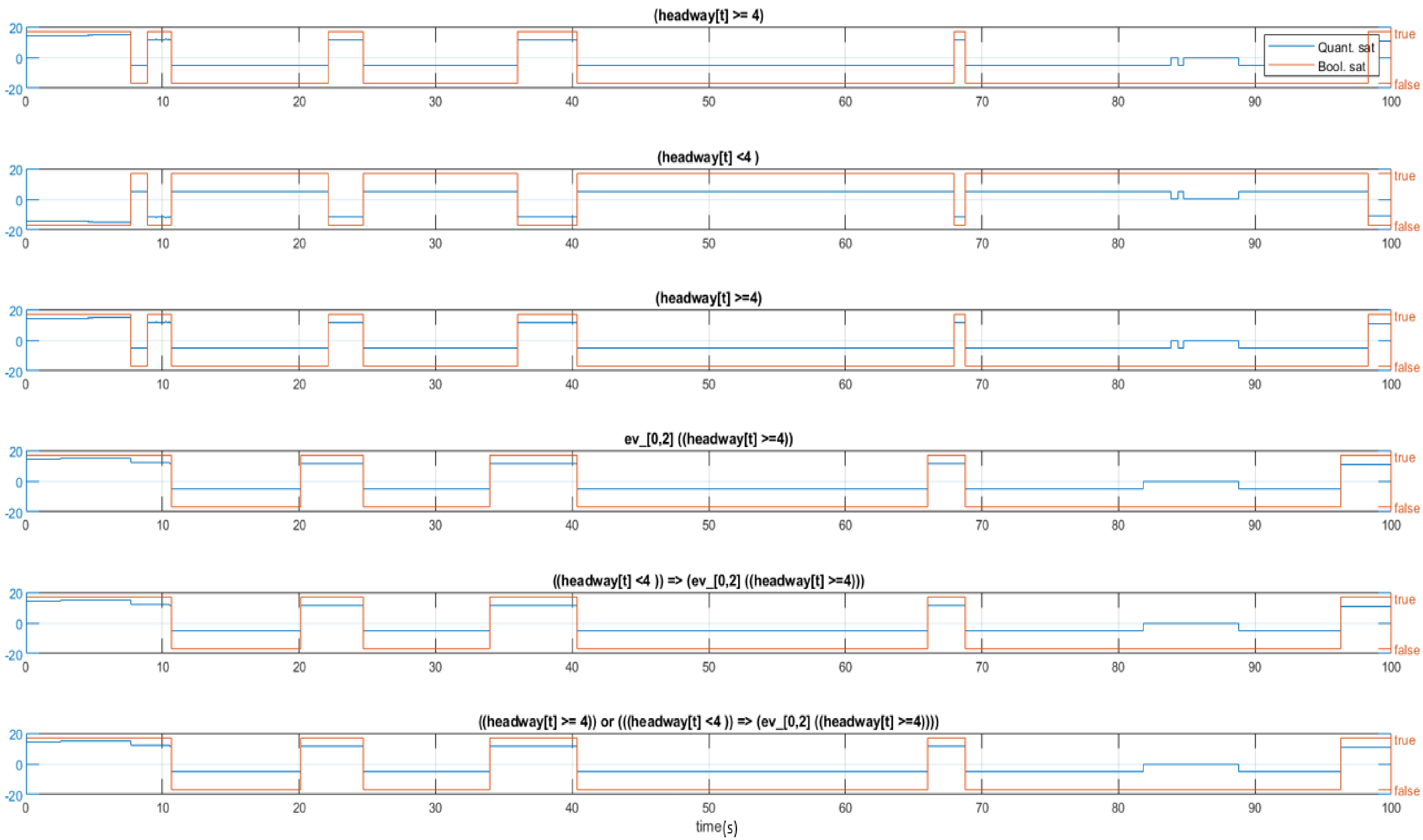



Figure 5.9: Headway Specification output

### 5.1.3 Lane Change Specifications Results

#### 5.1.3.1 Lane Change Detection

Based on the STL formula 4.6 defined in the previous chapter, the lateral position signal of a vehicle is used to test the monitor's ability to detect lane changes. Figure 5.10 shows the lateral position offset of a vehicle across the simulation time. As previously discussed, the lateral offset position of a vehicle can be used as an indicator for the onset of a lane-change maneuver. SUMO directly provides this information through TraCI functions so it is possible to extract the lateral dynamics of the vehicle. It can be noted here that there are multiple left and right lane change maneuvers taking place at multiple times during the simulation. The specification 4.6 is applied on the input trace and the corresponding satisfaction is plotted in figure 5.11. By studying the figure, it is found that the specification is satisfied during the times where there is a lane change detected and violated otherwise. Therefor, we are able to use this formula to further study the safety of lane change maneuvers in the following section.

