Traffic Rule Checking and Validation

by

Connor Raymond Stewart

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis presents a comprehensive exploration of traffic rule verification systems for diverse junction types, addressing key challenges in formalizing rules, determining violation thresholds, and covering a wide spectrum of relevant traffic scenarios. Leveraging iterative implementations and extensions of existing approaches, the associated program aims to concretize literature-based methods, and understand the severity of rule violations in naturalistic driving. The study extensively tests traffic rule adherence by vehicles in simulated and recorded traffic, utilizing Lanelet2, a versatile mapping system, to cover both signalized and non-signalized stop-regulated intersections. Through statistical analyses, the research delivers results on rule-violation thresholds, associated coefficients, and traffic violation rates, encompassing scenarios such as stop sign compliance, turns after stops, traffic light violations, offroad occurrences, speed limit violations, and tailgating instances. The thesis contributes specific test cases and insights from naturalistic driving, showcasing parameter settings and threshold determination for effective traffic rule implementation. The comprehensive approach taken in this research contributes to the advancement of traffic rule verification systems and provides a foundation for evaluating autonomous vehicle behaviours in diverse junction scenarios.

Acknowledgements

I would like to thank Krzysztof Czarnecki, Sean Sedwards, Michał Antkiewicz, and other members of Waterloo's WiseLab for helping me make this thesis possible.

Dedication

This is dedicated to my mother Sofia.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DAU Dominance Act Utilitarianism 11

HD Map high-definition map 1, 3, 119

Intel's RSS Intel's Responsibility-Sensitive Safety xiii, 11

S-PLTRI signalized permissive left-turn regulated intersection 30

S-RI signal regulated intersection xi, 2, 17, 24, 62–64, 70, 81, 96, 102–105, 107, 108

StVO Straßenverkehrsordnung 12

uS-SSRI unsignalized stop-sign regulated intersection viii, 2, 7, 11, 12, 14, 17, 31, 32, 34, 36, 38, 40, 42, 62, 64, 70, 72, 80, 81, 88, 95, 98, 107, 119, 120

WISE Lab Waterloo Intelligent Systems Engineering Lab 24, 27, 63, 108

List of Symbols

Δ A vehicle's distance to another point on the lanelet map M. It can represent the distance between a vehicle and the next stop line along the driving direction of the lanelet the vehicle is within. Delta can also represent the distance between one vehicle and another or between a vehicle and a waypoint. 18, 19, 26, 34, 38, 42, 50, 51, 57, 60, 66

s_t The state of a vehicle s at timestamp t in vehicles trajectory S. 19, 22–24, 26, 28–30, 34, 38, 40, 42, 45, 50, 51, 53, 55, 57–60

θ_Δ The threshold value for a vehicle at a stop to be considered within range of the stop line. x, 18–20, 34, 38, 42, 60, 66, 71, 93

ϵ Represents an error margin for a vehicle threshold value to be considered within a given target value. 45, 50, 51, 53, 57

M The lanelet map associated with recording a vehicle trip through an intersection. 19, 26, 30, 34, 38, 42, 45, 55, 57, 60, 70

d_{min} The minimum distance for a vehicle to be considered tailgating another vehicle according to Intel's Responsibility-Sensitive Safety (Intel's RSS) - *RSS.I* formulation [3]. 57–60, 119

w_p The state of a vehicle w at timestamp p in trajectory dataset R which is not the same as an associated given trajectory S. 28–30, 34, 38, 40, 42, 59, 60

$\theta_{\boldsymbol{speed}}$ Threshold value for a given vehicle *speed* to be considered stopped. x, 18, 19, 34, 38, 42, 45, 60, 81, 93

$t_{s|w}$ The amount of time a vehicle must be stopped before a stop sign for a stop to be registered as a valid stop. 17–19

R All vehicle trajectories in a given dataset. Vehicle trajectories are organized in the order they first appear along the intersection from the beginning to the end of the recording. 30, 34, 38, 42, 60

S Vehicle trajectory containing relevant information for a vehicle's position, velocity, and orientation (yaw) across a given intersection. The trip through the intersection is organized moment-to-moment by timestamps $(t_0, t_1, \dots)$ from beginning to end. 19, 23, 26, 30, 34, 38, 42, 45, 50, 51, 53, 55, 57, 60

Chapter 1

Introduction

The thesis centers on the assessment of traffic rule adherence and its application in traffic analysis and driving automation. The primary focus is on the development of algorithms for checking traffic rules given a high-definition map (HD Map) and a set of road-user trajectories. The challenges addressed involve the formulation of traffic rules and the related traffic concepts in a way that can be operationalized by the algorithms. This work builds on prior work by Maierhofer et al. (2022) and extends it to a comprehensive rule set for signalized and unsignalized intersections. It also validates the algorithms on real traffic recordings and extracts statistics on traffic rule violations. The thesis is driven by practical applications, including testing autonomous vehicles, analyzing traffic for traffic rule violations, and aiding autonomous vehicles in evaluating planned trajectories during route planning.

1.1 Thesis Objectives

The thesis addresses three objectives:

- **Objective One:** Devise algorithms for checking traffic rules given a HD Map and a set of road user trajectories.
- **Objective Two:** Validate algorithms on real traffic recordings.
- **Objective Three:** Extract statistics on traffic rule violations on real recorded traffic data.

1.2 Methodology and Contributions

The thesis builds upon the work of Maierhofer et al. (2022), focusing on the formalization of traffic rules in temporal logic for various types of intersections. In contrast to merely using temporal logic formalisms, the thesis introduces executable algorithms. This work extends the few sample rules of Maierhofer et al. (2022) in temporal logic are extended into an executable rule set to cover traffic rules for cross intersections, T-intersections, and roundabouts. The algorithms, implemented in C++, use the Lanelet2 format, a popular choice in automated driving and the WiseADS at Waterloo's WiseLab [4]. The Lanelet2 library is expanded to incorporate a broader range of traffic rules, including speed limits, target speeds, waypoints, collisions, permissive left turns, tailgating, and offroad violations. Datasets, such as the uS-SSRIs interaction dataset [5] and the WISE Labs Waterloo multi-agent traffic dataset for signal regulated intersections (S-RIs) [6], both utilizing Lanelet2, are employed for naturalistic data.

The thesis consists of five chapters, starting with an introduction providing an overview. The second chapter covers literature review topics, including background concepts and related works. The third chapter details the traffic rule checking system concept, addressing both intersectional and non-intersectional traffic rules. The fourth chapter focuses on the evaluation of the traffic rule-checking system, including implementation, testing, and statistical analysis of detected violations. The fifth chapter concludes the thesis, followed by references and appendices containing tailgating coefficients and detailed results of rule violation outcomes.

Chapter 2

Literature Review

This chapter first introduces important concepts and background knowledge relevant to the presented traffic rule checking system, followed by a discussion of related work.

2.1 Background

2.1.1 Lanelet2

Lanelet2 is an HD Map framework with specific map representations and operations to support planning in autonomous vehicles and simulation in automated driving projects. HD Maps offer precise information about a vehicle's surroundings, serving as a fundamental component for automated driving endeavours [1, 4]. These maps provide insights into areas that sensors may not observe, enable knowledge transfer from past trips, and are essential for accurately understanding diverse driving scenes.

The Lanelet2 system's approach builds upon earlier formats to meet evolving requirements, utilizing a top-down representation of roads. This representation involves defining roads using an imaginary center line, adding attributes to the center line, and storing information about the position of traffic lanes and the shape of the road border. Lanelet2 incorporates a novel approach derived from the Liblanelet framework, where atomic lanelets assemble to form lanes.

Critical considerations and requirements for automated driving using maps involve applications focusing on the road network, requiring detailed lane information and direct access to individual map elements. Road networks serve various purposes, including routing, which demands visibility of lane usage and ease of finding alternative lanes. Knowledge of individual lanes is also needed for behaviour generation, which involves choosing maneuvers such as overtaking or merging, necessitating knowledge of right-of-way rules, and motion prediction, which entails anticipating the actions of other road users under normal circumstances, incorporating traffic regulations.

Maps must contain traffic rules for all road users, encompassing pedestrians, cyclists, buses, and emergency vehicles. Precise lane geometry knowledge is crucial for path planning, involving factors like trajectory adjustment based on speed, obstacle avoidance, defining threshold distances for leaving the centerline, and comprehensive mapping of lanes with left and right borders. Special maneuvers in highly automated driving, such as parking or evading collisions, require specific map data detailing the surroundings of vehicles on the road. Ultimately, Lanelet2 addresses these complexities, providing a comprehensive framework for representing maps in automated driving scenarios.

Lanelet Design Considerations

The map is structured into three layers: the physical layer, the relational layer, and the topological layer.

Physical elements play a pivotal role in mapping for automated vehicles, especially in the context of localization where observable elements are indispensable. Maps need to incorporate elements that can be observed by a variety of sensors, catering to vehicles with diverse sensor setups. These elements contribute to precise localization and facilitate motion planning, utilizing features such as crash barriers and roadsides.

The relational layer of maps incorporates traffic rules that establish associations between the source of a rule (e.g., traffic lights) and the lanes to which they apply. Notably, not all map elements can be directly associated with observable objects, as implicit rules also play a role in traffic scenarios.

The topological layer emerges implicitly from the contexts and neighbourhood relationships defined in the relational layer. All elements in the map are assumed to be describable as projections onto a flat ground plane.

Lanelet Architecture

Lanelet2 maps are composed of five key elements, categorically organized into the physical layer and the relational layer. In the physical layer, maps include points and linestrings, while the relational layer encompasses lanelets, areas, and regulatory elements. Each element is uniquely identified and assigned key-value pairs.

1. **Points:**
   - Represent vertical structures and are integral elements of the map.
   - Only elements with position information and other elements are directly or indirectly composed of points.
2. **Linestrings:**
   - Ordered arrays of two or more points, enabling linear interpolation between them.
   - Define the shape of elements in the map.
3. **Lanelets:**

- Define atomic sections of the map where directed motion occurs.
- Comprise a one-line string on the left and exactly one on the right border.
- As atomic elements, lanelets represent current and unchanging traffic rules within a lanelet.
- Topological relationships with other lanelets remain constant.

4. **Regulatory Elements:**

   - Express traffic rules that apply to specific lanelets.

5. **Areas:**

   - Sections of the map where undirected or no movement is possible.
   - Include diverse spaces like parking areas, squares, green spaces, and buildings.
   - Defined in line strings and may incorporate regulatory elements.

These elements, each equipped with a unique ID and key-value pairs, collectively form the Lanelet2 maps, providing a structured representation of the physical and relational aspects of the environment. The combination of points, linestrings, lanelets, areas, and regulatory elements serves to create a comprehensive and versatile mapping system for various applications in automated driving.

The following image (Figure 2.1) gives an illustration of lanelets:

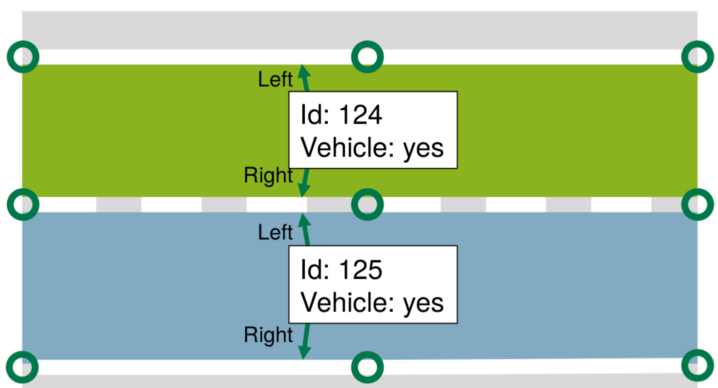


Figure 2.1: Lanelet Diagram [1].

Regulatory elements elements within Lanelet2 maps play a crucial role in defining traffic rules, encompassing aspects such as speed limits, priority rules, or the presence of traffic lights. These elements are referenced by one or more lanelets or areas to which they apply. The structure of regulatory elements can vary significantly due to the diverse nature of traffic rules that they represent. This flexibility allows Lanelet2 to accommodate a wide range of traffic regulations, reflecting the nuanced and varied rules governing different parts of the road network.

Below is an illustration of a stop sign regulatory elements in a Lanelet2 environment:

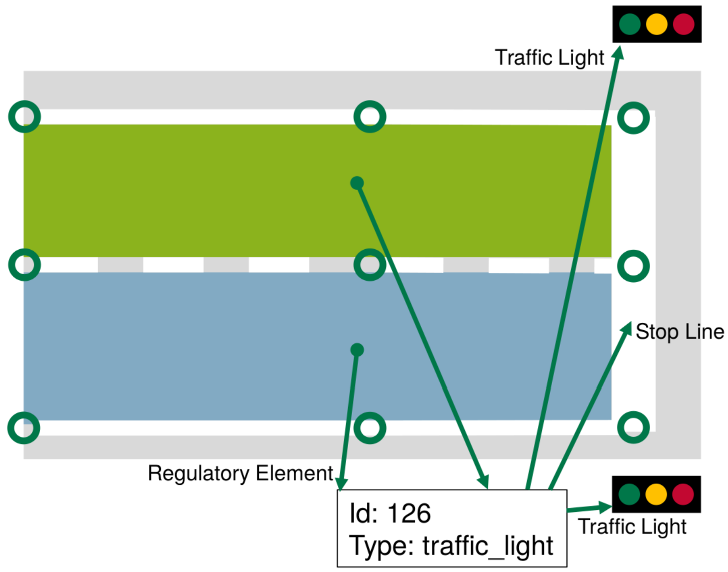


Figure 2.2: Lanelet Regulatory Elements [1].

Lanelet2 incorporates multiple regulatory elements, each serving distinct functions within the automated driving context [1]:

1. **Speed Limits:**
   - Used to specify permissible vehicle speeds in designated areas.
   - Determine and identify violations if a vehicle exceeds the specified speed limit.
2. **Traffic Signs:**
   - Express restrictions are typically represented by generic traffic signs.
   - Provide standardized symbols to convey specific instructions or limitations.
3. **Traffic Lights:**
   - Define the positions of traffic lights, including associated stop lines if applicable.
   - Govern vehicle movements based on the signals displayed by the traffic lights.
4. **Right of Way Regulatory Elements:**
   - Reference lanelets, indicating those that must yield the right-of-way and those that have the right-of-way over yielding lanelets.
   - Crucial for establishing the hierarchy of movement at intersections.
5. **All Way Stop Regulatory Elements:**

- Regulate uS-SSRI scenarios.
- Order of arrival determines the right-of-way, with potentially all lanelets in the intersection having to yield the right-of-way.

These regulatory elements collectively contribute to the comprehensive representation of traffic rules within Lanelet2 maps, ensuring the system's capability to handle a variety of real-world scenarios and traffic conditions in an automated driving environment.