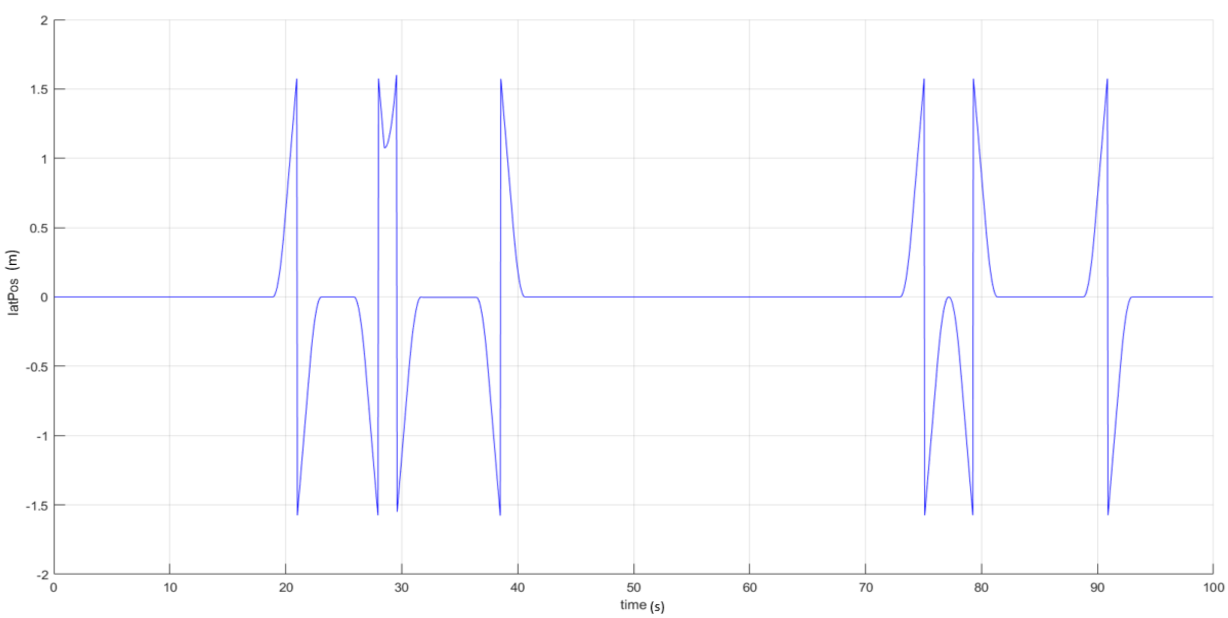



Figure 5.10: Lateral Position Offset

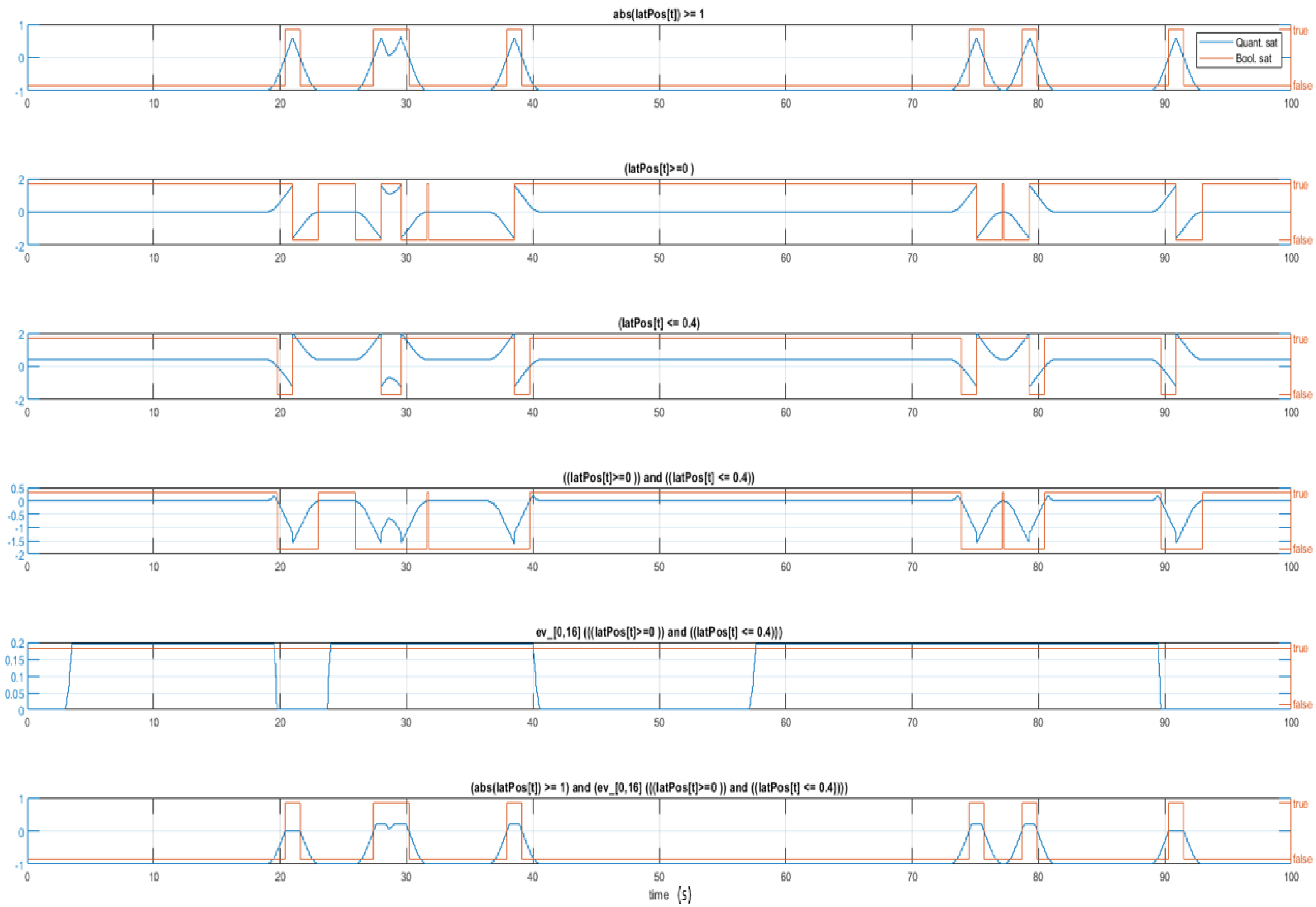



Figure 5.11: Lane Change Detection Specification output

#### *5.1.3.2 Lane Change Safety*

##### *5.1.3.2.1 Method 1: Gap Acceptance Behavior*

In figure 5.12, the input trace signals are shown. The first sub-figure depicts the lateral offset position, while the second sub-figure shows the total critical gap between the subject vehicle and any neighboring vehicles that interact with it during the simulation.

The formula 4.7 was applied to the trace. The output of the monitor is shown in figure 5.13. Since the subject vehicle is always keeping appropriate gaps between itself and the neighboring vehicles, it can be seen that the specification is satisfied throughout the simulation time.

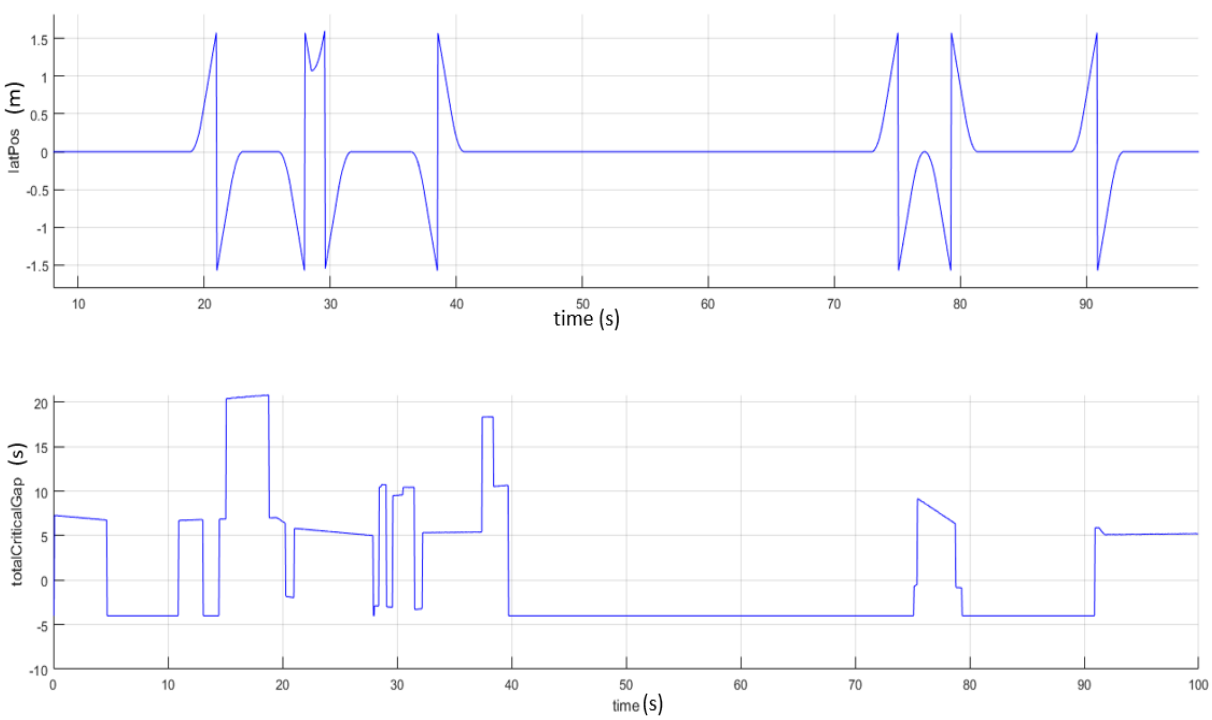



Figure 5.12: Critical Gap Trace

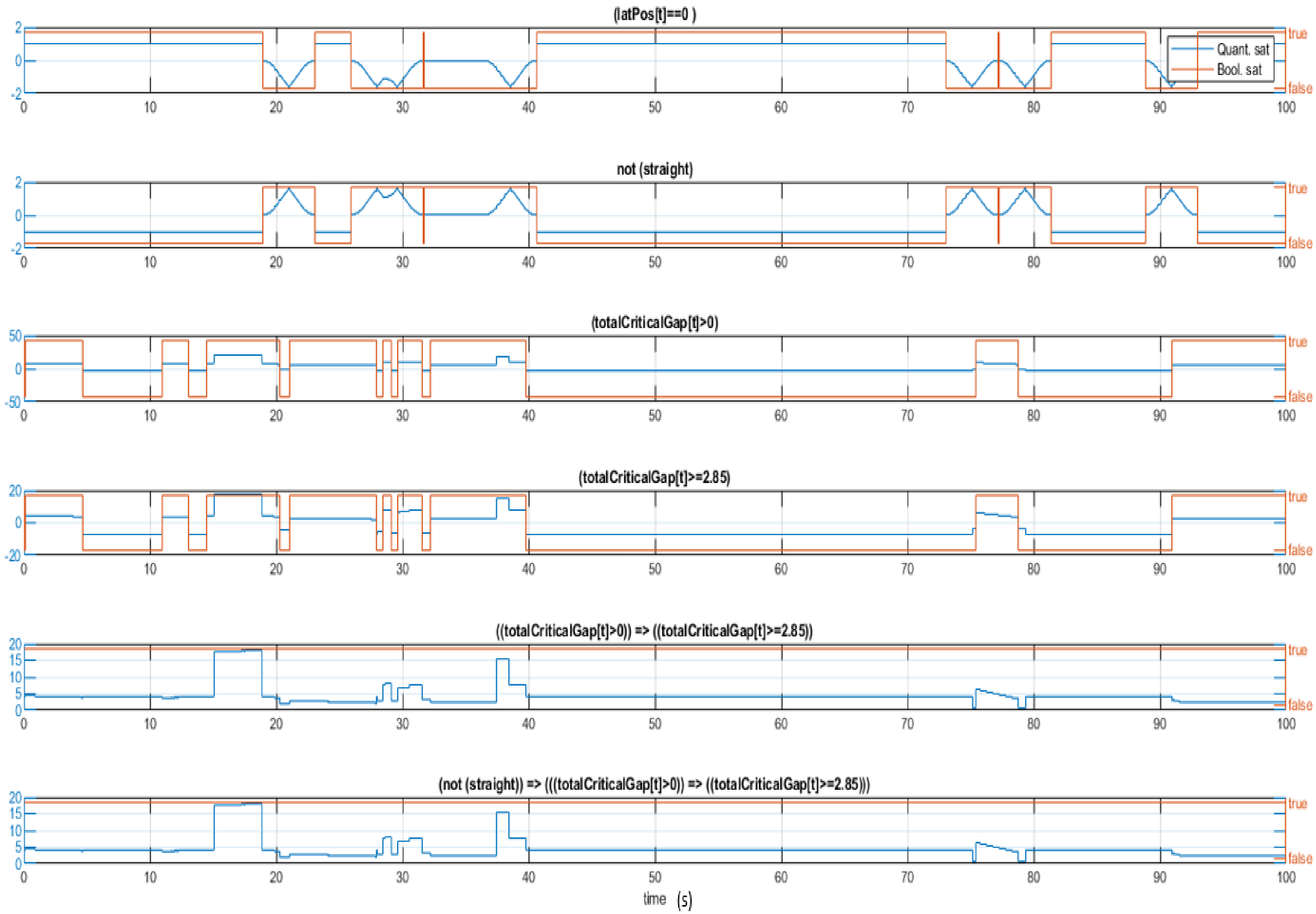



Figure 5.13: Critical Gap Specification output

##### *5.1.3.2.2 Method 2: Stopping Distance Index*

In order to calculate the SDI for the subject vehicle, information about the leading vehicle is needed. For the purposes of testing the proposed STL formula, TraCI functions were used in the SUMO simulation in order to obtain this information. These functions returns information about the neighboring vehicles to a subject vehicle, so the SDI for the lead vehicle was calculated. The formula 4.10 was used with the obtained signals that consisted of the calculated SDI for the subject vehicle. An example trace was used, which is shown in figure 5.14. The trace shows one lane change, and the values of the calculated SDI for the subject vehicle. The output from the monitor is shown in figure 5.15, where the specification is satisfied as the SDI profile of the vehicle is always above 0 during the lane-change maneuver.

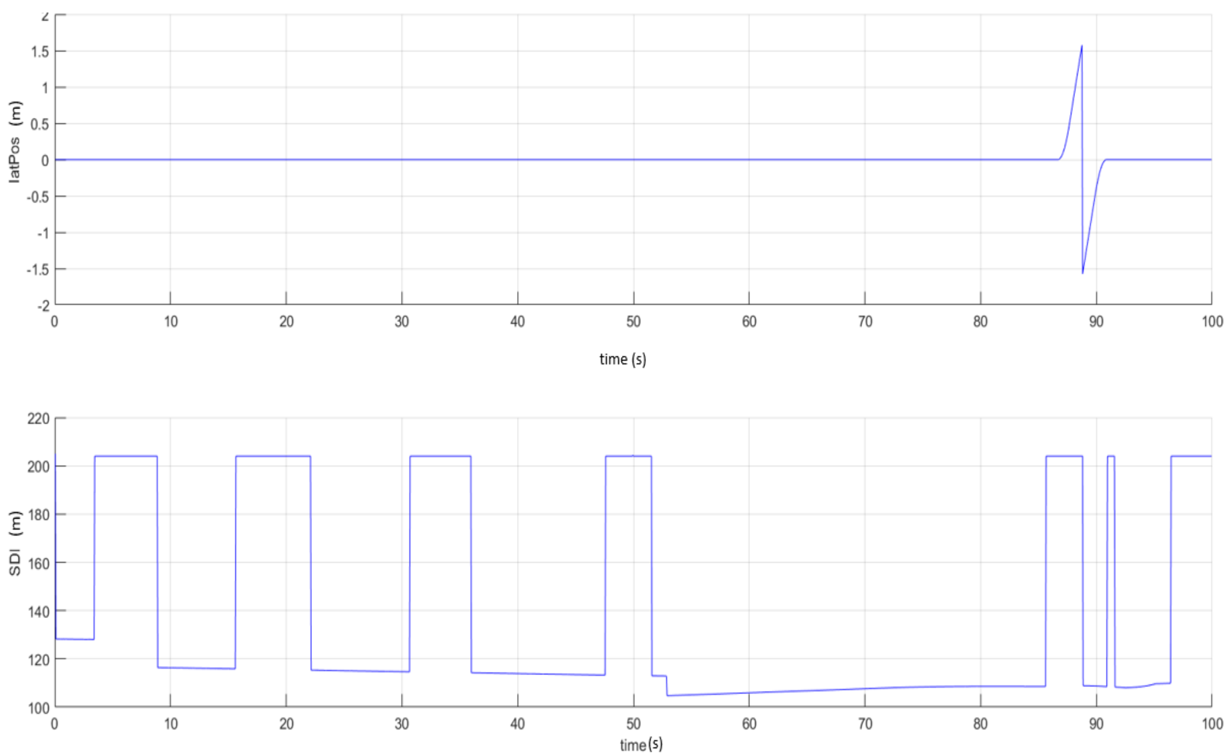



Figure 5.14: SDI Specification trace

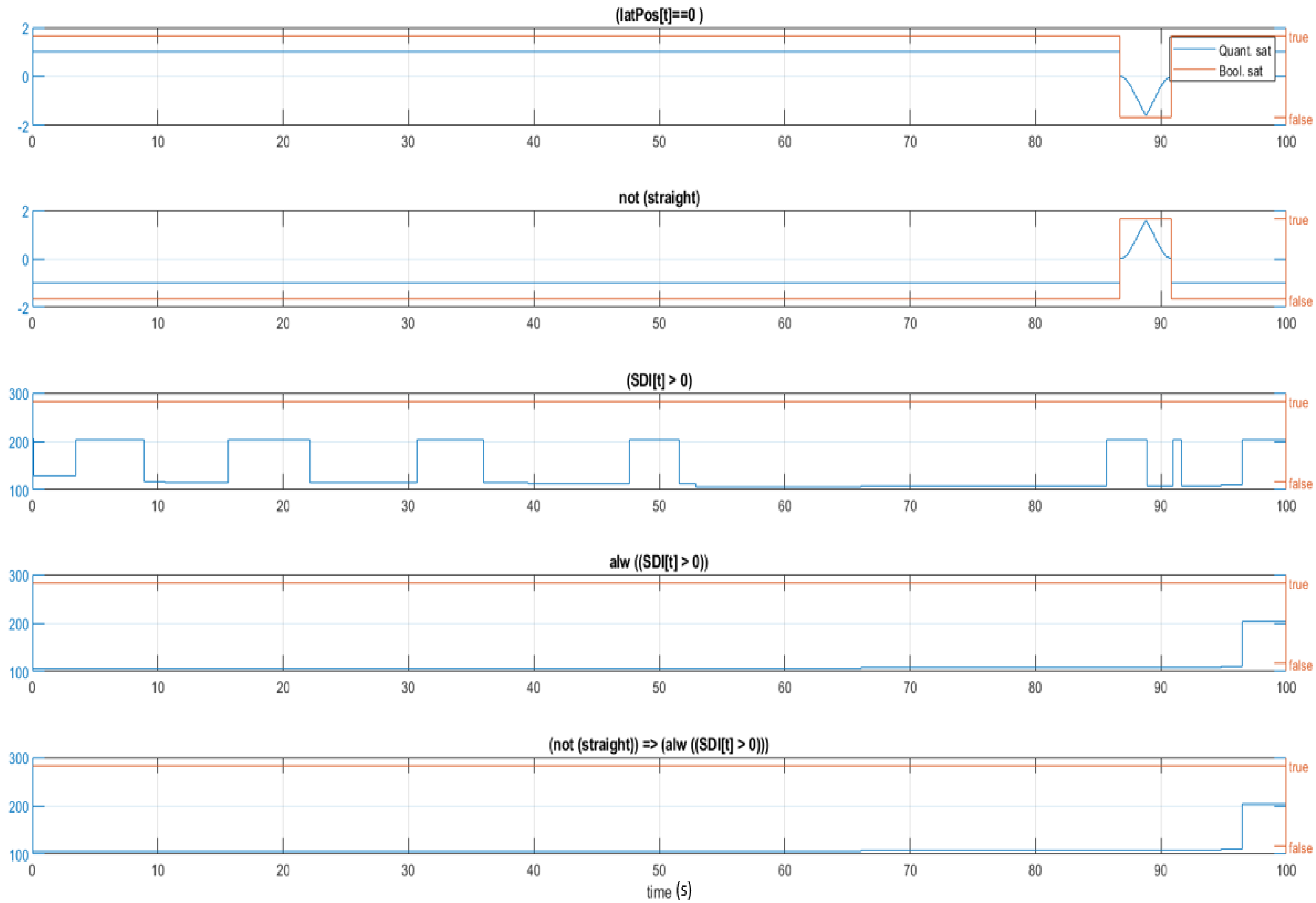


Figure 5.15: SDI Specification output

#### *5.1.3.2.3 Method 3: Post Encroachment Time*

In this section, we discuss the output of the lane change specification described in equation 4.11. The specification is applied to two vehicle trajectories; where one is the subject vehicle performing a lane change, while the other is its lag vehicle in the target lane. In figure 5.16, the lateral offset position of the subject vehicle is shown where it performs multiple lane-change maneuvers throughout the simulation. In order to retrieve this vehicle's neighboring vehicle trajectories, SUMO's built-in functions were used. Both the subject and follower vehicle x-positions are then plotted as seen in figure 5.17. Since it is known at what time the lane-change has started, it is then possible to extract the subject's vehicle location at that time. Using this information, the formula 4.11 is applied to the trajectories and the output can be seen in figure 5.18. The output shows that there exists a PET during the lane change time.