Laenlet Modularization

In Lanelet2, a fundamental principle is the clear separation between the representation of map elements and their interpretation, promoting consistent modularization of tasks. The core module, equipped with basic primitives and geometry functions, handles the representation of map elements. The interpretation of these elements is then delegated to specialized modules, ensuring a modular and extensible architecture. Lanelet2 comprises the following modules:

1. **Core:**
   - Contains basic primitives and layers, along with essential geometry functions.
2. **Traffic rules:**
   - Interprets rules in the map, considering the type of road user and country.
   - Determines the feasibility and permissibility of lane changes based on traffic rules.
3. **Physical:**
   - Allows direct access to elements in the physical layer of the map.
4. **Routing:**
   - Determines routes to be driven or possible routes, including points of conflict.
   - Used to construct maneuverable zones within the map.
5. **Matching:**
   - Assigns lanelets to road users to ascertain their positions on the map.
6. **Projection:**
   - Converts global latitude/longitude coordinates to metric coordinates for consistency.
7. **IO (Input/Output):**

- Provides functions for reading and writing maps from various map formats.

8. **Validity:**
   - Searches for and reports typical mapping errors in a map, ensuring data integrity.
9. **ROS (Robot Operating System):**
   - Establishes a connection to the Robot Operating System (ROS) for demonstrator purposes.
10. **Python:**
    - Facilitates the use of Lanelet2 modules in Python, enhancing accessibility and usability.

This modular approach allows Lanelet2 to be versatile and adaptable, supporting diverse tasks in automated driving applications. Each module is specialized for specific functions, enabling efficient and comprehensive handling of map-related operations.

Lanelet Routing

Lanelet routing relations encompass various attributes that define the connections and relationships between different lanelets within a Lanelet map. These attributes include:

1. **Left and Right Lanelets Reachable by Lane Changes:**
   - Identifies lanelets that can be reached by making lane changes to the left or right.
2. **Adjacent Left and Right Lanelets:**
   - Describes lanelets that are neighbours but may not be directly reachable by lane changes.
   - Implies a spatial adjacency between lanelets without explicit connections for lane changes.
3. **Succeeding Lanelets:**
   - Refers to lanelets that follow a given lanelet in a sequential manner.
   - Defines the order or sequence of lanelets along a particular route.
4. **Conflicting Lanelets:**
   - Involves lanelets with intersecting areas.
   - Indicates potential conflicts where the trajectories or paths of different lanelets intersect.

- Awareness of conflicting lanelets is crucial for managing traffic interactions and ensuring safe navigation.

These attributes provide a comprehensive description of how lanelets are related in terms of spatial and sequential arrangements, as well as potential conflicts. This information is essential for tasks such as routing, maneuver planning, and traffic rule interpretation in the context of automated driving systems using Lanelet maps. See the image (Figure 2.3) below:

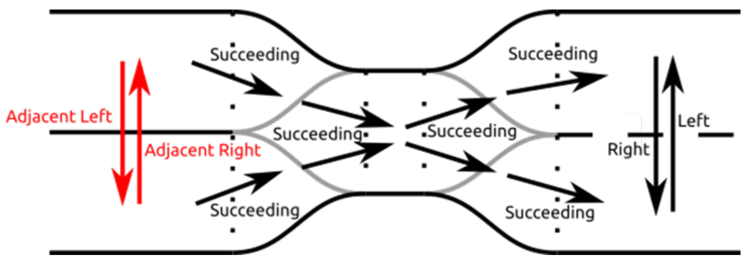


Figure 2.3: Lanelet Routing Diagram [1].

Listed are fundamental concepts for understanding and describing navigation, route planning, and sequential organization of lanelets and areas within the Lanelet2 framework:

1. **Route:**
   - A route consists of all lanelets that can be traversed to reach a destination without driving on a different road.
   - Generic lane changes serve as connectors between different routes.
2. **Path:**
   - Paths are ordered lists of lanelets and areas leading to a destination.
   - Lane changes bound paths, indicating a specific sequence of lanelets and areas to follow for reaching the destination.
3. **Sequence:**
   - A sequence refers to a continuous series of lanelets not separated by a lane change.
   - It encompasses lanelets that are directly connected without the need for changing lanes.

The concepts provide a structured way to represent and manage the complex relationships and interactions involved in automated driving scenarios.

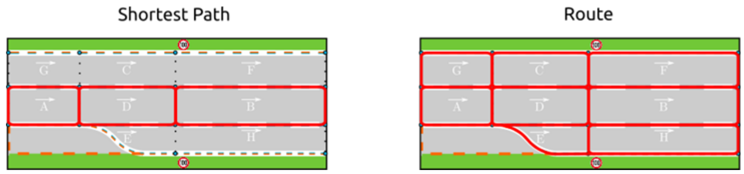


Figure 2.4: Lanelet Pathing Diagram [1].

Predicates for Traffic Rules

The essential predicates and functions for formalizing and establishing traffic rules are categorized into position, regulatory, velocity, braking, and temporal elements [2]. When vehicles occupy lanelets, it is formalized as follows: lanelets(x_k) $=$ l $\in$ L | occ(l) $\cap$ occ(x_k) $\neq \emptyset$, where the occupied lanelet (l) intersects with the vehicle's occupancy (occ(x_k)). The lanelet a vehicle occupies depends on its reference point, with the centerpoint determining the occupied lanelet.

In terms of vehicle interactions, a vehicle is considered to be approaching an intersection from the left of another vehicle if its lane is to the left of the other vehicle's lane. Conflict arises when two vehicles are in the same lane but driving in different directions and approaching an intersection from different incoming lanelets.

Specific traffic signs influence a vehicle's behaviour if its lanelet references a sign of a specified type. Stopping at a designated stop line is considered when a vehicle plans its movements, and priority between vehicles is determined by the occupancy of lanelets with higher priority values.

Checking traffic-light-regulations involves verifying the active status of traffic lights on the lanelet being driven on and filtering out inactive lights. The colour and direction of a traffic light are considered based on the active lights for the vehicle and their corresponding designated directions.

Vehicle braking is determined by the proximity and acceleration of other vehicles, causing braking if certain conditions are met. Vehicles can come to a standstill before entering intersections, and the definition of a standstill involves near-zero velocity, accounting for measurement uncertainties.

To model the relationship of passing a stop line, the formulation includes conditions such as the stop line being in front of the vehicle at the current time step and not in front of it at the next time step. This relationship is expressed as Passing_stop_line(x_k)::= stop_line_in_front(x_k) $\wedge$ X($\neg$stop_line_in_front(x_k)).

2.2 Related Works

The thesis at hand is contextualized within the broader landscape of recent works that have addressed the same problem. Notable among these is *"A Deontic Logic Analysis of Autonomous Systems' Safety"* by Shea-Blymyer and Abbas (2020). Additionally, the work by Maierhofer et al. (2022) titled *"Formalization of Intersection Traffic Rules in Temporal Logic"* provides further insights into the problem domain. These related works offer diverse approaches to the challenges outlined in the present thesis.

2.2.1 Deontic Logic Analysis of Autonomous Systems Safety

Shea-Blymyer and Abbas (2020) delve into the implementation and application of deontic logic analysis for ensuring the safety of autonomous systems. The exploration aims to enhance our comprehension of the logical framework guiding agents in adhering to specific obligations and abstaining from impermissible behaviours. Dominance Act Utilitarianism (DAU) serves as the encoding and reasoning tool for autonomous system obligations. The study employs DAU to analyze Intel's RSS rules, a comprehensive set designed to eliminate traffic collisions if universally followed by all vehicles. By applying DAU, the paper uncovers potential undesirable consequences of these rules, offering insights into the design of systems with well-defined obligations. The authors present a method for model checking DAU obligations, showcasing the practical application of this analytical approach. The thesis also explores the analysis and implementation of Intel's RSS rules for traffic route validation.

The implemented system follows Intel's RSS rules, focusing on rules one through three, and six. These rules dictate that vehicles must avoid rear-end collisions, refrain from reckless cutting in, yield the right-of-way without assuming it, and initiate lane changes without waiting for a perfect gap. Notably, the system's ability to derive autonomous vehicle obligations, permissions, and potential undesirable outcomes on the road illustrates a novel approach. Instead of exhaustively programming specific rules, the study demonstrates the power of deontic logical reasoning in enabling autonomous vehicles to make inferences and engage in logical knowledge representation. The thesis discusses the derivation of rule violations and the determination of coefficients for rule violation thresholds to identify rule-violating outcomes.

2.2.2 Formalization of Intersection Traffic Rules in Temporal Logic

A challenge arises when attempting to model traffic using logic because traditional logic approaches, such as Answer Set Programming (ASP), propositional logic, or ontologies, lack the capability to incorporate time relations. For instance, widely used temporal logics like Linear Temporal Logic (LTL) and co-safe Temporal Logic (scLTL) are limited in their ability to represent durations. On the other hand, Metric Temporal Logic (MTL) and Signal Temporal Logic (STL) have the capability to model durations. The determination of the right-of-way for drivers stopping simultaneously along uS-SSRIs was a focal point of the Line 5, Line 6, and Line 7 algorithms. Formulations and examples from the paper were leveraged to assist in the implementation and formulation of the algorithms used in the thesis.

To address this challenge, the use of rulebooks becomes crucial. Rulebooks provide a means to formalize and prioritize traffic rules, facilitating decision-making processes, especially in situations where one rule may need to be violated to adhere to another.

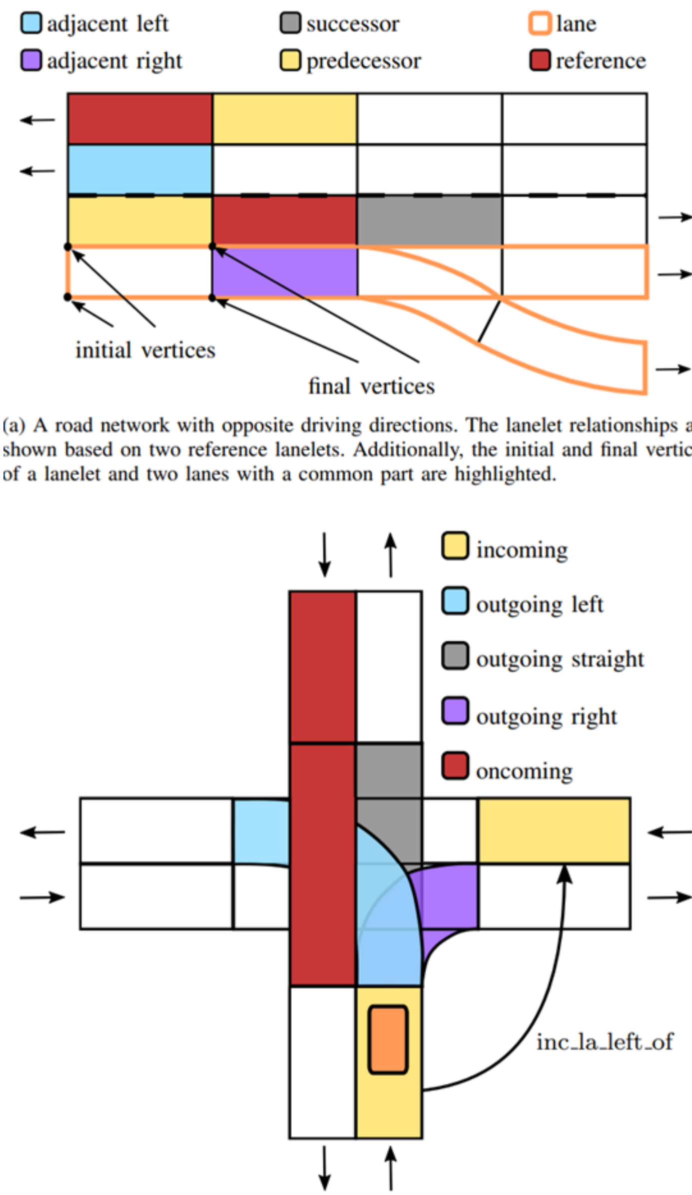
This prioritization is essential for effective integration into a trajectory planner's control strategy, allowing for dynamic and context-aware navigation in response to real-time traffic conditions. Determining the right-of-way of drivers who stop simultaneously along uS-SSRIs was a focus of the Line 5, Line 6, and Line 7 algorithms. Formulations and examples from the paper were used to aid with the implementation and formulation of the algorithms used in the thesis.

Road Networks, Regulatory Elements, and Methodology

The paper titled *"Formalization of Intersection Traffic Rules in Temporal Logic"* by Maierhofer et al. (2022) focuses on defining road networks for intersections and provides formalized traffic rules based on legal sources in temporal logic for various intersection types. The article introduces sets of predicates and functions in higher-order logic to formalize diverse traffic rules, making them adaptable to additional national requirements. The evaluation of intersection rules involves the use of over 2000 recorded and simulated traffic participants. Legal sources for traffic information are drawn from German traffic regulations (Straßenverkehrsordnung (StVO)), judicial decisions, law literature comments, and feedback from legal experts.

Several assumptions underlie the formalization process, including the exclusion of signs from police officers, consideration of only a single intersection in the road network, and the omission of aspects related to crossing, pedestrians, cyclists, railroad vehicles, and buses. Additionally, the formalization assumes an unobstructed view of the ego vehicle over the intersection, and certain aspects like congestion, waiting within intersections, and behaviour before intersections are not covered.

Road networks are structured based on the CommonRoad format, utilizing the concept of lanelets, which are detailed further below. These road networks are defined as a set of lanelets: $L \cup \perp$, with $\perp$ serving as a bottom element in cases where no lanelet exists. A lanelet is characterized by a left and right boundary defined by a polyline, denoted as $l \in L$. Crucial information, including initial, final, and left boundary vertices, along with successor, predecessor, left, and right lanelets, is employed to assemble lanelets into lanes – as can be seen in the figure below:



(a) A road network with opposite driving directions. The lanelet relationships are shown based on two reference lanelets. Additionally, the initial and final vertices of a lanelet and two lanes with a common part are highlighted.

(b) A schematic intersection with labels assigned from the orange vehicle's perspective.

Figure 2.5: Schematic Road Networks [2].

The description of turn directions in the current state involves the representation of traffic lights, where each lanelet is associated with a single traffic light. The current traffic light states fall into distinct categories:

- **Red:** Vehicles must stop in front of the intersection. If the traffic light displays an arrow, this restriction only applies to the specified direction.
- **Yellow:** Vehicles are required to wait in front of the intersection for the next signal.
- **Green:** Traffic is permitted to proceed, but only in the specified direction.

- **Inactive:** The traffic light is not in operation and should be disregarded.

Intersections are defined by adding special intersection labels to lanelets that form part of the intersection. The labelling of turning directions for lanelets occurs from incoming lanelets to ongoing lanelets, providing a comprehensive perspective from the viewpoint of vehicles navigating the intersection. In the thesis, Lanelet2 is utilized instead of the CommonRoad format, enabling extensions to the rule violation checking system through the capabilities of the Lanelet2 library. Furthermore, rule implementations are coded in C++ rather than relying on temporal logic.

2.2.3 GeoScenario

GeoScenario serves as a domain-specific language (DSL) designed for the representation and evaluation of scenarios [8]. Scenarios encompass the utilization of actors, background details on these actors, and underlying assumptions related to environments, goals, actions, and events. The scenario unfolds starting with an initial scene, and temporal developments are characterized by actions, events, goals, and values spanning a defined period. Scenes can be perceived as snapshots capturing the environment, comprising scenery (stationary elements), dynamic elements (entities capable of movement or state changes within the scene), actors, and self-representation of the observer (attributes and states). The evaluation of naturalistic driving data in the thesis employs environments and elements similar to those presented in the GeoScenario format.