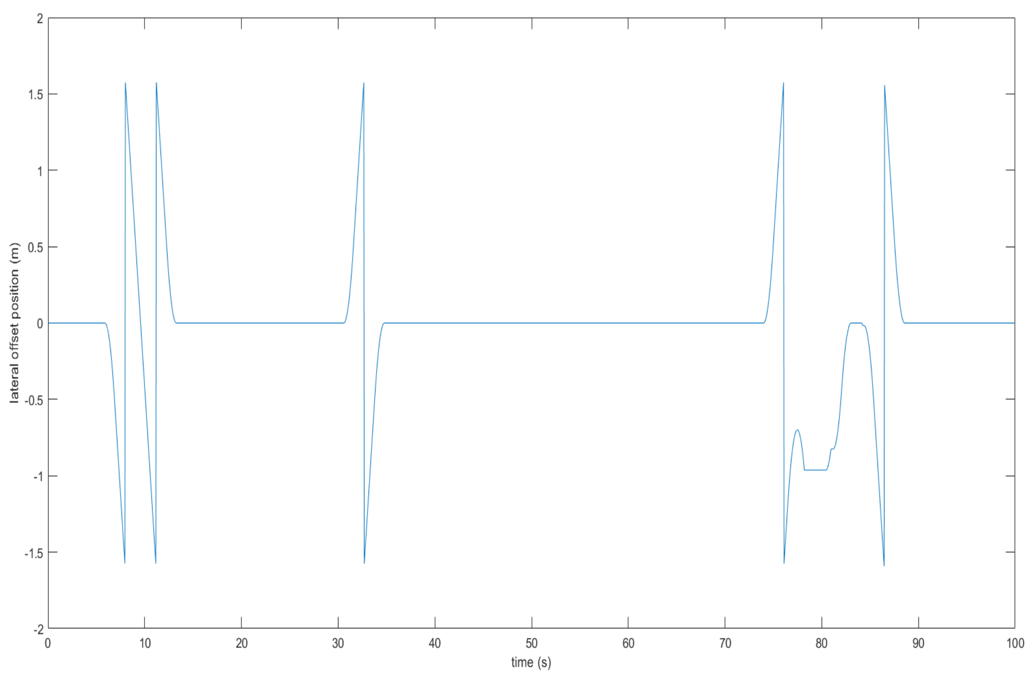



Figure 5.16: Subject Vehicle lateral offset position

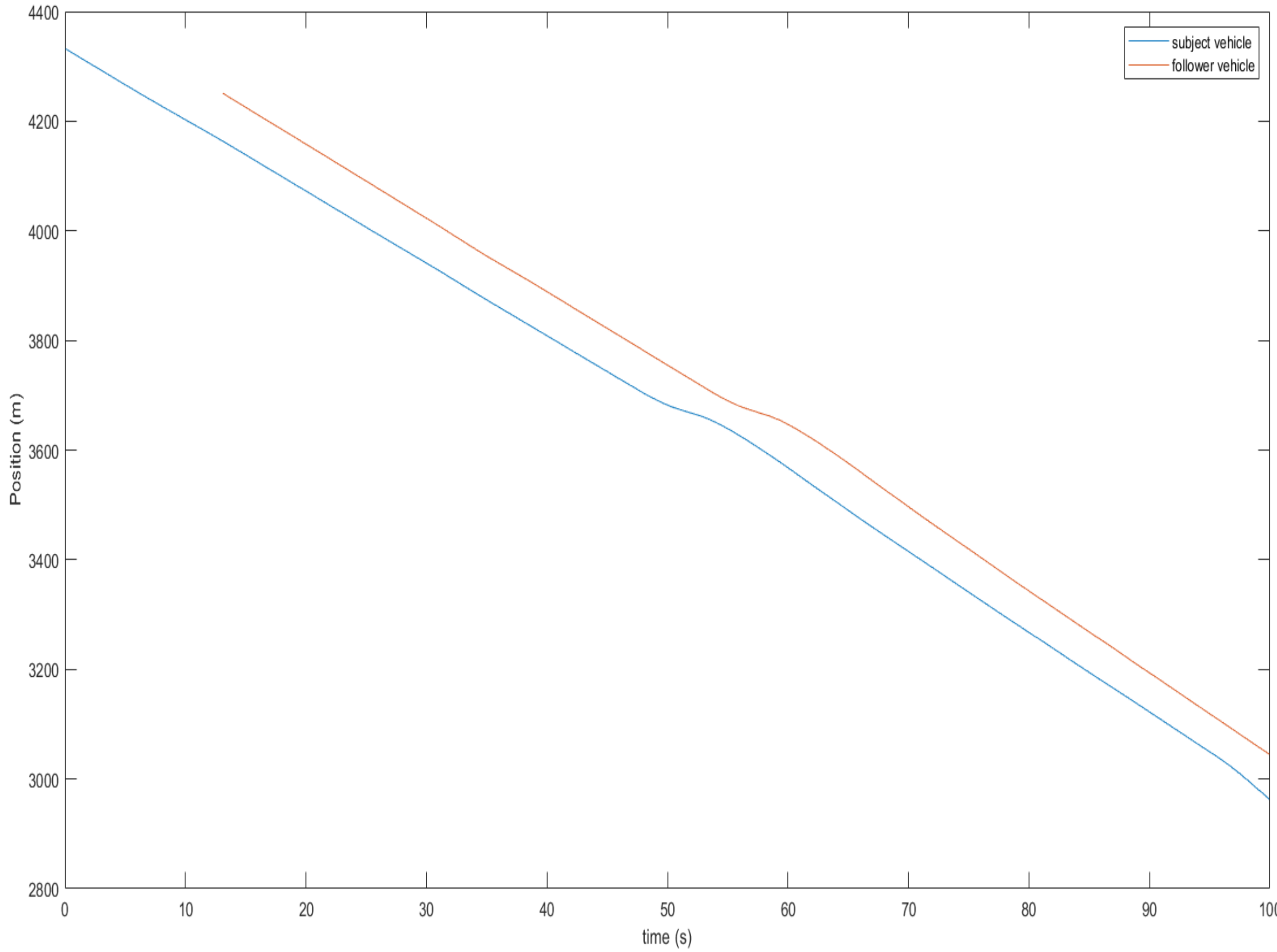



Figure 5.17: PET Specification trace

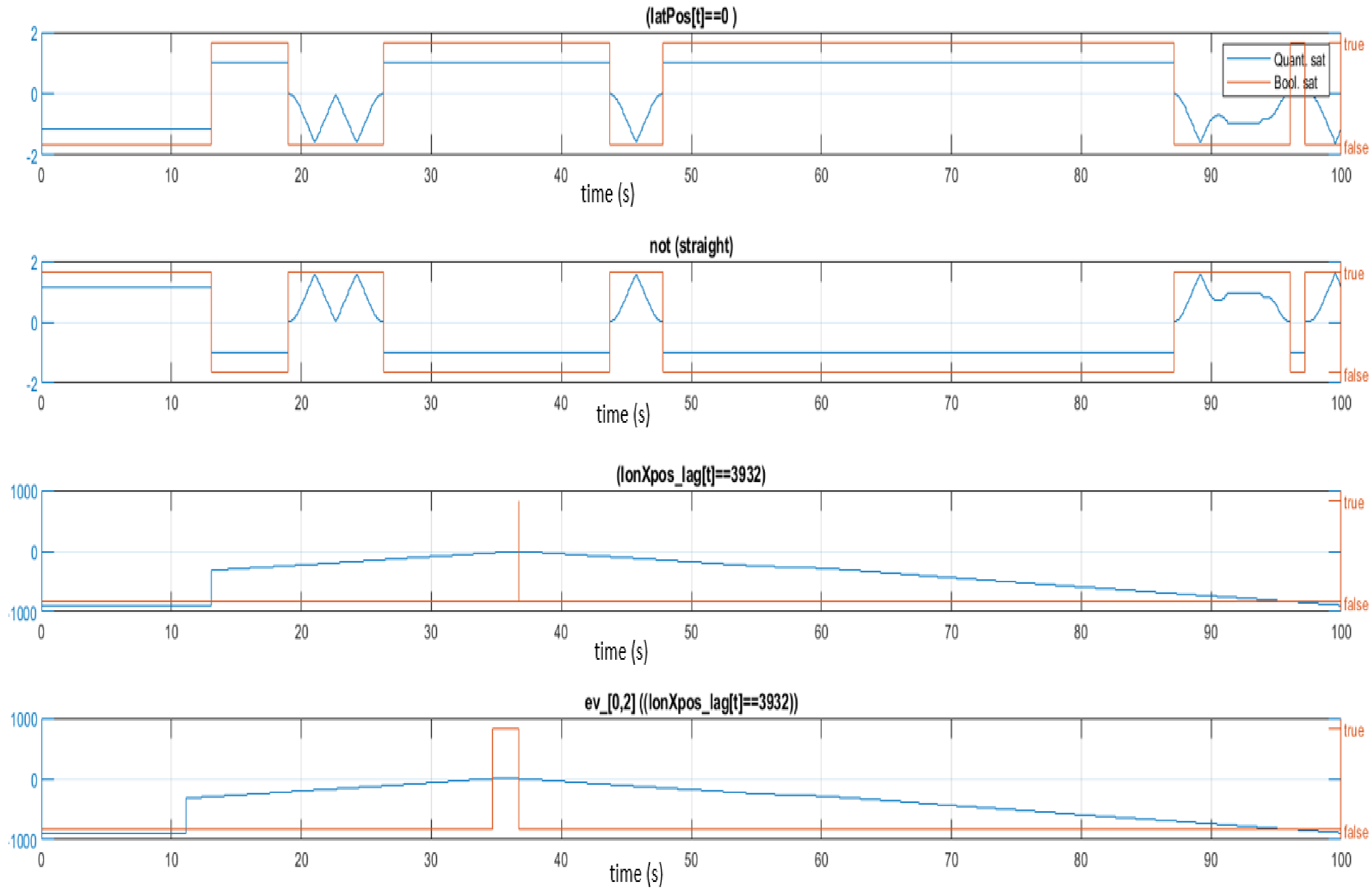



Figure 5.18: PET Specification output

To show case how the proposed monitors can be beneficial in understanding traffic characteristics, we run the simulation for $150s$ and apply the speed, headway and lane change safety specification to the individual vehicle trajectories. We use a minimum speed limit of $22.5m/s$ and a maximum speed limit of $31m/s$. Figure 5.19 shows the speed distribution of the scenario. The speed specifications that were introduced in equation (4.1) are used to generate these results, however we divide this specification into two scenarios; minimum speed limit and maximum speed limit.

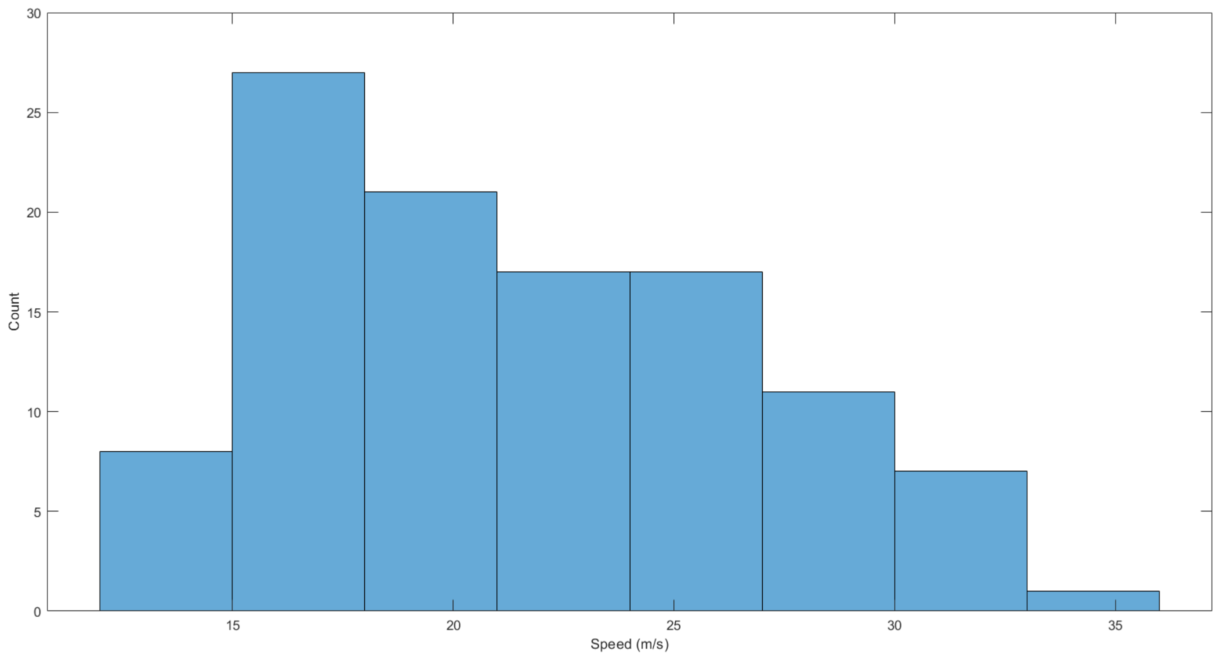



Figure 5.19: Speed distribution

We first start by the minimum speed limit specification where it produces the results as seen in table 5.1. It's expected that vehicles with speed lower than the minimum speed limit will violate the specification. The results show that the total number of violating vehicles is 99 with a mean speed of $19.27m/s$, which is lower than the threshold set in the simulation. On the other hand, the total number of conforming trajectories is 144 with a mean speed of $23.6m/s$.