Within the context of GeoScenario, the scenery consists of the road network and encompasses topological information regarding roads and their semantics. Temporal development from an initial scene involves the progression of scenarios along alternative paths, leading to distinct scenes. Each path represents an individual scenario, and the scenes provide interpretations of the environment at different points in time. The thesis incorporates pathing and planning mechanisms to assess rule violations specifically related to right-of-way violations in the context of uS-SSRIs.

2.2.4 Rule-Based Behaviour Planners

The research article by Bouchard et al. (2022) delves into the functionality of advanced decision-making for planning the motion of self-driving autonomous vehicles using a practical rule engine that learns from expert driving decisions. The study involves two layers of a rule-based theory: the maneuver layer, which identifies compatible sets of conservative behaviours, and the parameter layer, which resolves different parameters into a single high-level maneuver. The proposed rule engine operates with unordered rules, mapping input properties to parameterized output behaviours. The thesis operates by integrating the Lanelet2 library with naturalistic driving data. This integration is essential for identifying rule violations by analyzing the vehicle's behaviours and evaluating the potential choices it could have made at different moments.

The two-layer rule engine functions as follows: first, sensors provide perceived states to the maneuver layer, which identifies compatible sets of behaviours based on maneuver rules. Second, transformation properties transform and complete the resulting properties for input to the parameter layer. Third, the parameter layer resolves a single behaviour using parameter rules and outputs the information to the local planner for interpretation. Unlike GeoScenario's approach which relies on sensors mapped to behaviors, this thesis adopts a unique methodology by utilizing pre-recorded naturalistic data from drivers for its analysis.

The learning process involves experts providing finite sets of training scenes, and a backward-chaining coverage function facilitates learning. The rule engine filters candidate behaviours, and random selection ensures unbiased solutions. Discrepancy identification involves scenes that exemplify discrepancies, and maneuver diagnosis uses forward and backward chaining to identify misclassifications. The analysis of conflicting scenes and the elimination of irrelevant properties are crucial steps in refining rules and connecting rule theories with rule practices for self-driving autonomous vehicles. The thesis involves the derivation and testing of coefficients to refine rule implementations and determine optimal violation detection constraints.

Experimental results with a prototype vehicle demonstrate successful learning of driving policies, with autonomous vehicles operating in urban environments using a rule-based approach. Field tests show promising results, with safety drivers needing interventions comparable to end-to-end deep learning approaches. The study concludes that autonomous vehicles can handle tasks requiring interventions, but challenges may arise from external limitations beyond the rule engine. The project highlights the feasibility of the presented approach for achieving level-three autonomous vehicle behaviour, suggesting potential avenues for refining rules based on statistical preferences and addressing more complex vehicle interactions in future research. The presented thesis could prove valuable for researchers involved in implementing higher-level autonomous vehicle behaviors.

2.2.5 CommonRoad

CommonRoad is a comprehensive framework designed for motion planning on roads, offering a collection of composable benchmarks that enable researchers to evaluate and compare their motion planning algorithms [10]. Each benchmark in CommonRoad includes a scenario with a planning problem, a vehicle dynamics model, vehicle parameters, and a cost function, all associated with a unique ID. The framework provides tools for motion planning, benchmarks, and an Input-Output package facilitating the reading, writing, and visualization of CommonRoad scenarios and planning problems. The thesis focuses on implementations using the Lanelet2 library rather than the CommonRoad format. Like Lanelet2, CommonRoad's is also based on the Lanelet format and is used as an example format.

The CommonRoad Scenario Designer is a component of the framework tailored for creating new scenarios. It supports tasks such as converting maps from file formats like Open Street Maps and OpenDRIVE. The Scenario Designer offers a graphical user interface

(GUI) for manual editing of map scenarios and populating them with traffic using the traffic simulator SUMO. Interactive scenarios allow other traffic participants to react to the ego vehicle, and scenarios can be recorded if all other traffic participants follow pre-recorded trajectories. Rather than using a scenario designer format the thesis implements a program to evaluate naturalistic driving data.

For trajectory verification, CommonRoad provides a toolkit equipped with a drivability checker. This tool, commonly used in motion planning, verifies the drivability of a trajectory, checks for collisions, and performs transformations into curvilinear coordinate systems. The package also includes a route planner that offers high-level guidance for motion planning algorithms and defines reference paths. The Lanelet2 format also implements many subsystems similar to those in CommonRoad, which serves as an example for the thesis.

CommonRoad-Search is another component of the framework that offers a selection of search algorithms with motion primitives to solve motion planning problems. It illustrates how motion planning primitives are generated and demonstrates their use with a batch processor for searching solutions with parallel execution. The Lanelet2 format also implements primitives for evaluating rule violations, similar to the CommonRoad format.

Overall, CommonRoad serves as a valuable resource for researchers in the field of autonomous vehicles, providing a standardized framework for evaluating and developing motion planning algorithms.

Chapter 3

Traffic Rule Checking System Concept

3.1 Intersectional Traffic Rule Concepts

Intersectional traffic rule concepts are used to determine vehicle right-of-way priority and rule violation detection across intersections. Detecting right-of-way violation for S-RIs includes red light, yellow light, and permissive green left-turn violation detection. Furthermore, uS-SSRIs rule violation detection includes detecting stop sign compliance and illegal movements after stopping at stop-regulated intersections.

Intersectional traffic rule concepts play a crucial role in determining the right-of-way priority for vehicles and detecting rule violations at intersections. The detection of right-of-way violations for S-RIs scenarios involves identifying infractions related to red lights, yellow lights, and permissive green left-turn signals. Additionally, violations for uS-SSRIs scenarios include detecting non-compliance with stop signs and illegal movements after stopping at stop-regulated intersections.

3.1.1 Vehicle Stopping at Stop Signs

Stop Sign Violations The *Stop Sign Checking* rule involves analyzing a recorded trip for a vehicle and verifying whether the vehicle has come to a complete stop at all the designated stop signs along its route.

Rule Role Vehicles are expected to wait for a minimum duration of $t_{s|w}$ seconds when they are stationary in front of the stop line. This time duration can be adjusted, and setting it to zero represents a scenario where a vehicle performs a complete stop without a specific duration. This rule is applicable to vehicles positioned at a stop-sign-regulated stop line, rather than a traffic-light-regulated one, on their designated lane.

Example The provided example (Figure 3.1) illustrates a scenario in which a vehicle is required to come to a stop before crossing a double line at a four-way intersection. It's important to note that there are no traffic lights regulating this intersection. The example specifies a stopping threshold (θ_{speed}) value of 2 m/s and a distance to the stop line (θ_{Δ}) of six meters. In the depicted image, the vehicle's current speed is measured at 0.1 m/s, and its distance to the stop line (Δ) is one meter. According to the defined threshold values, the vehicle is considered to be at a complete stop since its current speed and distance to the stop line fall within the specified range.

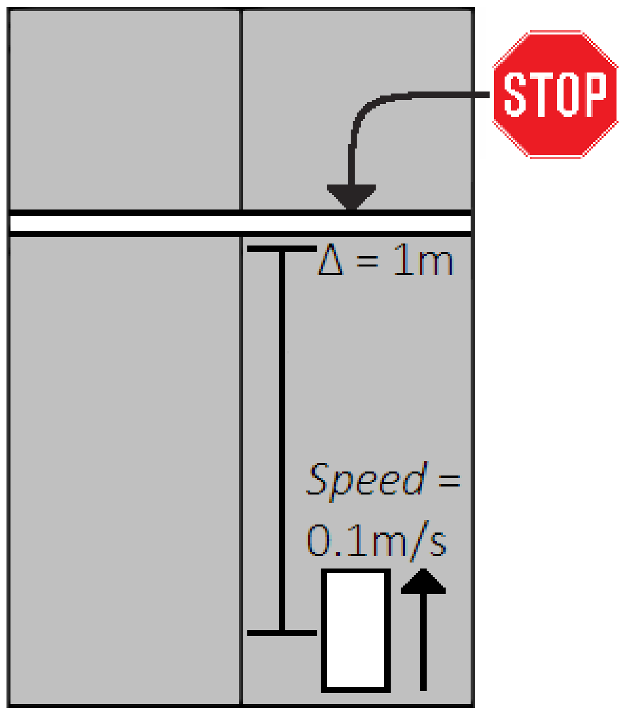


Figure 3.1: A stop-sign example showing stop threshold values.

Stop Sign Checking Algorithm The provided algorithm, named the *Stop Sign Checking Algorithm* is designed for evaluating stop compliance within a recorded vehicle trip. The algorithm returns a True value if the vehicle has successfully stopped at all designated stop signs along its recorded trip; otherwise, it returns a False value. Here's a breakdown of the algorithm's key components:

1. **Line 7:** An initial check to determine if the vehicle is close enough to a stop line, indicating the possibility of encountering a stop sign.

2. **Line 9:** An IF statement verifying if the vehicle has come to a complete stop, recording the time at which the stop occurred.

3. **Line 12:** An ELSE IF statement checking if the vehicle has started moving again after a stop.

4. **Line 14:** This line verifies if the vehicle, before reaching an upcoming stop line ($\Delta \leq \theta_{\Delta}$), has indeed come to a stop (vehicle_is_stopped) and remained stopped for the required duration ($t - stoptime \leq t_{s|w}$) to consider the stop valid.

Stop Sign Checking: Determines if a vehicle failed to stop at a traffic sign

Parameters: Distance threshold (θ_Δ), Speed threshold (θ_{speed}), Minimum stop duration ($t_{s|w}$)
Input Data: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \ldots$), Lanelet map ($M$)
Result: Returns FALSE if a vehicle fails to stop at a stop line appropriately

vehicle_must_stop $\leftarrow$ FALSE
stopped_at_sign $\leftarrow$ FALSE
vehicle_is_stopped $\leftarrow$ FALSE
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
    speed $\leftarrow$ Get vehicle's speed in state s_t
    $\Delta \leftarrow$ Get vehicle's distance to next stop line on lanelet map (M) in state s_t
    `// Check if the vehicle has encountered a nearby stop line`
    if $\Delta \leq \theta_\Delta$ **then**
        vehicle_must_stop $\leftarrow$ TRUE
    `// Check if the vehicle is at a standstill before a stop line`
    if $speed \leq \theta_{speed}$ *AND* ¬vehicle_is_stopped **then**
        vehicle_is_stopped $\leftarrow$ TRUE
        stoptime $\leftarrow t$
    else if $speed > \theta_{speed}$ *AND* vehicle_is_stopped **then**
        `// Flag that the vehicle has started moving again`
        vehicle_is_stopped $\leftarrow$ FALSE
    `// Check if a vehicle has completed its stop within` $t_{s|w}$ `time`
    if vehicle_is_stopped *AND* $\Delta \leq \theta_\Delta$ *AND* $t - stoptime \geq t_{s|w}$ **then**
        stopped_at_sign $\leftarrow$ TRUE
return (stopped_at_sign OR ¬vehicle_must_stop)

Limitations The *Stop Sign Checking Algorithm* has some limitations that need to be considered. One limitation is related to using a vehicle's centerpoint to determine the distance to the stop line. This approach may result in inaccuracies for very long vehicles, such as trucks, where stopping far past the stop line could still be considered a valid stop.

Another limitation involves the use of stop line distances (θ_Δ) that are too long, which can lead to false negatives and allow rule-violating vehicles to pass the check. In certain situations, vehicles positioned behind others may be incorrectly flagged as having stopped before the stop line, even if they run the stop sign rather than making a complete stop. Conversely, using a small stop line distance (θ_Δ) may lead to false positives, where valid stops are not registered. Adjusting the θ_Δ parameter is crucial for accurate stop sign violation detection. Below are examples illustrating these limitations:

- **False Negative Example** (Figure 3.2):
  - **Scenario:** Vehicles behind others can be flagged as having stopped before the stop line, even if they violate the stop sign by not coming to a complete stop.
  - **Observation:** Vehicle 71 (inside the black rectangle) is incorrectly flagged as successfully stopping (in other words, it *failed* the violation check inappropriately) before the stop line despite being partially behind another vehicle.
- **False Positive Example** (Figure 3.3):

- **Scenario:** A false-positive stop line violation detection occurs when the $\theta_\triangle$ is set too small for a valid stop to be registered.
- **Observation:** Vehicle 30 (inside the black rectangle) triggers a false-positive stop line violation detection with a $\theta_\triangle$ of 2m.

These examples emphasize the importance of carefully tuning parameters like $\theta_\triangle$ to avoid both false negatives and false positives in stop sign violation detection. The optimal parameter values may vary based on the characteristics of the road network and the types of vehicles involved.

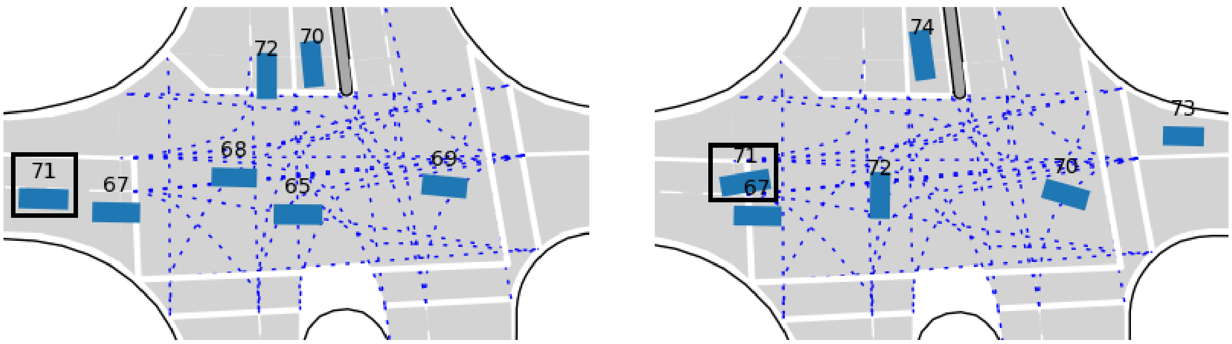


Figure 3.2: An example of a vehicle being falsely flagged as having successfully stopped before a stop line.

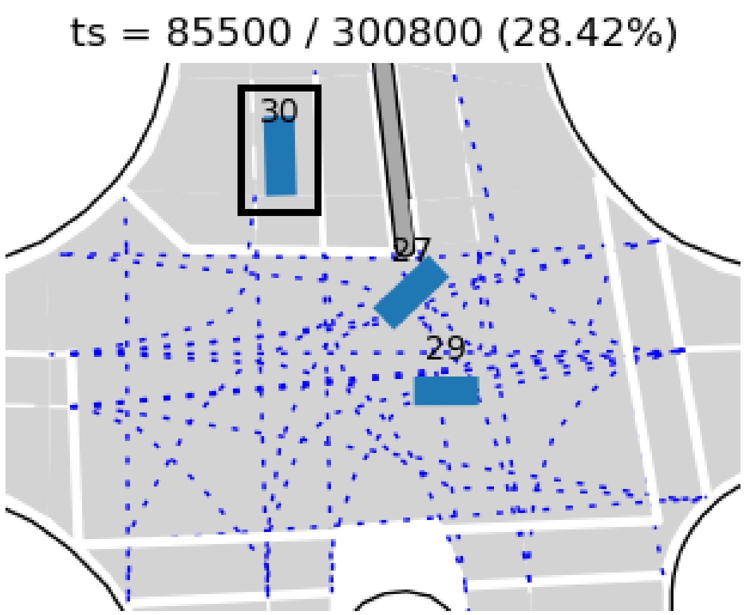


Figure 3.3: An example of a false-positive when checking for stop sign violations.

3.1.2 Traffic Light Rules

To enforce traffic light rules, the system needs to check various conditions related to the state of the traffic lights and the behaviour of the vehicle. The following rules can be considered for traffic lights:

- **Stop at Red Lights:**
  - Condition: If the traffic light is red.
  - Action: The vehicle must come to a complete stop before the stop line.
- **Stop at Yellow Lights:**
  - Condition: If the traffic light is yellow.
  - Action: The vehicle should prepare to stop, and if it is safe to do so, come to a stop before the stop line.
- **Permissive Left Turns at Green Lights:**
  - Condition: If the traffic light is green, and the vehicle intends to make a left turn.
  - Action: The vehicle can proceed with a left turn if there is no conflicting traffic.

It's important to consider additional factors such as the presence of other vehicles and the specific traffic rules of the intersection. Additionally, the behaviour of the vehicle should align with local traffic regulations such as speed limits.

These rules aim to ensure safe and compliant behaviour at traffic lights. The system is designed to interpret the state of the traffic lights accurately and make decisions based on the current traffic signal phase. Violation checks are performed to identify instances where the vehicle does not adhere to these rules.

Stopping at Red Lights

Red Light Violations The *red light violations* is designed to assess whether a vehicle has disregarded traffic regulations concerning red lights at intersections.

Rule Role The rule regarding red light violations at intersections is straightforward: vehicles are only permitted to enter intersections when the traffic light is displaying a green or yellow signal. If a vehicle enters an intersection during a red light phase, it is considered a violation of the red light intersection rule regulation. This rule aims to ensure that vehicles adhere to traffic signal indications, promoting safe and orderly intersection crossings. Violations trigger appropriate actions or alerts to address non-compliance and enhance overall traffic safety.

Example The provided illustration (Figure 3.4) depicts a clear instance of a red light violation. It shows a vehicle crossing a stop line at an intersection while the traffic signal is displaying a red light.

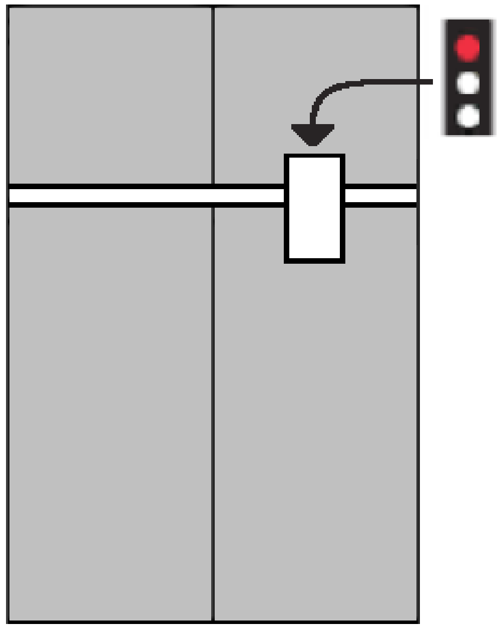


Figure 3.4: An illustration of a vehicle at a red light.

Red Light Violation Algorithm The provided algorithm, named the *Red Light Violation Algorithm* is designed for evaluating red light violation detection. Here's a breakdown of the algorithm's key components:

- **Line 4:** Examines every state linked to the trajectory of the reference vehicle (s_t).
- **Line 7:** Verifies if the reference vehicle crosses the intersection during an active red light signal state at the traffic light. Identifies a red-light traffic violation when the specified conditions are met.
- **Line 10:** Indicates failure if no red light violations occur throughout the trajectory.

Red Light Violations: Determines if a vehicle failed to stop at a red light during a trip.

```
Input Data: Vehicle trajectory (S = s_{t_0}, s_{t_1}, ...)
Parameters: None
Result: Returns a Boolean value representing whether a vehicle ran a red light.
entry_gate ← The intersection-entry gate direction of the vehicle along vehicle_track
exit_gate ← The intersection-exit gate direction of the vehicle along vehicle_track
gateway_traffic_light ← The light regulating the intersection between entry_gate and
    exit_gate
// Every moment in the recorded vehicle trajectory
for every state s at timestamp t (s_t) in S do
    // Check if the entry-gate traffic light state change occurred within
       the vehicle trajectory time-frame
    current_light_state ← Get the light state at the current vehicle state (s_t) for the
        traffic light gateway_traffic_light
    entered_intersection ← Check if the front of the vehicle at state s_t has entered the
        intersectional entry gate stop line
    if current_light_state is Red AND entered_intersection then
        return TRUE
    entered_intersection ← FALSE
return FALSE
```

Limitations A limitation of the approach is its expectation that vehicles should come to a stop before red lights, irrespective of intersection constraints. For instance, if an intersection is located close to a train track level crossing, the algorithm would anticipate a vehicle to halt before a red light, potentially leading to the obstruction of traffic, especially in or near the train track crossing.

Stopping and Yellow Lights

Yellow Light Violations The *yellow light verification* rule assesses whether a vehicle has infringed yellow light stopping regulations, indicating a violation if the vehicle proceeds through a yellow light when it could have safely come to a stop at the intersection.

Rule Role Vehicles approaching intersections are expected to come to a stop at a yellow light if it is feasible. However, they are permitted to proceed through the intersection if they are moving too fast to safely stop at a yellow light. This decision is based on factors such as driver reaction time and the vehicle's stopping distance after the brakes are applied. The determination of whether a vehicle can stop at a yellow light or should continue through the intersection is made when the light signal transitions from green to yellow.

Example As evident from the image (Figure 3.5), a specific vehicle (Vehicle 258 from dataset 775) sourced from the Signal Regulated Intersection database provided by the Waterloo Intelligent Systems Engineering Lab (WISE Lab) [6] failed to stop at a yellow light. This failure occurred because its calculated braking distance was shorter than its calculated gate distance. The calculated braking distance considers the distance covered during the driver's reaction to traffic signals and the subsequent distance travelled while the vehicle comes to a stop after applying the brakes. In contrast, the calculated gate distance is determined based on the reported x-and y-coordinates of the vehicle at the moment the traffic light turns yellow and the corresponding values at the entry gate.

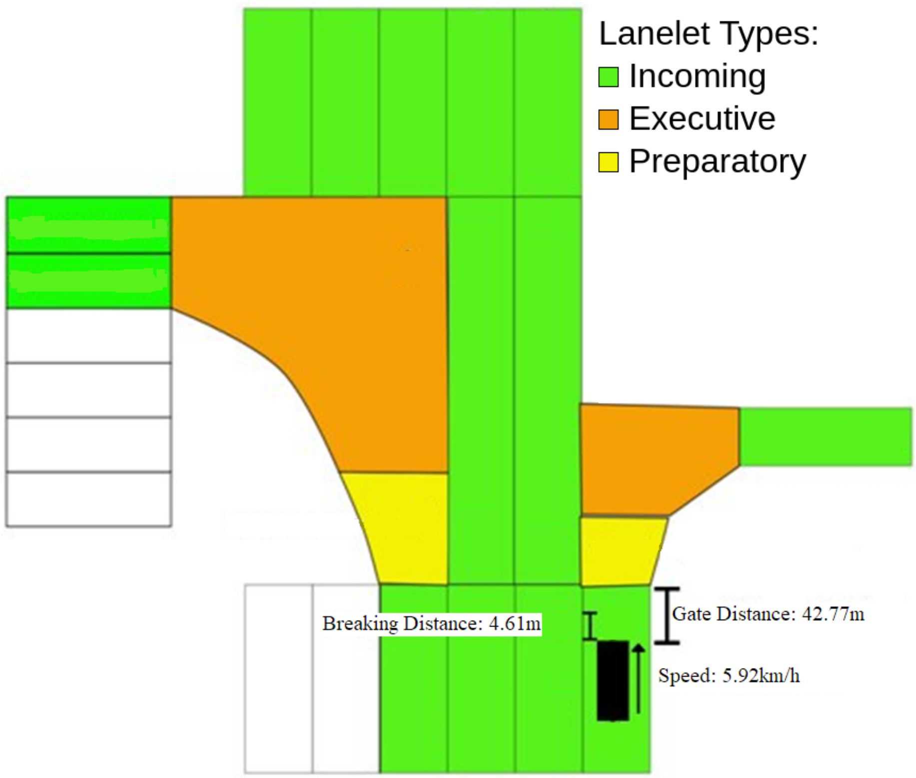


Figure 3.5: Yellow Light Stopping Threshold Example.

Yellow Light Violation Algorithm An algorithm (Yellow Light Violation Algorithm) for yellow light violation detection is as follows:

- **Line 5:** Iteration Through Vehicle States
  - Utilize a FOR loop to examine vehicle states (s_t) for each recorded moment in the trajectory information.
- **Line 8-9:** Yellow Light Encounter Check
  - Determine if the vehicle is currently encountering a yellow light (IF statement).

- Check if the previous moment's traffic light state was not yellow, indicating a new yellow light phase (IF statement).

- **Lines 10-13:** Parameters Definition
  - Define parameters such as speed, braking distance, reaction distance, and delta representing the distance from the stop line associated with the traffic light when it transitioned to yellow.
- **Line 14-15:** Intersection Entry and Stopping Capability Assessment
  - Verify if the vehicle has entered the intersection at the current state (IF statement).
  - Assess whether the sum of the vehicle's braking and reaction distance is less than or equal to the actual distance to the stop line when encountering a yellow light (IF statement).
  - The purpose is to check if the vehicle was capable of stopping but crossed the yellow light, and if so to flag a violation of the yellow light stopping rule. If the vehicle crosses the yellow light as required, the vehicle is considered as not having violated the rule.
- **Line 19:** No Intersection Crosses Were Encountered
  - Return that the vehicle did not violate the yellow light stop rule.

Yellow Light Violations: Determines if a vehicle failed to stop at a yellow light when it could have.

Input Data: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), Lanelet map ($M$)

Parameters: *vehicle_braking_acceleration*, *reaction_time*

Result: Returns a Boolean value representing whether a vehicle ran a yellow light.

entry_gate ← The intersection-entry gate of the vehicle along track S
exit_gate ← The intersection-exit gate of the vehicle along track S
gateway_traffic_light ← The light regulating the intersection between *entry_gate* and *exit_gate*
previous_light ← is undefined
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
    current_light_state ← Get the light state at the current vehicle state (s_t) for the traffic light *gateway_traffic_light*
    entered_intersection ← Check if the front of the vehicle at state s_t has reached the intersectional entry gate stop line
    if *current_light_state = Yellow* **then**
        if *previous_light is undefined OR previous_light* $\neq$ *Yellow* **then**
            speed ← gets the speed of the vehicle at state s_t
            braking_distance ← determine the braking distance of the vehicle by using *speed* and *vehicle_braking_acceleration*
            reaction_distance ← get the reaction-time distance from *speed* and *reaction_time*
            Δ ← determine the distance between the vehicle at state s_t and the closest stopline associated with lanelet map (M)
        if entered_intersection **then**
            if *braking_distance* + *reaction_distance* $\leq \Delta$ **then**
                return TRUE
            return FALSE
    previous_light ← *current_light_state*
return FALSE

Limitations The approach has limitations in considering specific vehicle orientations or lane geometry when calculating the stopping distance to the start of the intersection. One limitation is that the algorithm relies on the distance between a vehicle and the stop line associated with the intersection, rather than considering the distance to the intersection stop line and the vehicle along the lanelets on the path to the stop line. Additionally, the algorithm does not account for complex geometry, such as curves along the lanelet, which can impact visibility, braking, and reaction speed. These limitations may affect the accuracy of the stopping distance calculation in scenarios with intricate lane structures or curved pathways.

Permissive Left Turns at Green Lights

Permissive Left Turn Violations The permissive left turn verification rule evaluates whether a vehicle has violated permissive left-turning restrictions, specifically by obstructing another vehicle while making a left turn into the oncoming lane of the intersection during a green light interval along its route. Permissive left-turn intersections are characterized by allowing left turns without dedicated left-turn signal heads on the intersection traffic light [11]. In these intersections, drivers interpret the green light as an indication to proceed with a left turn. Vehicles executing permissive left turns must always yield to oncoming traffic. It's important to distinguish permissive left turns from protected left turns, which involve dedicated arrows signalling vehicles' permission to turn left [11].

Rule Role Permissive left turning is a maneuver allowed for vehicles transitioning from one-way roads to other one-way roads during a green-light signal interval. To execute a permissive left turn, a vehicle must come to a complete stop, ensuring that the way is clear of all traffic. Importantly, the vehicle should not impede the speed of oncoming traffic at the intersection. Once these conditions are met, the vehicle can proceed with a left turn through the intersection. The validation of permissive left turns involves assessing the time it takes for the reference vehicle to traverse from the start of the entry gate to the end of the exit gate of the intersection. This duration is then compared with the hypothetical time it would take for oncoming traffic to enter the intersection, considering their speed at the moment the reference vehicle entered. If an oncoming vehicle, based on its velocity, would have reached the entry gate before the reference vehicle exits the intersection, it indicates that the oncoming vehicle slowed down due to the presence of the reference vehicle. In a specific example (Figure 3.6), Vehicle 89 is found to commit a permissive left turn violation with Vehicle 28 in database 782 from the WISE Lab database.

Example The image (Figure 3.6) illustrates an example of a permissive left-turn violation:

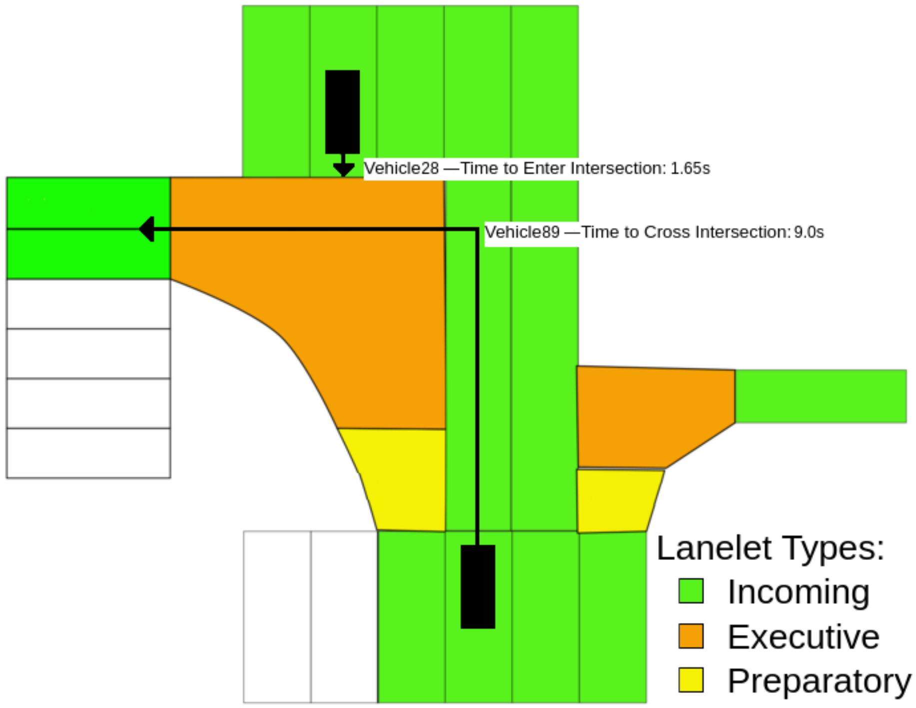


Figure 3.6: Example of a Permissive Left Turn Violation.