| Measure | Conforming trajectories | Violating trajectories |
|---|---|---|
| Volume | 144 | 99 |
| Mean Speed (m/s) | 23.66 | 19.27 |
| Std Dev (Speed) | 4.17 | 4.1 |

Table 5.1: Conforming vs violating trajectories to the minimum speed specification

We run the same simulation to test the specification for the maximum speed limit. It's expected that vehicles with a speed higher than the maximum speed limit will violate the specification. As per results in table 5.2, it's found that there are a total of 9 violating trajectories with mean speed of $27.5m/s$, on the other hand, we find a total of 232 conforming trajectories with a mean speed of $21.63m/s$.

| Measure | Conforming trajectories | Violating trajectories |
|---|---|---|
| Volume | 232 | 9 |
| Mean Speed (m/s) | 21.63 | 27.5 |
| Std Dev (Speed) | 4.4 | 6.1 |

Table 5.2: Conforming vs violating trajectories to the maximum speed specification

In figure 5.20, the headway distribution is shown, which was defined using equation 4.4. The graph 5.3 shows the mean headway of 168 conforming trajectories to be $5.17s$, while the mean headway for the 58 violating trajectories is $3.71s$.

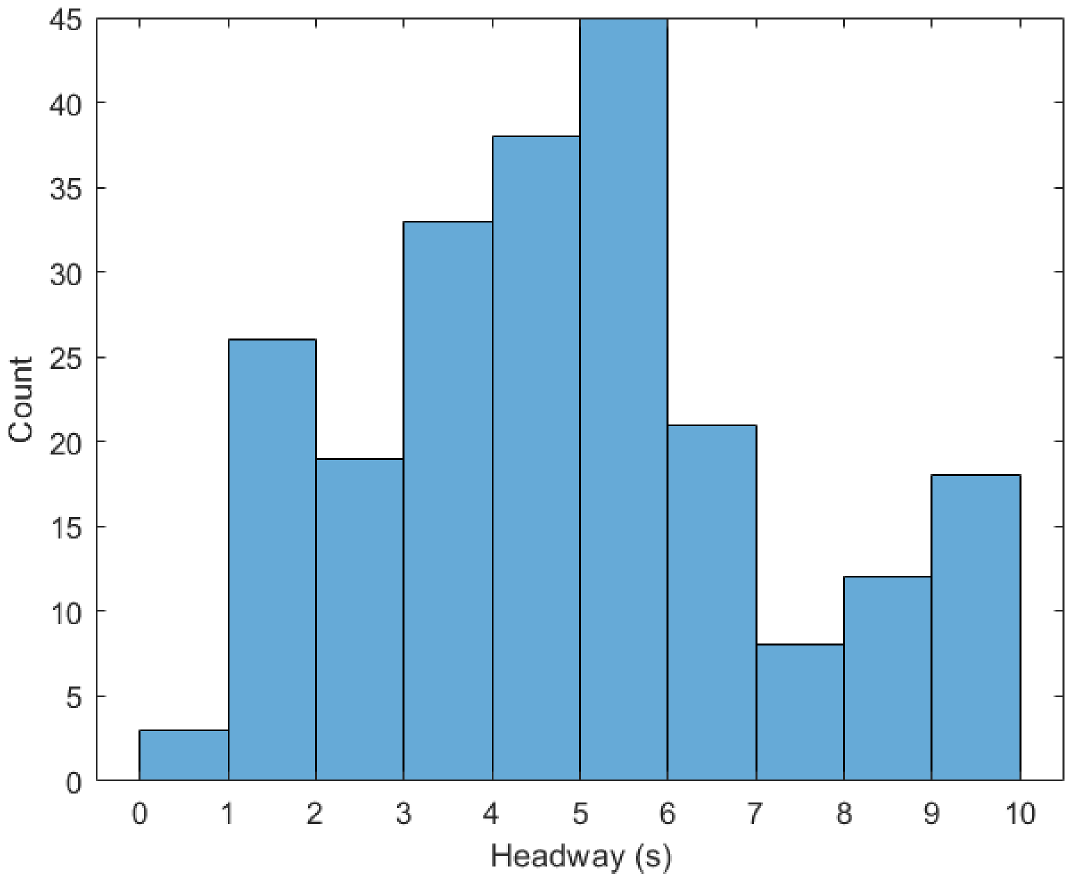


Figure 5.20: Headway distribution

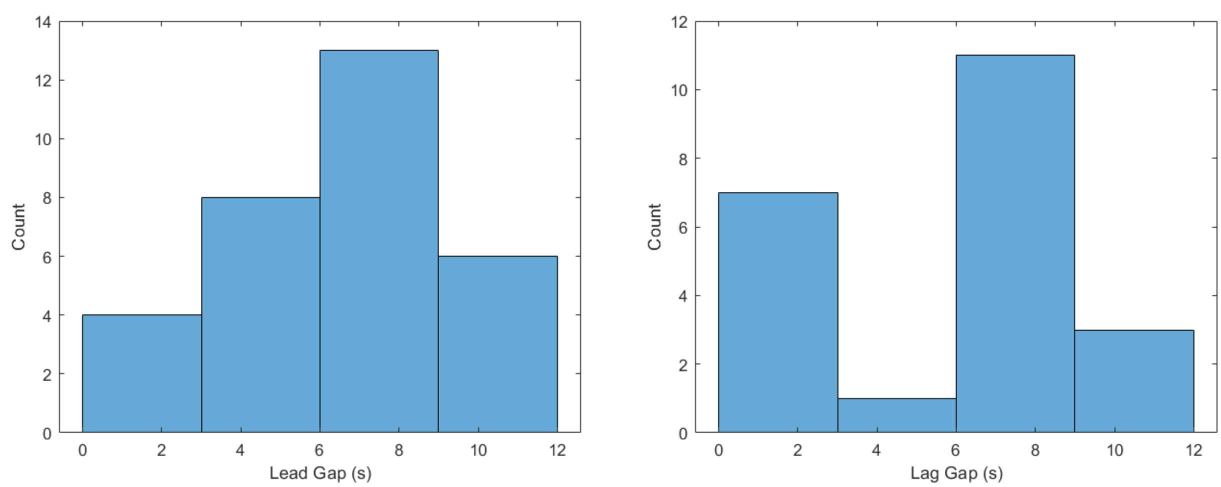
| Measure | Conforming trajectories | Violating trajectories |
|---|---|---|
| Volume | 168 | 58 |
| Mean Headway (s) | 5.17 | 3.71 |
| Std Dev (Headway) | 2.44 | 1.82 |

Table 5.3: Conforming vs violating trajectories to the headway specification

Finally, the lane change safety is tested using the specification defined in equation 4.7. In figure 5.21, both the distributions of the lead and lag gaps are plotted. Table 5.4 shows the mean and standard deviation values for both the lag and lead gaps. The mean values for the conforming trajectories are inline with what would be expected; both the lead and lag mean values are above the threshold we set in our specification. This can also be observed with the lead and lag mean values for the violating trajectories as they are well below the threshold.

| Measure | Conforming trajectories | Violating trajectories |
|---|---|---|
| Volume | 153 | 6 |
| Mean lead gap (s) | 7.40 | 1.88 |
| Std Dev (lead gap) | 1.9 | 0.6 |
| Mean lag gap (s) | 6.94 | 4.43 |
| Std Dev (lag gap) | 2.27 | 1.37 |

Table 5.4: Conforming vs violating trajectories to the lane-change safety gap specification




(a) Lead Gap distribution

(b) Lag Gap distribution

Figure 5.21: Gap Distribution

### 5.1.4 Proposed Communication Algorithm Results

For the implementation of the inter-vehicle communication algorithm proposed, we use the VEINs framework [40]. Veins is an open source simulator for VANETs. It couples both a network simulator (Omnet++) and an urban traffic network simulator (SUMO) through the TraCI protocol. SUMO provides the traffic network, the vehicle types, number and behaviour through car following models, while Omnet++ provides connectivity between the nodes which are the vehicles in this case. Veins is an Omnet++ project that has an implementation of all lower layers in a VANET network, which allows users to focus on development of data dissemination applications. It contains an implementation of TraCI protocol which allows the communication to SUMO and extract information from it while the simulation is running. TraCI protocol allows the movement of nodes in SUMO to be reflected in Omnet++ and vice versa. The simulation starts from running an Omnet++ project that uses Veins, where Veins initiates a client connection to SUMO that acts as the server. Each created vehicle is considered as a communicating node in the network as well as have the mobility defined by SUMO. Information goes back and forth between SUMO and Omnet++ through a TCP socket, making the simulation run simultaneously between both tools.

In order to validate that our proposed communication algorithm in fact positively impacts the safety by adjusting intra-vehicle spacing, we run two different simulations. The first case is without the application of the actuated acceleration, in this scenario, the vehicles are using the default car-following model implemented in SUMO. The second scenario uses the proposed algorithm to exchange the safety-related information between the vehicles, hence, making it possible for follower vehicles to actuate their acceleration according to the propsoed equation in 4.13. The simulation is run for $100s$, and a total of 110 vehicles are generated. Each vehicle has a minimum power level of $-90dBm$, transmission power of $15mW$, and a transmission range of $500m$.