Permissive Green Light Violation Algorithm Algorithm (*Permissive Green Light Violation Algorithm*) for permissive left-turn violation detection is as follows:

- **Line 5:** Check every state in the vehicle trajectory.
- **Line 9:** Check if the vehicle has entered the intersection at the currently iterated state.
- **Lines 10 and 12:** Determine if there is a green light without a permissive left turn. Return a failure for the violation check if either line 10 or line 12 is evaluated as true, as it indicates that no permissive left turn could have occurred.
- **Line 14:** Determine if a permissive left turn has occurred with the vehicle and store the time the vehicle entered the intersection for the left turn as a parameter.
- **Line 16:** Determine if a vehicle has exited an intersection at the currently iterated state.
- **Line 17:** Determine the time between the vehicle's front entering the intersection and the vehicle's rear leaving the intersection.
- **Line 18:** Check the states for every other vehicle along the intersection.
- **Line 19:** Determine if the vehicle being iterated through (w_p) is oncoming to the reference vehicle exiting the intersection (s_t).

- **Line 20:** Determine the amount of time for the other vehicle (w_p) to reach the entry gate of the intersection, given its speed and position during the moment the reference vehicle (s_t) enters the intersection ($vehicle\_entry\_time$).
- **Line 21:** If the time for the other vehicle to enter the intersection is less than the time for the reference vehicle to finish the permissive left turn, it indicates that the reference vehicle takes the right of way from the other vehicle and forces it to yield along its trip. Return a pass for the rule violation on line 22.
- **Line 25:** If no such event occurs, the violation test fails, and a fail state is returned.

Permissive Green Light Violations: Determines if a vehicle failed to perform a permissive left turn at a green light, if applicable.

Input Data: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), Lanelet map ($M$), Vehicle recording ($R = (S_0, S_1, \dots)$)

Parameters: None

Result: Returns a Boolean value representing whether a vehicle violated a permissive left turn at a green light.

```
entry_gate ← The intersection-entry gate of the vehicle along trajectory S
exit_gate ← The intersection-exit gate of the vehicle along trajectory S
turn_direction ← Turn direction between entry_gate and exit_gate
gateway_traffic_light ← The light regulating the intersection between entry_gate and exit_gate
// Check every state for the reference vehicle
for every state s at timestamp t (s_t) in vehicle trajectory S do
    current_light_state ← Get the light state at the current vehicle state (s_t) for the traffic
      light gateway_traffic_light
    entered_intersection ← Check if the front of the vehicle at state s_t has reached the
      intersectional entry_gate stop line
    exited_intersection ← Check if the rear of the vehicle at state s_t has reached the intersection
      exit_gate stop line
    if entered_intersection then
        // Check if there is a green light and a left turn on a signalized
           permissive left-turn regulated intersection (S-PLTRI) gate
        if current_light_state = Green AND turn_direction is not a Left Turn then
            return FALSE
        else if current_light_state = Green AND turn_direction is a Left Turn AND
          entry_gate does not have a permissive left turn then
            return FALSE
        else if current_light_state = Green AND turn_direction is a Left Turn AND
          entry_gate has a permissive left turn then
            vehicle_entry_time ← t
    if exited_intersection then
        vehicle_crossing_time ← t - vehicle_entry_time
        // Check every vehicle state for all other vehicles in the recording
        for every state w at timestamp p (w_p) for every vehicle trajectory S_i in-vehicle recording
          R where S_i ≠ S do
            is_oncoming ← determine if the vehicle at state w_p is oncoming to the vehicle at
              state (s_t)
            if vehicle_entry_time = p AND is_oncoming then
                other_vehicle_entry_time ← determine the time for the front of a vehicle at
                  state w_p to reach the entry-gate stop-line based on its current velocity
                if other_vehicle_entry_time - vehicle_crossing_time < 0 then
                    return TRUE
    entered_intersection ← FALSE
    exited_intersection ← FALSE
return FALSE
```

Limitations A limitation of the approach is that it relies on the speed of oncoming traffic precisely when the reference vehicle enters the intersection by crossing the gate. If oncoming traffic happens to be moving slightly faster or slower during that specific moment, it can alter the prediction for the time it should take the oncoming vehicle to enter the intersection. An improvement to address this issue could involve calculating an average or median value for the speed of oncoming vehicles around the time the reference vehicle enters the intersection, providing a more robust estimate rather than relying on the speed at the exact moment of entry.

3.1.3 Turning After Stops at Unsignalized Stop-Sign Regulated Intersections (uS-SSRI)

The procedures for right-of-way priority when moving after stopping at uS-SSRIs is detailed in the following sections. In instances where multiple vehicles come to a stop at an intersection simultaneously, specific rules dictate the right-of-way for each vehicle and which must yield. However, a limitation of the current approach is its lack of flexibility in setting a time window to consider multiple vehicles as having stopped simultaneously. For example, if two vehicles come to a stop within a single frame of the recording, representing a fixed time window (e.g., 100ms), they are considered to have stopped roughly at the same time. This fixed time window may not accurately capture real-world scenarios where vehicles may genuinely stop simultaneously but within a slightly broader timeframe. Additionally, the system does not recognize vehicles facing the wrong way while stopped at the intersection stop line, which impacts the identification of incoming left turns or oncoming vehicles in relation to a reference vehicle. Another limitation is the assumption that lanelets are placed approximately ninety degrees from one another to form an intersection. If lanelets are arranged at different angles, left turns and oncoming vehicles to a reference vehicle may go undetected.

Illegal Right Before Left Movements

Right Before Left Movement Violations The *right-before-left movement priority* rule is formulated to evaluate if a vehicle has infringed the rule stipulating that the right-of-way should be given to the vehicle on the right when multiple vehicles come to a stop at or around the same time at an intersection. This rule also examines whether a vehicle has violated the rule for an intersection by proceeding before another vehicle that is already stopped along the intersection.

Rule Role If a vehicle is approaching from the right of a reference vehicle, meaning it is incoming left of the reference vehicle at the intersection, the approaching vehicle is granted the right-of-way, and the reference vehicle is required to yield the right-of-way.

Example In addition to the examples presented in Figure 4.5 and Figure 4.11, the image shown in Figure 3.7 illustrates a scenario where a reference vehicle at the bottom must yield to a vehicle situated to its right. This determination is based on both vehicles coming to a stop at the intersection at approximately the same moment in time. In the described scenario, the vehicle to the right has the flexibility to move left, right, or proceed straight, while the reference vehicle (bottom vehicle) intends to turn right. This situation remains applicable even if the bottom vehicle chooses to turn left or proceed straight, as depicted by the dashed lines in Figure Figure 3.7.

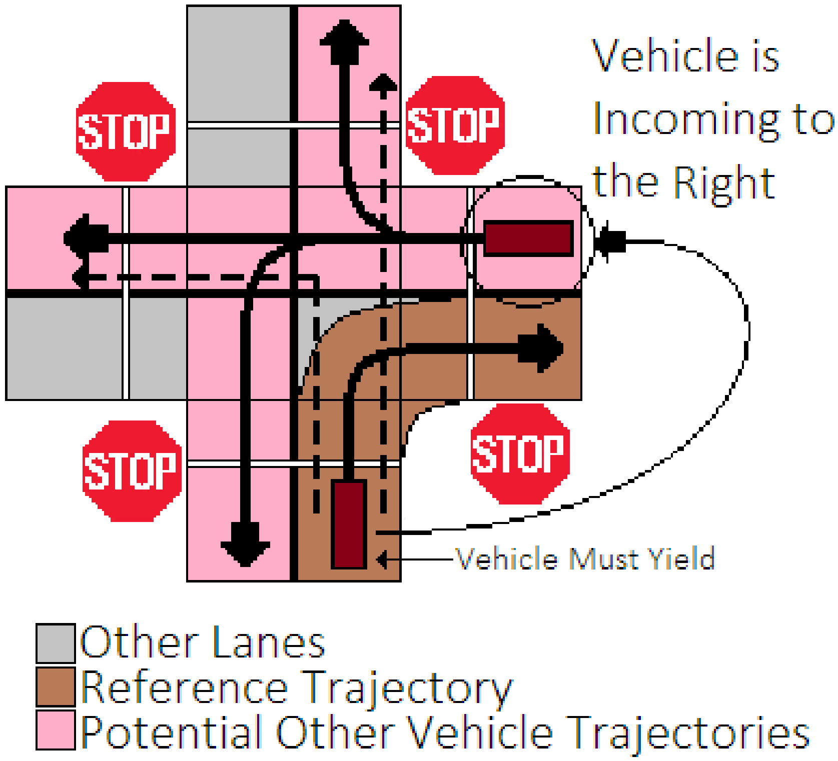


Figure 3.7: Example of a uS-SSRI where two vehicles stop at roughly the same time with one vehicle incoming to the right of the reference vehicle.

Right Before Left Movement Priority Algorithm The algorithm utilized for determining right-before-left rule violations is the *Right Before Left Movement Priority Algorithm*, and its functioning is detailed below:

- **Line 6:** Checks if a vehicle has come to a stop before a stop line and was not previously identified as having stopped.
- **Line 8:** The algorithm appends all other vehicles stopped at the intersection to a list. Additionally, any vehicles that are already stopped along the intersection are also appended to the list. As far as the rule is concerned, this implies that the current vehicle must yield to vehicles already stopped along the intersection, allowing them to proceed first.
- **Line 9:** Checks if the vehicle is currently in motion and if it was flagged as having been stopped before.

- **Line 11:** Iterates through all vehicles stopped at the intersection with the reference vehicle.
- **Line 12:** Iterates through all trajectory states associated with each vehicle one by one.
- **Line 16:** Determines if the lane associated with the other vehicle is to the right of the road related to the reference vehicle at the intersection.
- **Line 21:** Checks if there are no traffic lights regulating the intersection.
- **Line 25:** Determines if the reference vehicle has the same right-of-way priority as the other vehicle at the intersection, indicating that no unique signs modify the standard right-of-way transfer.
- **Line 26:** Checks if the other vehicle is stopped before the line and is along the same intersection, confirming that the lanes are connected and intersect with each other.
- **Line 27:** Returns a rule violation pass state if the reference vehicle is found to have started moving along the intersection before the other vehicle.
- **Line 28:** Returns a failure state for violation detection if no such violation is found.

Right Before Left Movement Priority Violation Check: Returns a Boolean value representing if the vehicle violated the right-of-way constraints after stopping at a uS-SSRI at the same time as another vehicle to its right.

Input Data: Distance threshold (θ_{Δ}), Speed threshold (θ_{speed})
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), Lanelet map ($M$), Vehicle trajectory recording ($R = (S_0, S_1, \dots)$)
Result: Returns if vehicle V violated the right-of-way over a uS-SSRI.

vehicle_interval_start ← FALSE
vehicles_in_way ← create an empty list
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
    speed ← get the vehicles speed at state s_t
    Δ ← gets the distance between the vehicle at state s_t and the closest stop line on lanelet map (M)
    if $speed \leq \theta_{speed}$ *AND* $\Delta \leq \theta_{\Delta}$ *AND* ¬vehicle_interval_start **then**
        vehicle_interval_start ← TRUE
        vehicles_in_way ← Append all vehicles in trajectory dataset R which are stopped along the same intersection as the vehicle at state s_t
    else if ($speed > \theta_{speed}$) *AND* vehicle_interval_start **then**
        vehicle_interval_start ← FALSE
        for *every vehicle* v_i *in vehicles_in_way* **do**
            for *state w at timestamp p* (w_p) *in* v_i **do**
                other_speed ← get the vehicles speed at state w_p
                reference_lanelet ← get the lanelet the vehicle is contained on at state s_t on lanelet map (M)
                other_lanelet ← get the lanelet the vehicle is contained on at state w_p on lanelet map (M)
                if *other_lanelet is to the right of reference_lanelet along the intersection* **then**
                    reference_movement_direction ← get the movement direction of the vehicle at state s_t across the intersection on lanelet map (M)
                    other_movement_direction ← get the movement direction of the vehicle at state w_p across the intersection on lanelet map (M)
                    same_priority ← check if the vehicle (s_t) on *reference_lanelet* with movement direction *reference_movement_direction* has the same right-of-way priority as the vehicle (w_p) on *other_lanelet* with movement direction *other_movement_direction*
                    relevant_traffic_lights ← TRUE
                    if *there are no traffic lights along both reference_lanelet and other_lanelet* **then**
                        relevant_traffic_lights ← FALSE
                    Δ_p ← determine the distance between the vehicle at state w_p and the closest stop line on lanelet map (M)
                    same_intersection ← determine if vehicles at states s_t and w_p are along the same intersection (either at the entry gate or inside the intersection, not exiting the intersection) on lanelet map (M)
                    if same_priority *AND* ¬relevant_traffic_lights **then**
                        if $other\_speed \leq \theta_{speed}$ *OR* $\Delta_p \leq \theta_{\Delta}$ *AND* same_intersection **then**
                            return TRUE
return FALSE

Limitations In addition to the limitations highlighted in the section introduction for moving after stops (Section 3.1.3), an additional consideration arises when vehicles approach an intersection with the intention to turn. In such scenarios, vehicles may exhibit a tilt towards the direction of their intended turn. When both the reference vehicle and another incoming vehicle to the right of the reference vehicle are significantly tilted, the system may misclassify them as oncoming rather than incoming to the right of the reference vehicle. This introduces a potential source of error in the determination of vehicle orientations and their respective movements at the intersection.

Illegal Intersection Priority Movements

Priority Movement Violations The *intersection priority movement* rule checks if a vehicle has violated lane priority when stopping along an intersection simultaneously with another vehicle.

Rule Role A vehicle is not allowed to enter an intersection if there is another vehicle with the right-of-way that will be endangered by it. A vehicle must wait for the right-of-way if it does not have priority. Priority rules differ by nation, and in some countries (i.e., Germany), lanes can have assigned priorities based on traffic regulation signs indicating movement priorities. The formulation allows specific lanes to hold higher priorities than other lanes based on traffic signs posted to specific lanelets. This rule also examines whether a vehicle has violated the rule for an intersection by proceeding before another vehicle that is already stopped along the intersection.

Example In addition to the example provided by Figure 4.6, the following scenario illustrates a situation where a one-time priority sign is used to signal that the bottom vehicle has priority over the vehicle to its right. In the described scenario, the vehicle to the right has the flexibility to move left, right, or proceed straight, while the reference vehicle (bottom vehicle) intends to turn right. This situation remains applicable even if the bottom vehicle chooses to turn left or proceed straight, as depicted by the dashed lines in Figure Figure 3.8.

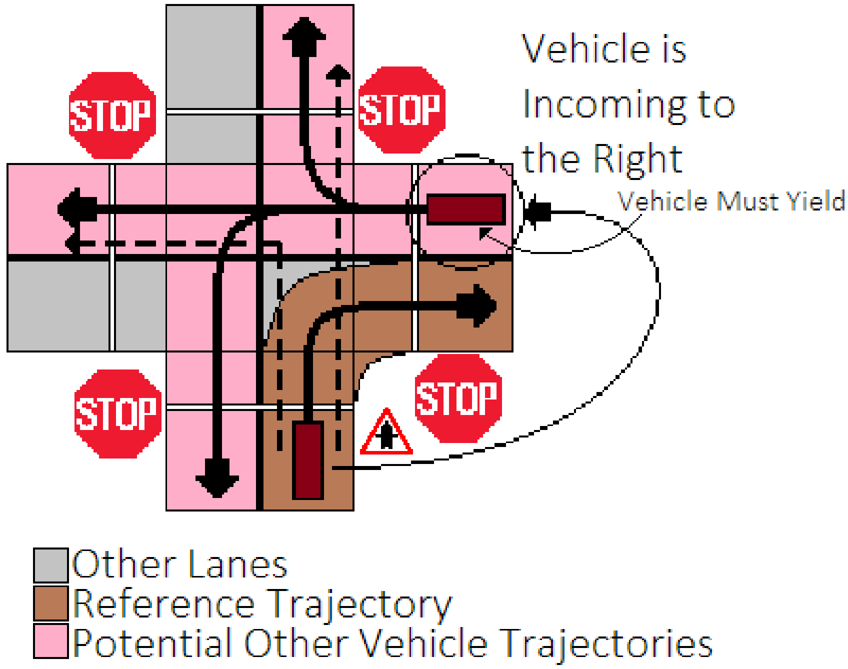


Figure 3.8: Example of a uS-SSRI where two vehicles stop at roughly the same time with the reference vehicle being granted intersectional right-of-way priority by a sign.

Intersection Priority Algorithm The following algorithm *Intersection Priority Algorithm* is used to determine if an uS-SSRI intersection-priority-movement rule violation has occurred:

- **Line 7:** The IF statement checks if a vehicle has stopped before a stop line and was previously not flagged as having stopped.
- **Line 10:** All other vehicles stopped along the intersection are appended to a list. Additionally, any vehicles that are already stopped along the intersection are also appended to the list. As far as the rule is concerned, this implies that the current vehicle must yield to vehicles already stopped along the intersection, allowing them to proceed first.
- **Line 11:** The ELSE IF statement determines if the vehicle is moving and if the vehicle was flagged as having been stopped before.
- **Lines 13 and 14:** Iterates through all vehicles stopped along the intersection with the reference vehicle and iterates through all trajectory states associated with the vehicles one by one.
- **Line 24:** Determines if the reference vehicle turns left and the other turns straight.
- **Line 25:** Determines if the other vehicle has special lane right-of-way priority over the reference vehicle if the vehicles are not oncoming to one another and if the other vehicle is either still stopped or before the line to enter the intersection.

- **Line 27:** Performs a similar check but for reference vehicles movement left and other vehicles movement right.
- **Line 30:** Reviews if the other vehicle has priority over the reference vehicle for every movement combination.
- **Lines 26, 29, and 31:** Return a pass for rule violation detection since the reference vehicle moved first down the intersection without letting a vehicle with special right-of-way privileges go first.
- **Line 32:** Returns that no rule violations were detected.

Intersection Priority Movement Violation Check: Returns a Boolean value representing whether the vehicle violated the right-of-way constraints after stopping at a uS-SSRI at the same time as another vehicle on its oncoming lane. The function uses priority checking to determine right-of-way violations in nations where lanes can be assigned priority over one another.

Input Data: Distance threshold (θ_{Δ}), Speed threshold (θ_{speed})
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), Lanelet map ($M$), Vehicle recording ($R = (S_0, S_1, \dots)$)
Result: Returns if vehicle V violated the right-of-way over a uS-SSRI.

```
starting_interval ← FALSE
vehicles_in_way ← create an empty list
vehicle_must_stop ← FALSE
for every state s at timestamp t (s_t) in vehicle trajectory S do
    speed ← get the vehicles speed at state s_t
    Δ ← determine the distance between the vehicle at state w_p and the closest stop line on lanelet map (M)
    if speed ≤ θ_speed AND Δ ≤ θ_Δ AND ¬starting_interval then
        starting_interval ← TRUE
        starting_time ← t
        vehicles_in_way ← Append all vehicles in trajectory dataset R which are stopped along the same intersection as the vehicle at state s_t
    else if speed > θ_speed AND starting_interval then
        starting_interval ← FALSE
        for every vehicle v_i in vehicles_in_way do
            for vehicle state w at timestamp p (w_p) in trajectories v_i do
                other_vehicle_speed ← get the vehicles speed at state s_t
                reference_lanelet ← get the lanelet the vehicle is contained on at state s_t on lanelet map (M)
                other_lanelet ← get the lanelet the vehicle is contained on at state w_p on lanelet map (M)
                reference_movement_direction ← get the movement direction of the vehicle at state s_t across the intersection on lanelet map (M)
                other_movement_direction ← get the movement direction of the vehicle at state w_p across the intersection on lanelet map (M)
                has_priority ← determine if the vehicle (w_p) with movement direction other_movement_direction on lanelet other_lanelet has priority over the vehicle (s_t) with movement direction reference_movement_direction on lanelet reference_lanelet
                oncoming_traffic ← determine if the vehicle at state s_t is oncoming to the vehicle at state w_p
                Δ_p ← determine the distance between the vehicle at state w_p and the closest stop line on lanelet map (M)
                same_intersection ← determine if vehicles at states s_t and w_p are along the same intersection (either at the entry gate or inside the intersection, not exiting the intersection) on lanelet map (M)
                if reference_vehicle_turn is Left AND other_vehicle_turn is Straight then
                    if has_priority AND ¬oncoming_traffic AND other_vehicle_speed ≤ θ_speed OR Δ_p ≤ θ_Δ AND same_intersection then
                        return TRUE
                else if reference_vehicle_turn is Left AND other_vehicle_turn is Right then
                    if has_priority AND ¬oncoming_traffic AND other_vehicle_speed ≤ θ_speed OR Δ_p ≤ θ_Δ AND same_intersection then
                        return TRUE
                else if has_priority then
                    return TRUE
return FALSE
```

Limitations The approach presents a limitation in that street signs must be directly referenced to their corresponding lanelets for evaluation when comparing the priority of different lanelets along an intersection. Additionally, variations in rules across different countries, particularly concerning one-time priority pass signs or signs altering the priority of stop-regulated intersections, pose a challenge. Implementing different system configurations for the *has_priority* function becomes necessary to accommodate diverse rule systems across countries and ensure accurate determination of intersectional priority.

Left Turning Priority

Left Turn Violations The *intersection left turn* rule checks if a vehicle has violated left-turning priority when turning left down an intersection after stopping at the same moment as another vehicle across the intersection in the reference vehicle's oncoming lane.

Rule Role A left-turning ego vehicle does not have priority over vehicles in their oncoming lane across an intersection. The left-turning ego vehicle obtains the right-of-way for the intersection once the oncoming vehicle crosses the intersection. This rule also examines whether a vehicle has violated the rule for an intersection by proceeding before another vehicle that is already stopped along the intersection.

Example In addition to the examples provided by Figure 4.7 and Figure 4.12, the following example (Figure 3.9) illustrates a situation where a vehicle turning left must yield to a vehicle in its oncoming lane across the intersection.

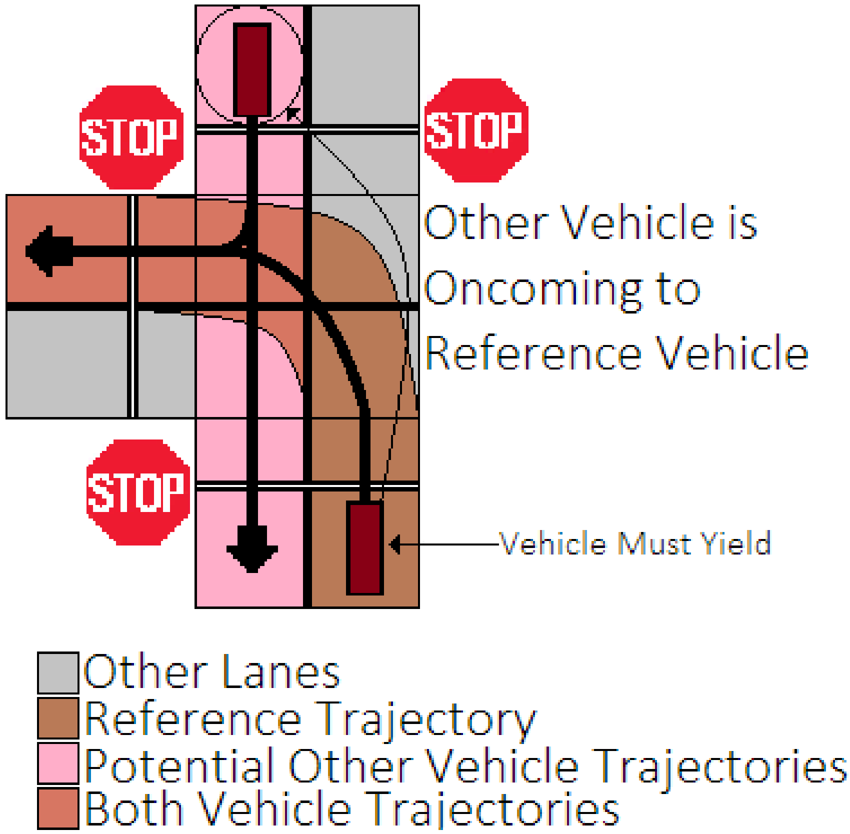


Figure 3.9: Example of a uS-SSRI where two vehicles stop at roughly the same time with one vehicle oncoming to a left-turning reference vehicle.

Left Turns Algorithm The following algorithm is used to check if a left-turn priority violation has occurred:

- **Line 7:** Determine if the reference vehicle at the current state (s_t) has stopped before the stop line and was previously not stopped.
- **Line 9:** Append all vehicles along the intersection stop line that are also stopped to a list of vehicles. Additionally, any vehicles that are already stopped along the intersection are also appended to the list. As far as the rule is concerned, this implies that the current vehicle must yield to vehicles already stopped along the intersection, allowing them to proceed first.
- **Line 10:** Check if the vehicle has started moving again.
- **Lines 12 and 13:** Check every vehicle with its corresponding trajectory states in the list of vehicles stopped along the intersection after the vehicle has started moving again.
- **Line 18:** Determine if the other vehicle (w_p) is oncoming to the reference vehicle (s_t).

- **Line 23:** Determine if the reference vehicle is turning left and the other is moving straight.
- **Line 26:** Check if the other vehicle does not have special right-of-way priorities over the reference vehicle and if the other vehicle has stopped before the stop line along the same intersection as the reference vehicle.
- **Line 28:** Check for a left-turning reference vehicle and a right-turning other vehicle.
- **Lines 27 and 30:** Return a pass for the rule violation detection since the reference vehicle has turned left down an intersection without letting an oncoming vehicle stop along the intersection pass first.
- **Line 31:** Evaluate a failure since the reference vehicle was not found to have violated the rule in the previous loop.

Left Turn Violation Check: Returns a Boolean value representing if the vehicle violated the right-of-way constraints after stopping at a uS-SSRI at the same time as another vehicle on its oncoming lane.

Input Data: Distance threshold (θ_{Δ}), Speed threshold (θ_{speed})

Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), Lanelet map ($M$), Vehicle recording ($R = (S_0, S_1, \dots)$)

Result: Returns if vehicle V violated the right-of-way over a uS-SSRI.

start_interval ← FALSE
vehicles_in_way ← create an empty list
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
  speed ← get the speed of the vehicle at the state s_t
  vehicle_point ← finds the nearest lanelet to vehicle at state s_t on lanelet map (M)
  Δ ← gets the distance between the vehicle at state s_t and the stopline on lanelet map (M)
  if speed $\leq \theta_{speed}$ *AND* $\Delta \leq \theta_{\Delta}$ *AND* ¬start_interval **then**
    start_interval ← TRUE
    vehicles_in_way ← Append all vehicles in trajectory dataset R which are stopped along the same intersection as the vehicle at state s_t
  else if *speed* $> \theta_{speed}$ *AND* start_interval **then**
    start_interval ← FALSE
    for *every vehicle* v_i *in vehicles_in_way* **do**
      for *vehicle state w at timestamp p* (w_p) *in* v_i **do**
        other_vehicle_speed ← get the vehicles speed at state s_t
        reference_lanelet ← get the lanelet the vehicle is contained on at state s_t on lanelet map (M)
        other_lanelet ← get the lanelet the vehicle is contained on at state w_p on lanelet map (M)
        oncoming_traffic ← determine if the vehicle at state s_t is oncoming to the vehicle at state w_p
        if oncoming_traffic **then**
          skip_checking ← FALSE
          reference_turn_direction ← get the turn direction of the vehicle at state s_t across the intersection on lanelet map (M)
          other_turn_direction ← get the turn direction of the vehicle at state w_p across the intersection on lanelet map (M)
          has_priority ← determine if the vehicle (w_p) with turn direction *other_turn_direction* on lanelet *other_lanelet* has priority over the vehicle (s_t) with turn direction *reference_turn_direction* on lanelet *reference_lanelet*
          if *reference_vehicle_turn is Left AND other_vehicle_turn is Straight* **then**
            Δ_p ← determine the distance between the vehicle at state w_p and the closest stop line on lanelet map (M)
            same_intersection ← determine if vehicles at states s_t and w_p are along the same intersection (either at the entry gate or inside the intersection, not exiting the intersection) on lanelet map (M)
            if ¬has_priority *AND other_vehicle_speed* $\leq \theta_{speed}$ *OR* $\Delta_p \leq \theta_{\Delta}$ *AND* same_intersection **then**
              return TRUE
          else if *reference_vehicle_turn is Left AND other_vehicle_turn is Right* **then**
            if ¬has_priority *AND other_vehicle_speed* $\leq \theta_{speed}$ *OR* $\Delta_p \leq \theta_{\Delta}$ *AND* same_intersection **then**
              return TRUE
return FALSE

Limitations The approach has a limitation as it assumes that oncoming vehicles along an intersection are facing each other, suggesting that left turns cut off oncoming traffic. This assumption may not be valid in scenarios where intersections are slanted or do not align directly through the intersection center, leading to an oncoming lane positioned to the right of a vehicle along an intersection. In such cases, the logic governing left-turning rules would require adjustment, as a left-turning vehicle may not necessarily cut off the oncoming lane while making a turn across the intersection.