Next, the headway traces for the two scenarios were extracted and analyzed using our headway monitor defined earlier using equation 4.4. For the first scenario, a headway trace is shown in figure 5.22, by observing the trace, it can be noticed that from around $20s$ up until $95s$ the vehicle's headway trace is below the defined threshold of $4s$. The monitor is then applied to analyze the headway trace. As seen in figure 5.23, the monitor output confirms the initial observation as it returns a false value for that same duration from $20s$ up until $95s$.

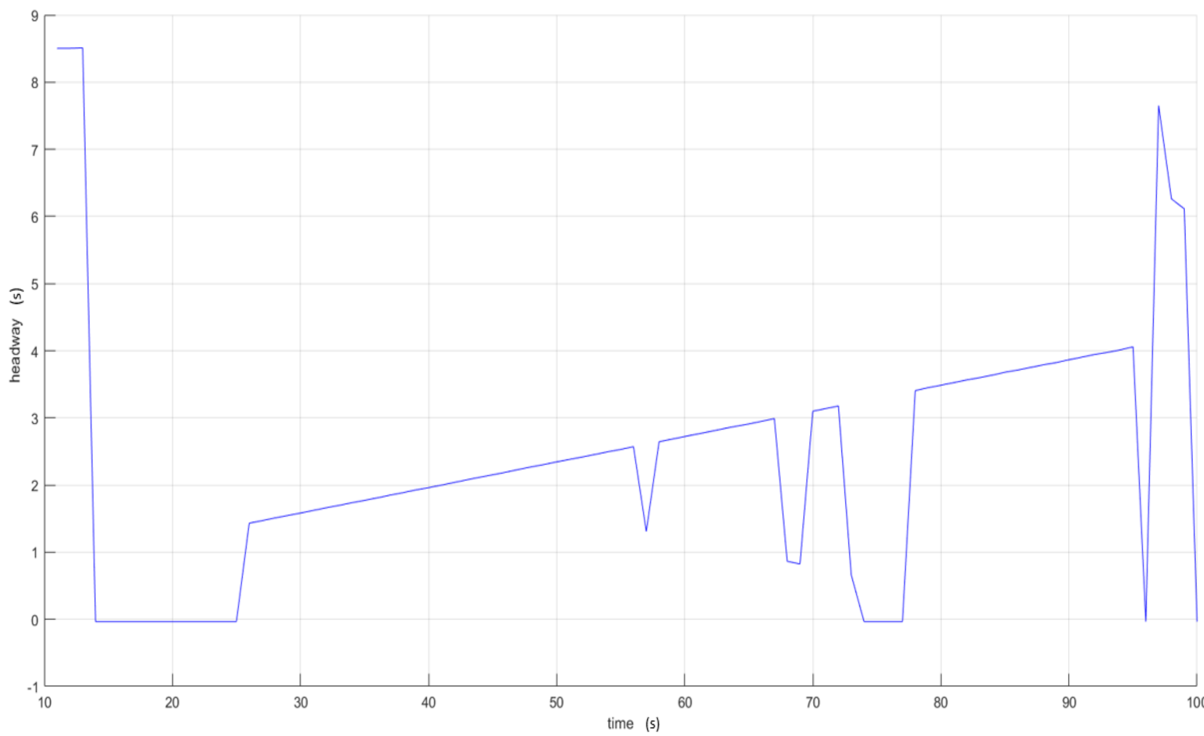


Figure 5.22: Headway Trace with no IVC

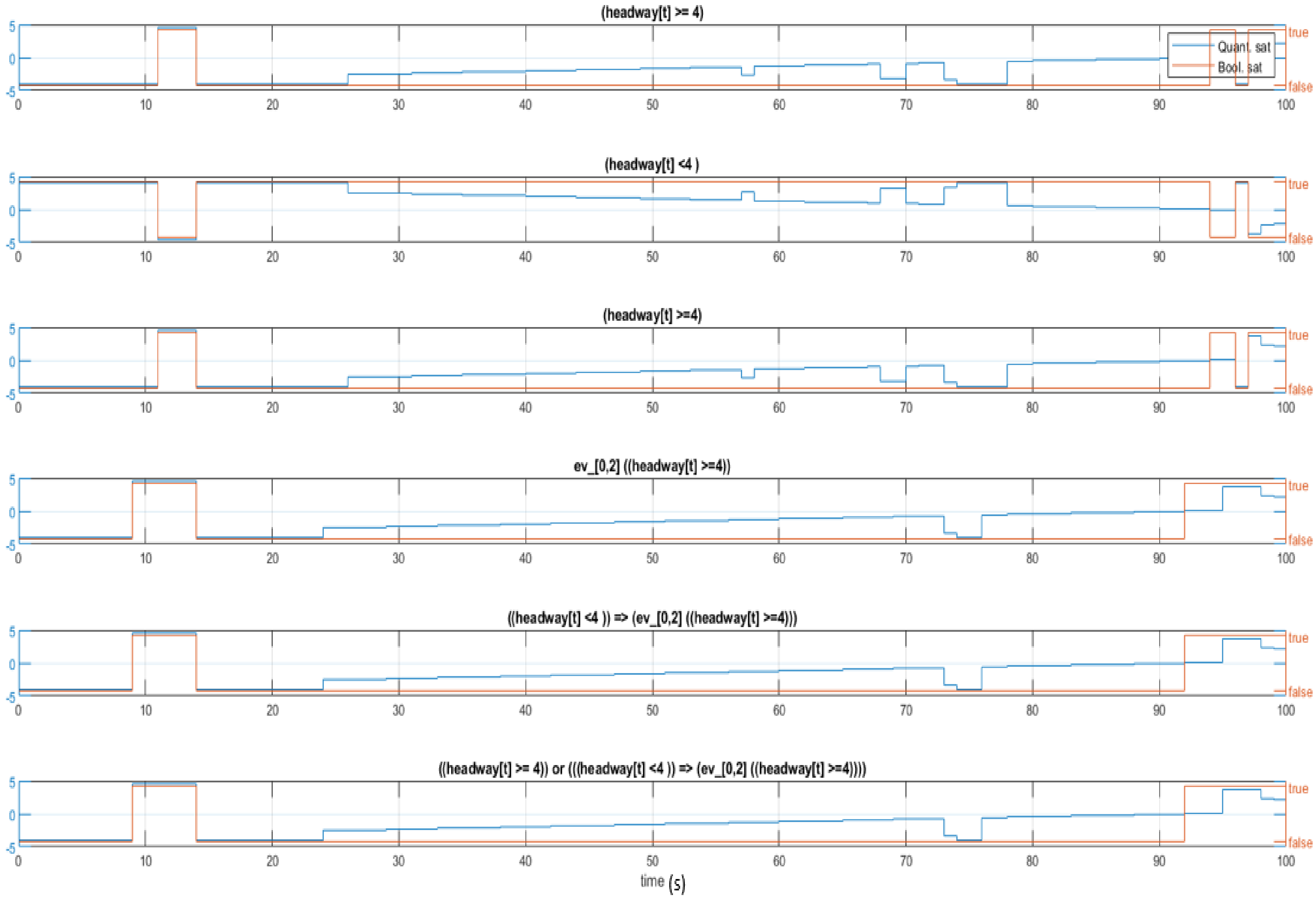



Figure 5.23: Headway Specification output with no IVC

In the second scenario, we apply the communication algorithm to the simulation where the follower vehicles adjust their acceleration based on the disseminated information in an online fashion. By inspecting the headway traces from the second scenario, we find that the follower vehicles were complying to the headway specification. Figure 5.24 shows the headway trace for a vehicle, and the output trace 5.25 shows that the trace is indeed conforming to the headway specification.