3.2 Non-Intersectional Traffic Concepts

Non-intersectional traffic concepts encompass traffic rule violations and maneuver detection methods that are unrelated to intersection right-of-way priority or signal violations. These concepts are applicable to non-junction-related traffic systems and include various detection methods:

1. **Speed Limit Violation Detection:** Calculates speed limit violation rates as a proportion of the total time spent driving for vehicles passing through the intersection.
2. **Turn Direction Detection:** Determining the direction in which a vehicle is turning, whether left or right.
3. **Waypoint Detection:** Identifying specific points or locations along a route where a vehicle should perform a particular action or maneuver.
4. **Collision Detection:** Detecting potential collisions or instances where vehicles may be on a collision course.
5. **Target Speed Violation Detection:** Detecting instances where a vehicle exceeds or falls below the prescribed speed limits.
6. **Offroad Detection:** Identifying when a vehicle deviates from the designated roadway onto non-road surfaces.
7. **No Offroad With Stop Sign Checking:** Determining if a vehicle failed to stop before a stop sign while on the road.
8. **Target Speed With Waypoints:** Determining if a vehicle failed to maintain a target speed within proximity of a waypoint.
9. **Tailgating Detection:** Identifying vehicles that follow too closely behind one another, violating safe following distances.

These non-intersectional traffic concepts contribute to enhancing overall traffic safety and adherence to traffic rules in various scenarios outside of intersections.

3.2.1 Speed Limits

Speed Limit Violations The *speed limit* rule assesses whether a vehicle exceeds the posted speed limit for a specific lanelet during a given trip.

Rule Role The rule examines the vehicle's position to identify the lanelet it is on and its associated speed limit. It then evaluates whether the vehicle exceeds the speed limit specified for that lanelet.

Example The image (Figure 3.10) illustrates an example of a vehicle flagging a speed limit violation. As can be seen, a speed limit violation for Vehicle 13 (**red circle**) is encountered for a trip between timestamps 11800ms-14800ms in the dataset *vehicle_ tracks_ 000* for the *DR_ USA_ Intersection_ MA* map:

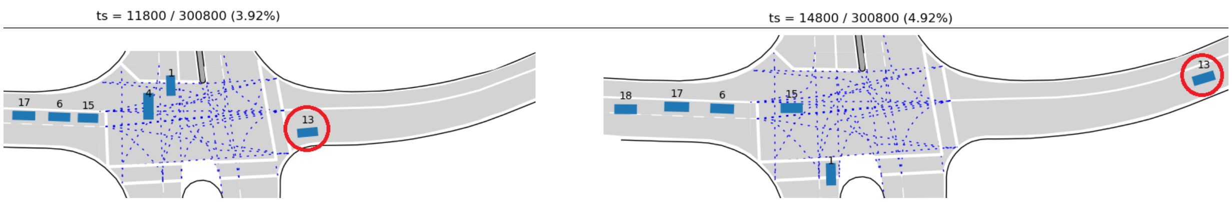


Figure 3.10: An example of a speed limit violation (**red circle**).

Speed Limits Algorithm The Speed Limits Algorithm (Speed Limits Algorithm) is employed to evaluate the percentage of time a vehicle has committed speed limit violations during its journey through the intersection.

Speed Limit Violation Check: Determines if a vehicle is braking lanelet speed limits for every moment it was recorded during its trip.

Input Data: Speed threshold (θ_{speed}), Speed-epsilon ϵ
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \ldots$), Lanelet map ($M$)
Result: A rational number representing the percentage of trajectory-states where the vehicle was driving (i.e. not stopped) and surpassed the speed limit plus the error epsilon.

list ← create an empty list
`// Counter for the number of states where the vehicle is moving`
moving ← 0
`// Counter for the number of states where a speed violation occurs`
violation ← 0
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
  speed_limit ← obtain the speed limit of the current lanelet occupied by the vehicle at state s_t on lanelet map (M), in terms of meters-per-second
  speed ← get vehicle's speed in state s_t
  `// Check if the vehicle is moving`
  if $speed > \theta_{speed}$ **then**
    moving ← add 1 to the counter
    `// Check if the current speed is greater than the speed limit`
    if $speed\_limit - speed + \epsilon < 0$ **then**
      violation ← add 1 to the counter

return $(\frac{violation}{moving})$

Limitations A limitation of the approach is that it evaluates speed limit violations moment by moment based on single frames of data from the interaction dataset. This means that a vehicle is considered to violate the speed limit if, in a specific 100ms interval, it exceeds the speed limit plus its corresponding error delta value. Speed limit violation checking using real-time data instead of individual frames, or employing statistical methods (such as the Box–Jenkins method or the Jackknife method) to average out groups of frames and identify changes in data over time, could enhance the detection of true instances of speeding rather than momentary occurrences. However, isolated instances of speeding within such short time frames may not necessarily constitute a speed limit violation in reality. Moreover, certain segments of a lane in many intersections may allow for higher speeds without posing a safety risk. Considering traffic flow and the speed of nearby vehicles can also influence a reference vehicle's speed for safety reasons. Incorporating information about nearby traffic flow to determine if a vehicle is speeding relative to other vehicles would contribute to a more accurate assessment of real speed limit violations.

3.2.2 Turn Direction

Turning directions at intersections can be determined by comparing the yaw angles of vehicles. Yaw angles play a crucial role in identifying whether a vehicle is turning left or right concerning another vehicle at an intersection. Additionally, they help ascertain whether another vehicle is oncoming to a reference vehicle. Yaw angles provide valuable information about the rotational movement of vehicles, aiding in the analysis of their turning behaviour and interactions at intersections.

Turn Direction Finder The *turn direction checking rule* is responsible for determining the turning direction of a vehicle as it travels across an intersection.

Rule Role Vehicle yaw angles play a crucial role in determining the change in orientation as a vehicle navigates through intersections. By comparing the yaw angles at the beginning and end of an intersection, it becomes possible to calculate the total difference in orientation. This information is then utilized to ascertain the specific turn direction of the vehicle, providing valuable insights into its maneuver at the intersection.

Example The images Figure 3.11 and Figure 3.12) visually illustrate the process of determining a vehicle's turn direction by measuring its orientation. The first image (Figure 3.11) conveys the same information as the first but presents orientation values in radians rather than yaw. The second image ((Figure 3.12) demonstrates the application of turn direction to determine whether lanelets are oncoming or incoming to each other. This involves comparing the orientations of two vehicles to establish if their respective lanelets are facing each other or if one lanelet is positioned to the right of the other.

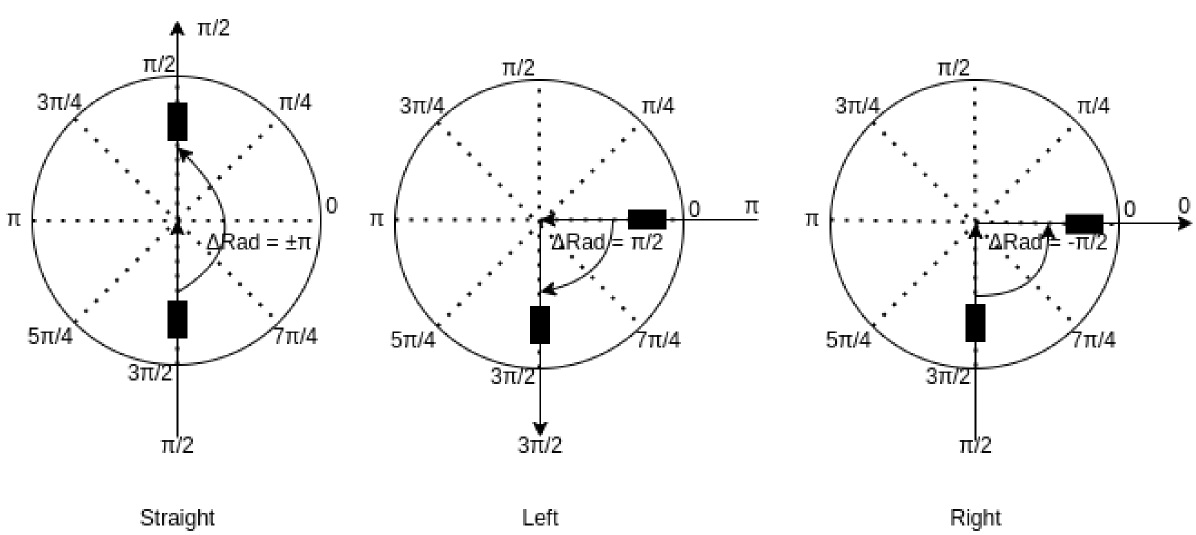


Figure 3.11: Determining the turn direction of a vehicle using orientation (yaw) measured in radians.

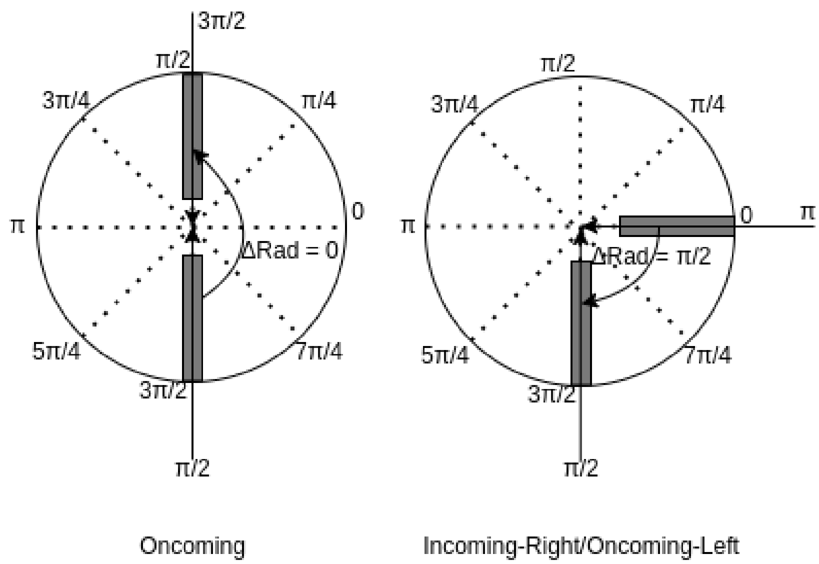


Figure 3.12: Illustration of lanelet geometry checks using reported vehicle yaw values.

Turn Direction Algorithm The following algorithm (Turn Direction Algorithm) is used to check a vehicle turn direction between lanelets:

Turn Direction Detection: Determines the turn direction of a vehicle from its change in yaw.

Input Data: Starting yaw (θ_i), Ending yaw (θ_f)
Parameters: None
Result: Returns the direction of a vehicle turn based on a vehicle-reported change in yaw.

radian_yaw_start ← Convert θ_i from yaw to radians
radian_yaw_end ← Convert θ_f from yaw to radians
`// Check if the vehicle is making a left turn by looking for a counter-clockwise rotation`
if *(radian_yaw_end - radian_yaw_start* $> \frac{2\pi}{9}$ *AND radian_yaw_end - radian_yaw_start* $< \frac{13\pi}{18}$*) OR (radian_yaw_end - radian_yaw_start* $< -\frac{5\pi}{4}$ *AND radian_yaw_end - radian_yaw_start* $> -\frac{7\pi}{4}$*)* **then**
    return LEFT
`// Check if the vehicle is making a left turn by looking for a clockwise rotation`
else if *(radian_yaw_end - radian_yaw_start* $< -\frac{2\pi}{9}$ *AND radian_yaw_end - radian_yaw_start* $> -\frac{13\pi}{18}$*) OR (radian_yaw_end - radian_yaw_start* $> \frac{23\pi}{18}$ *AND radian_yaw_end - radian_yaw_start* $< \frac{16\pi}{9}$ **then**
    return RIGHT
`// Check if the vehicle is moving straight by seeing if it did not rotate within a margin of error`
else if *((radian_yaw_end - radian_yaw_start) % 2π)* $< \frac{2\pi}{9}$ *OR ((radian_yaw_end - radian_yaw_start) % 2π)* $> \frac{16\pi}{9}$ **then**
    return STRAIGHT
return OTHER

Limitations The approach has a limitation in determining turn direction based on the difference in orientation from the start to the end of an intersection. This limitation arises from the fact that very tight turns or illegal U-turns may result in an orientation change that falls below the required threshold for recognizing a left or right turn. Additionally, defining precise criteria for what constitutes a left, right, or straight movement through an intersection can be challenging. Straight movements may not be perfectly straight in cases where an intersection is slanted diagonally, and the interpretation of movements as left or right turns can be subjective.

3.2.3 Waypoints

Waypoint Violations The *waypoint* rule is designed to ascertain whether a vehicle has reached a specific waypoint while navigating through an intersection.

Rule Role The waypoint rule examines the vehicle's position at every moment captured in its recording to verify whether it has closely approached a predefined waypoint within the specified radius of an epsilon threshold. If the vehicle remains distant from the waypoint, a failure state is appended to the results list for every recorded moment within the intersection. Conversely, if the vehicle successfully reaches the waypoint at any given moment in the recording, a success state is appended to the list of data structures. The determination of whether the vehicle has arrived at the waypoint during its journey is based on the presence of at least one success state within the list of data structures.

Example The image (Figure 3.13) provides an illustration of a vehicle satisfying the waypoint condition. The image features a waypoint represented by a hollow black circle on the map with a radius of 12 meters, covering the entirety of the intersection. Notably, vehicles that bypass the intersection (given the presence of two junctions on the map) are considered to have violated the waypoint. It's important to highlight that the waypoint rule assesses all vehicles passing through the intersection, rather than solely those navigating the roundabout. (Note: The image is a snapshot from the *USA_Roundabout_EP* map within the interaction dataset [5]).

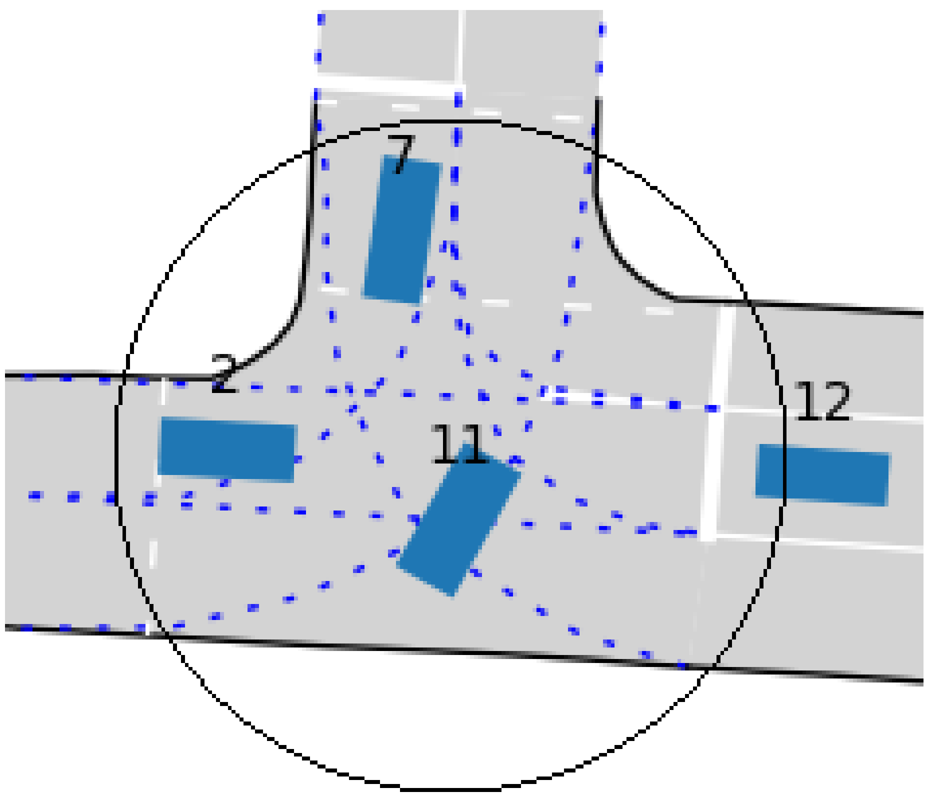


Figure 3.13: Waypoint example (**hollow black circle**).