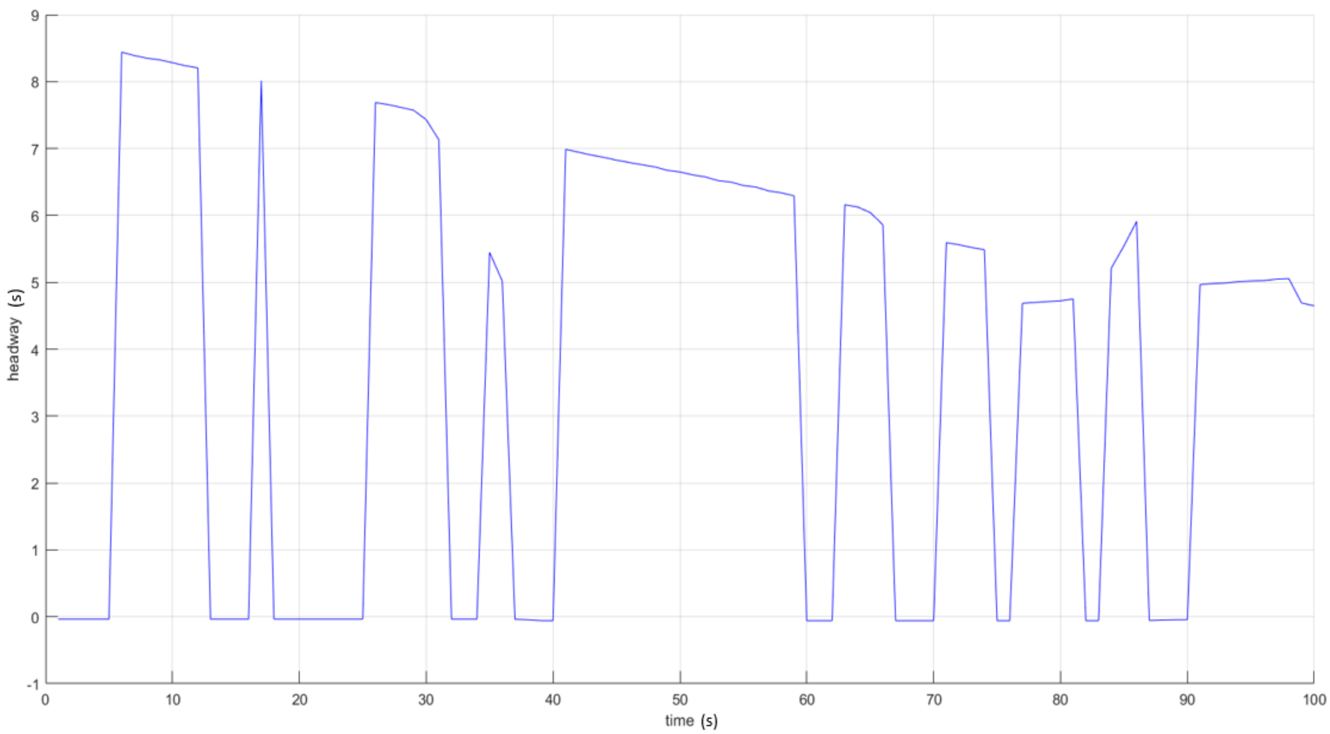



Figure 5.24: Headway Trace with IVC

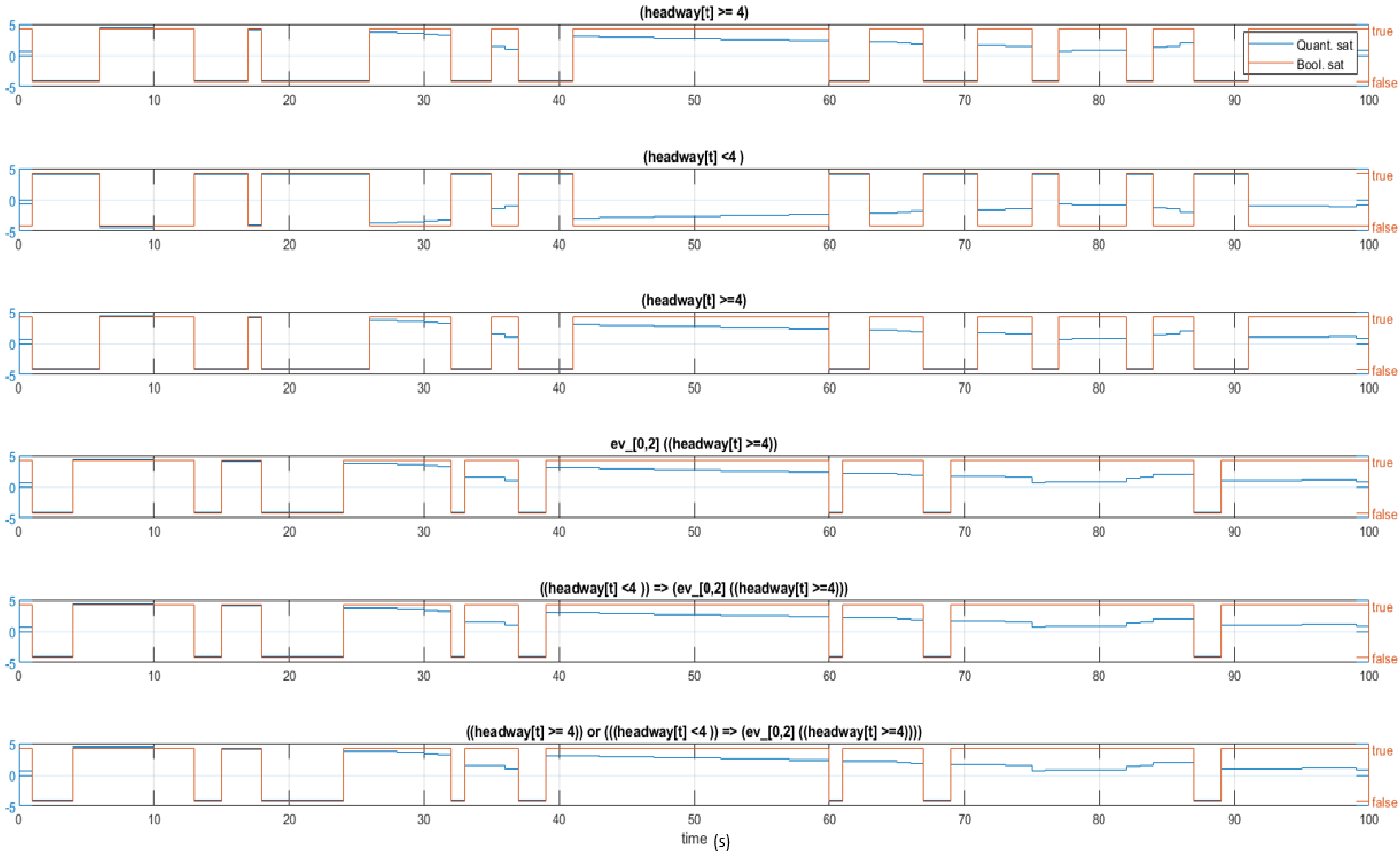



Figure 5.25: Headway Specification output with IVC

To conclude, the results of this work showed that the specification-based monitoring using STL language is able to capture various traffic stream properties. We first started by showing the speed-related specifications and how we can capture various safety aspects such as speeding, safe braking and safe exiting of a highway. The results also showed that we are able to identify safe car-following behavior based on the time headway between individual vehicles. Then, we studied the detection of lane change maneuvers as well as monitored their safety using various indicators. Finally, we implemented a vehicular-communication based scenario where vehicles adjust their acceleration based on information received from leader vehicles. This enabled us to showcase how the developed STL monitors can be used in practice to identify the safe headway before and after applying the communication.

# CHAPTER 6: CONCLUSION

In this thesis, we proposed a new method for monitoring the safety of traffic behavior using formal methods. In our approach, we used vehicle trajectories extracted from calibrated micro-simulation models and studied their conformity to safe behavior by applying monitors that use formally defined specifications. Signal Temporal Logic (STL) was chosen as the formal language for our approach. The use of formal methods in traffic allows for the high-level reasoning about the traffic state. Formal methods have the ability to precisely define logical rules that would guarantee the correctness of the traces it's being applied to.

We also implemented a communication algorithm for the online adjustment of individual vehicle acceleration to observe the impact vehicle-to-vehicle communication can have on safety enhancement. We used the formal language STL to define the different traffic-related specifications and employed a framework that consists of multiple tools for the implementation and simulation of our work. The framework consists of a traffic micro-simulator (SUMO), a tool that defines and analyzes vehicle trajectories using formal specification-based monitoring (Breach toolbox), a communication network simulator (OMNET++) and an application that allowed for the integration between them (VEINs). A highway scenario was simulated and the extracted vehicle trajectories were used to test the proposed specifications.

The results showed that our approach can successfully capture different safety attributes of the traffic flow. These include vehicle trajectories conformity to speed limits, safe braking, safe car following behavior, lane-change detection and safety. It was also shown that the proposed vehicular communication algorithm successfully disseminated information to vehicles and they were able to adjust their acceleration in such a way to conform to safe car-following behavior.

One of the limitations of this work is that the proposed specifications are able to only capture temporal aspects of the traffic behavior. This can be remedied by using a more expressive specification language that would incorporate spatial aspects as well. These operators would allow for describing spatial components; such as defining certain regions of interests in the traffic network. Another limitation is that we tested our vehicular communication algorithm using only the headway specification as it was the relevant one. But other aspects of the traffic stream can also be studied for that scenario.

As for future work, this work can be enhanced upon to include more surrogate safety measures such as time-to-collision (TTC) to study the safety of road segments. The addition of macroscopic traffic parameters monitors can provide valuable insights for traffic network safety. Other road geometries can also be used to test the synthesized traffic monitors under different conditions. More case studies with a focus on Florida traffic environment can also be used for the application of the developed monitors. Finally, online monitoring is the logical next step for the synthesis of real-time monitors. These monitors will be able to detect the specified traffic behavior and their safety in real-time. This can then be integrated within a self-contained solution that observes, analyzes and alerts decision-makers of safety hazards on the road.

# LIST OF REFERENCES


[1] A. M. de Souza, C. A. Brennand, R. S. Yokoyama, E. A. Donato, E. R. Madeira, and L. A. Villas, "Traffic management systems: A classification, review, challenges, and future perspectives," *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, p. 1550147716683612, Apr 2017.

[2] N. K. Jain, R. K. Saini, and P. Mittal, "A review on traffic monitoring system techniques," in *Soft Computing: Theories and Applications*, ser. Advances in Intelligent Systems and Computing, K. Ray, T. K. Sharma, S. Rawat, R. K. Saini, and A. Bandyopadhyay, Eds. Springer, 2019, p. 569–577.

[3] C. Wang and N. Stamatiadis, "Surrogate safety measure for simulation-based conflict study," *Transportation Research Record*, vol. 2386, no. 11, p. 72–80, Jan 2013.

[4] S. Mitsch, S. M. Loos, and A. Platzer, "Towards formal verification of freeway traffic control," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Apr 2012, p. 171–180.

[5] A. Platzer, "Verification of cyberphysical transportation systems," *IEEE Intelligent Systems*, vol. 24, no. 4, p. 10–13, Jul 2009.

[6] A. Fantechi, F. Flammini, and S. Gnesi, "Formal methods for intelligent transportation systems," in *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies*, ser. Lecture Notes in Computer Science, T. Margaria and B. Steffen, Eds. Springer, 2012, p. 187–189.

[7] M. Ma, J. A. Stankovic, and L. Feng, "Runtime monitoring of safety and performance requirements in smart cities," in *Proceedings of the 1st ACM Workshop on the Internet*

*of Safe Things - SafeThings'17*. ACM Press, 2017, p. 44–50. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3137003.3137005

[8] S. Coogan, M. Arcak, and C. Belta, "Formal methods for control of traffic flow: Automated control synthesis from finite-state transition models," *IEEE Control Systems Magazine*, vol. 37, no. 2, p. 109–128, Apr 2017.

[9] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv:1708.06374 [cs, stat]*, Oct 2018, arXiv: 1708.06374. [Online]. Available: http://arxiv.org/abs/1708.06374

[10] M. H. ter Beek, S. Gnesi, and A. Knapp, "Formal methods for transport systems," *International Journal on Software Tools for Technology Transfer*, vol. 20, no. 3, p. 237–241, Jun 2018.

[11] J. Mao and L. Chen, "Runtime monitoring for cyber-physical systems: A case study of cooperative adaptive cruise control," in *2012 Second International Conference on Intelligent System Design and Engineering Application*, Jan 2012, p. 509–515.

[12] C. Pek, P. Zahn, and M. Althoff, "Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2017, p. 1477–1483.

[13] M. Althoff, O. Stursberg, and M. Buss, "Safety assessment of autonomous cars using verification techniques," in *2007 American Control Conference*, Jul 2007, p. 4154–4159.

[14] S. Coogan and M. Arcak, "Freeway traffic control from linear temporal logic specifications," in *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, Apr 2014, p. 36–47.

[15] E. S. Kim, M. Arcak, and S. A. Seshia, “Compositional controller synthesis for vehicular traffic networks,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, p. 6165–6171.

[16] A. Müller, S. Mitsch, and A. Platzer, “Verified traffic networks: Component-based verification of cyber-physical flow systems,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep 2015, p. 757–764.

[17] S. Coogan, E. A. Gol, M. Arcak, and C. Belta, “Traffic network control from temporal logic specifications,” *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, p. 162–172, Jun 2016.

[18] ——, “Controlling a network of signalized intersections from temporal logical specifications,” in *2015 American Control Conference (ACC)*, Jul 2015, p. 3919–3924.

[19] N. Mehr, D. Sadigh, and R. Horowitz, “Probabilistic controller synthesis for freeway traffic networks,” in *2016 American Control Conference (ACC)*, Jul 2016, p. 880–880.

[20] A. Rashid, M. Umair, O. Hasan, and M. H. Zaki, “Toward the formalization of macroscopic models of traffic flow using higher-order-logic theorem proving,” *IEEE Access*, vol. 8, p. 27291–27307, 2020.

[21] S. Sadraddini, J. Rudan, and C. Belta, “Formal synthesis of distributed optimal traffic control policies,” ser. ICCPS ’17, Apr 2017, p. 15–24. [Online]. Available: https://doi.org/10.1145/3055004.3055011

[22] M. Ma, E. Bartocci, E. Lifland, J. Stankovic, and L. Feng, “Sastl: Spatial aggregation signal temporal logic for runtime monitoring in smart cities,” in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, Apr 2020, p. 51–62.

[23] K. A. Ismail, “Application of computer vision techniques for automated road safety analysis and traffic data collection,” Ph.D. dissertation, University of British Columbia, 2010. [Online]. Available: https://open.library.ubc.ca/cIRcle/collections/ubctheses/24/items/1.0062871

[24] D. Gettman and L. Head, “Surrogate safety measures from traffic simulation models,” *Transportation Research Record*, vol. 1840, no. 1, p. 104–115, Jan 2003.

[25] J. C. Hayward, “Near-miss determination through use of a scale of danger,” 1972. [Online]. Available: https://trid.trb.org/view/115323

[26] P. J. Cooper, “Experience with traffic conflicts in canada with emphasis on “post encroachment time” techniques,” in *International Calibration Study of Traffic Conflict Techniques*, ser. NATO ASI Series, E. Asmussen, Ed. Springer, 1984, p. 75–96.

[27] F. Goudarzi and H. Asgari, “Non-cooperative beacon rate and awareness control for vanets,” *IEEE Access*, vol. 5, p. 16858–16870, 2017.

[28] Y. P. Fallah, C. Huang, R. Sengupta, and H. Krishnan, “Analysis of information dissemination in vehicular ad-hoc networks with application to cooperative vehicle safety systems,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, p. 233–247, Jan 2011.

[29] J. Aznar-Poveda, E. Egea-Lopez, A.-J. Garcia-Sanchez, and P. Pavon-Mariáo, “Time-to-collision-based awareness and congestion control for vehicular communications,” *IEEE Access*, vol. 7, p. 154192–154208, 2019.

[30] P. Fernandes and U. Nunes, “Platooning with ivc-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, p. 91–106, Mar 2012.

[31] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. L. Cigno, and F. Dressler, "Toward communication strategies for platooning: Simulative and experimental evaluation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, p. 5411–5423, Dec 2015.

[32] E. Bartocci, J. Deshmukh, A. Donze, G. Fainekos, O. Maler, D. Nickovic, and S. Sankaranarayanan, *Specification-Based Monitoring of Cyber-Physical Systems: A Survey on Theory, Tools and Applications*. Springer International Publishing, 2018, p. 135–175. [Online]. Available: https://doi.org/10.1007/978-3-319-75632-5_5

[33] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, ser. Lecture Notes in Computer Science, Y. Lakhnech and S. Yovine, Eds. Springer, 2004, p. 152–166.

[34] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Formal Modeling and Analysis of Timed Systems*, ser. Lecture Notes in Computer Science, K. Chatterjee and T. A. Henzinger, Eds. Springer, 2010, p. 92–106.

[35] N. Mehr, D. Sadigh, R. Horowitz, S. S. Sastry, and S. A. Seshia, "Stochastic predictive freeway ramp metering from signal temporal logic specifications," in *2017 American Control Conference (ACC)*. IEEE, May 2017, p. 4884–4889. [Online]. Available: http://ieeexplore.ieee.org/document/7963711/

[36] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[37] A. Wegener, M. Piorkowski, M. Raya, H. Hellbruck, S. Fischer, and J.-P. Hubaux, "Traci: an interface for coupling road traffic and network simulators," in *Proceedings*

*of the 11th communications and networking simulation symposium*, ser. CNS '08. Association for Computing Machinery, Apr 2008, p. 155–163. [Online]. Available: https://doi.org/10.1145/1400713.1400740

[38] A. Varga and R. Hornig, *An overview of the OMNeT++ simulation environment*, Jan 2008.

[39] A. Varga, *OMNeT++*. Springer, 2010, p. 35–59. [Online]. Available: https://doi.org/10.1007/978-3-642-12331-3_3

[40] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, January 2011.

[41] A. Donze, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, T. Touili, B. Cook, and P. Jackson, Eds. Springer, 2010, p. 167–170.

[42] P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, R. M. Murray, A. Donze, and S. A. Seshia, "A contract-based methodology for aircraft electric power system design," *IEEE Access*, vol. 2, p. 1–25, 2014.

[43] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Towards verified artificial intelligence," *arXiv:1606.08514 [cs]*, Jul 2020, arXiv: 1606.08514. [Online]. Available: http://arxiv.org/abs/1606.08514

[44] M. H. Zaki, T. Sayed, and K. Shaaban, "Use of drivers' jerk profiles in computer vision–based traffic safety evaluations," *Transportation Research Record*, 2014. [Online]. Available: https://journals.sagepub.com/doi/abs/10.3141/2434-13

[45] O. Bagdadi and A. Varhelyi, "Jerky driving—an indicator of accident proneness?" *Accident Analysis & Prevention*, vol. 43, no. 44, p. 1359–1363, Jul 2011.

[46] A. Rizaldi, F. Immler, and M. Althoff, “A formally verified checker of the safe distance traffic rules for autonomous vehicles,” in *NASA Formal Methods*, ser. Lecture Notes in Computer Science, S. Rayadurgam and O. Tkachuk, Eds. Springer International Publishing, 2016, p. 175–190.

[47] Y. Xuan and B. Coifman, “Lane change maneuver detection from probe vehicle dgps data,” in *2006 IEEE Intelligent Transportation Systems Conference*, Sep 2006, p. 624–629.

[48] A. Das, M. N. Khan, and M. M. Ahmed, “Nonparametric multivariate adaptive regression splines models for investigating lane-changing gap acceptance behavior utilizing strategic highway research program 2 naturalistic driving data,” *Transportation Research Record*, vol. 2674, no. 5, p. 223–238, May 2020.

[49] U. NHTSA, “Traffic safety facts,” *National Highway Traffic Safety*, 2009.

[50] M. Yang, X. Wang, and M. Quddus, “Examining lane change gap acceptance, duration and impact using naturalistic driving data,” *Transportation Research Part C: Emerging Technologies*, vol. 104, p. 317–331, Jul 2019.

[51] L. Zheng, K. Ismail, and X. Meng, “Freeway safety estimation using extreme value theory approaches: A comparative study,” *Accident Analysis & Prevention*, vol. 62, p. 32–41, Jan 2014.

[52] A. Tageldin, T. Sayed, K. Shaaban, and M. H. Zaki, “Automated analysis and validation of right-turn merging behavior,” *Journal of Transportation Safety & Security*, vol. 7, no. 22, p. 138–152, Apr 2015.

[53] T. Toledo and D. Zohar, “Modeling duration of lane changes,” *Transportation Research Record*, vol. 1999, no. 1, p. 71–78, Jan 2007.

[54] H. Park, C. Oh, J. Moon, and S. Kim, “Development of a lane change risk index using vehicle trajectory data,” *Accident Analysis & Prevention*, vol. 110, p. 1–8, Jan 2018.

[55] W. Qi, W. Wang, B. Shen, and J. Wu, “A modified post encroachment time model of urban road merging area based on lane-change characteristics,” *IEEE Access*, vol. 8, p. 72835–72846, 2020.

[56] A. Kesting, *Doctoral Thesis Microscopic Modeling of Human and Automated Driving: Towards Traffic-Adaptive Cruise Control*, 2008.

[57] M. Gueriau and I. Dusparic, “Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic,” in *23rd IEEE International Conference on Intelligent Transportation Systems*, 2020.