Waypoints Algorithm The following algorithm (Waypoints Algorithm) is utilized to determine if a vehicle satisfies a waypoint requirement throughout its journey:

Waypoint Violation Check: Determines if a vehicle is within proximity of a waypoint or not at any given time.

Input Data: *waypoint_position*, Waypoint-epsilon (ϵ_W)
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$)
Result: Obtain a list representing moments where the vehicle was within an epsilon-based proximity of the given waypoint.

list ← create an empty list
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
    Δ ← get the distance between the vehicle at state s_t and *waypoint_position*
    if $\Delta < \epsilon$ **then**
        `// Report that the vehicle has made it to its waypoint at the given state s`
        list ← append the current time and a pass-state-flag s_t to the given list
return *list*

Limitations The current method lacks consideration for instances where vehicles briefly pass by a waypoint in a single moment within the recording. To address this limitation, a potential enhancement involves evaluating the duration during which the vehicle remains away from the waypoint in the recording. By comparing this off-waypoint duration with the time the vehicle spends at the waypoint, one can assess whether the vehicle satisfies the required duration criteria at the waypoint. Additionally, users have the option to count the number of frames in which a vehicle maintains proximity to a waypoint, providing a more comprehensive analysis of whether the vehicle meets the specified duration conditions for waypoint fulfillment. This adjustment aims to improve the accuracy and reliability of waypoint violation detection, especially in scenarios involving brief encounters with waypoints. Thorough testing across diverse scenarios and datasets is recommended to validate the effectiveness of this refined approach.

3.2.4 Collisions

Collision Violations The *collision* rule evaluates whether a vehicle is in close proximity to a collision waypoint throughout its journey within an intersection.

Rule Role A collision waypoint is defined by a specific point along with an epsilon value representing the radius around that point. The rule checks whether a vehicle, based on its length, comes close to the specified point which represents a static object in the intersection area. If a vehicle is in proximity to the point, a failure state is added to a list of structs.

Example The image (Figure 3.14) provides an example of a vehicle triggering a collision violation, as indicated by the black circle. It's important to note that this image is a snapshot from the *USA_Roundabout_EP* map in the interaction dataset [5]:

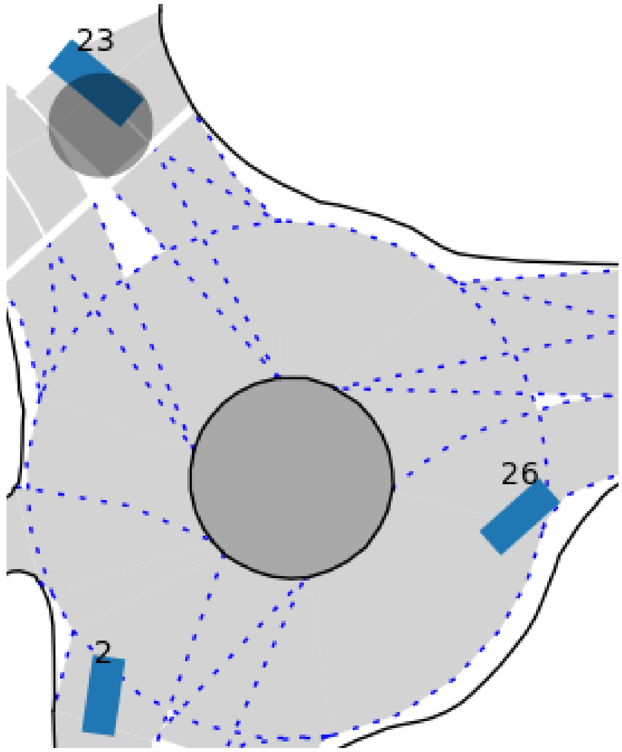


Figure 3.14: A vehicle encountering a collision (**see black circle**) waypoint and triggering a rule violation.

Collisions Algorithm The following algorithm (Collisions Algorithm) is employed to assess whether a vehicle is in violation of a collision rule throughout its journey:

Collision Check: Determines if a vehicle has encountered a collision with a restricted area.

Input Data: Collision-point (P), Collision-epsilon (ϵ)
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \ldots$), *vehicle_length*
Result: Returns moments where a vehicle was in bounds of a collision area.

list ← create an empty list
for *every state s at timestamp t (s_t) in vehicle trajectory S* **do**
  front_point ← obtain the front point of vehicle at state s_t given its yaw, *vehicle_length*, and its current position
  rear_point ← obtain the rear point of vehicle at state s_t given its yaw, *vehicle_length*, and its current position
  Δ_f ← determine the distance between the collision point at the vehicles *front_point*
  Δ_b ← determine the distance between the collision point at the vehicles *rear_point*
  if $min(\Delta_f, \Delta_b) \leq \epsilon$ **then**
    list ← append the current time s_t to the given list
return *list*

Limitations Similar to the waypoint limitations, a vehicle is deemed to have violated a collision rule even if it briefly enters the collision region. To address this, one can count the number of frames in which the vehicle violates the rule and compare it with the total duration of the recording or a predefined threshold for determining a collision rule violation. The collision detection system currently in place does not incorporate dynamic objects and solely utilizes points to represent static objects. To accommodate moving objects, additional implementation for a dynamic collision detection system would be necessary.

3.2.5 Target Speeds

Target Speed Violations The *target speed* rule assesses whether a vehicle has violated a predetermined target speed by exceeding the specified rate by a certain epsilon value.

Rule Role Two modes are available: one treats the target speed as a maximum speed limit that the vehicle must not exceed, while the other interprets the target speed as a speed the vehicle should adhere to without deviation. If a vehicle's speed is less than or equal to the target speed plus the epsilon value in the maximum speed limit mode, it is not deemed to violate the target top speed at that specific moment in the recording data. In the target speed limit mode, a vehicle is considered within the target speed limit if its speed falls within the range of the target speed and the epsilon value.

Example The figure (Figure 3.15) illustrates an instance where Vehicle 62 (red square) fails to adhere to the target speed with waypoint condition. The vehicle does not maintain a target speed of $5m/s \pm 4m/s$ over the waypoint (**depicted by the black circle**) across the intersection.

Figure 3.15: An illustration of a vehicle (**red square**) failing to maintain a target speed over a waypoint (**black circle**).

Target Speed Algorithm The following algorithm (Target Speed Algorithm) is utilized to examine whether a vehicle violates a target speed rule throughout the duration of a given recording:

Target Speed Violation Check: Determines if a vehicle has violated a target speed threshold at all given moments in its trip recording, based on either checking if a maximal speed limit or a target speed limit are being satisfied within an error ϵ-value.

Input Data: *target_speed*, Target-speed-epsilon (ϵ), *speed_checking_mode*
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$)
Result: A rational number representing the percentage of trajectory-states where the vehicle violated a preset target speed within an error epsilon.

`// Contains the number of time-frames evaluated`
frames ← 0
`// Contains the number of target speed violation occurrences`
speed_violation ← 0
for *every state s at timestamp t (s_t) in vehicle trajectory S* **do**
  speed ← get vehicle's speed in state s_t
  velocity_difference ← *target_speed* - *speed*
  total_speed_difference ← obtain the absolute value of *velocity_difference*
  frames ← add 1
  if *(total_speed_difference* $<\epsilon$ *AND speed_checking_mode is set to check if the speed is within the target speed limit) OR (velocity_difference+* $\epsilon < 0$ *AND speed_checking_mode is set to check if the speed is surpassing a maximum speed limit)* **then**
    speed_violation ← add 1
return ($\frac{speed\_violation}{frames}$)

Limitations Similar to previous limitations, the approach considers rule violations on a frame-by-frame basis, potentially flagging a violation for a single moment in time. To address this, users can assess the ratio of moments when the vehicle violates the target speed to the total moments the vehicle is in motion, providing a proportion of the time the rule is breached. Additionally, the rule does not account for the influence of other vehicles on the road. A vehicle might need to exceed the speed limit momentarily to avoid obstructing nearby traffic, leading to a false flag for rule violation.

3.2.6 No Offroad

No Offroad Violations The *no offroad* rule checks whether a vehicle has deviated from the designated road surface during its journey through a specific intersection. If the vehicle strays off the road, a violation of the rule is flagged.

Rule Role The offroad violation checker offers various modes to assess whether a vehicle deviates from the road surface. Users can examine the corners of a vehicle at distances of 50%, 80%, 90%, or 100% from its centerpoint to determine if these corners are offroad at a given moment. Alternatively, users can focus on the vehicle's centerpoint alone to assess offroad status, without considering its length or width. The Lanelet2 *search* method is utilized to determine if a point falls within any designated lanelet. If the resulting list of lanelets is empty, it indicates that the point is not situated on the road as defined by the OpenStreetMap file.

Example The image (Figure 3.16) depicts a situation where a vehicle is involved in an offroad violation. Please be aware that this image is taken from the *USA_Roundabout_MA* map within the interaction dataset. [5]:

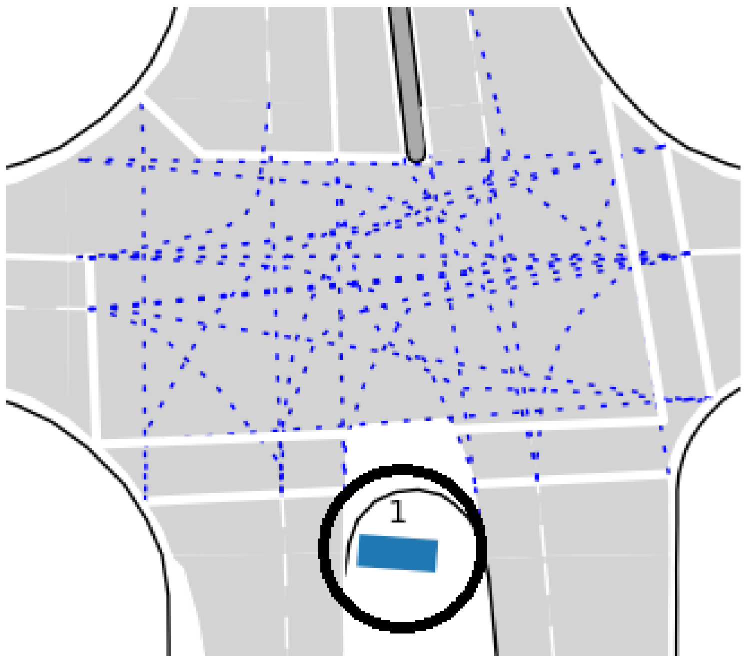


Figure 3.16: Example of a vehicle engaging in an offroad violation.

Offroad Violations Algorithm The following algorithm (Offroad Violations Algorithm) is implemented to assess whether a vehicle is violating offroad restrictions throughout its journey:

Offroad Violations Check: Checks if vehicles are engaged in offroad occurrences for the duration of a vehicle's trip by checking the corners of a vehicle by using its length, width, yaw, and a bounds constraint for the distance to travel from the vehicle's centerpoint to its corners.

Input Data: Bounds (B)

Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), *vehicle_length*, *vehicle_width*, Lanelet map (M)

Result: A rational number representing the percentage of trajectory-states where the vehicle was offroad.

```
// Contains the number of time-frames evaluated
frames ← 0
// Contains the number of offroad occurrences
offroad_occurance ← 0
// Effective length and width of the vehicle given the bounds (B) value
effective_vehicle_length ← vehicle_length * B
effective_vehicle_width ← vehicle_width * B
for every state s at timestamp t (s_t) in vehicle trajectory S do
    top_left ← gets the top left corner of the vehicle at state s_t given its yaw,
      effective_vehicle_length, and effective_vehicle_width
    bottom_left ← gets the bottom left corner of the vehicle at state s_t given its yaw,
      effective_vehicle_length, and effective_vehicle_width
    bottom_right ← gets the bottom right corner of the vehicle at state s_t given its yaw,
      effective_vehicle_length, and effective_vehicle_width
    top_right ← gets the top right corner of the vehicle at state s_t given its yaw,
      effective_vehicle_length, and effective_vehicle_width
    frames ← add 1
    if points top_left, bottom_left, bottom_right, and top_right are not within any
      lanelet on lanelet map (M) then
        offroad_occurance ← add 1
return (offroad_occurance / frames)
```

Limitations A limitation of the approach is that the search method may erroneously flag points as not located on any lanelet, even when they are. This issue can arise due to small gaps between lanelets and their bounding boxes, especially in areas between separate lanes on a road. One potential remedy is to define a single, large area covering the entire intersection. If a vehicle is within the intersection, it should at least register within the intersection area when using the search method. Another limitation is that brief offroad occurrences for a single frame are considered offroad violations, even if they occurred so briefly that they may have been caused by a sensor error in the recording equipment. To address this, one could consider the total time of the recording and compare the time spent on offroad violations with the overall trip duration. By comparing a vehicle's time spent offroad with a baseline average, users could determine whether an offroad violation occurred. Additionally, if certain road segments (such as streets, parking lots, or driveways) are not included in the map, offroad violations may be triggered inaccurately.

3.2.7 Target Speed With Waypoints

Target Speed With Waypoint Violations The *target speed with waypoint* rule assesses whether a vehicle has met the condition of reaching a waypoint while maintaining a designated target speed.

Rule Role The rule checks whether a vehicle has fulfilled the requirement to reach a specified target speed, within an epsilon threshold, at a designated waypoint, within an epsilon range. Two methods are available for speed checking: the first treats the target speed as a maximum speed limit, and the second considers the target speed as a range. In the first method, the vehicle must be below the maximum speed limit plus its epsilon value. In the second method, the vehicle must be within a range specified by an epsilon value of its target speed. The system can determine if a vehicle maintains a target speed over a specific area, stops before a stop-line in a construction zone, or is parked in a parking lot tile.

Example The image (Figure 3.17) depicts two instances of target speed limit violations. In the bottom row, vehicle 27 (circled in red) exceeded the user-set speed limit of 12m/s between moments 48100ms-51400ms. In the top row, vehicle 164 (circled in black) violated the user-set speed limit of 6m/s ± 2m/s between moments 299300ms-300800ms. Both scenarios involve vehicles from the *vehicle_tracks_000.csv* dataset on the *DR_USA_Intersection_MA.osm* map.

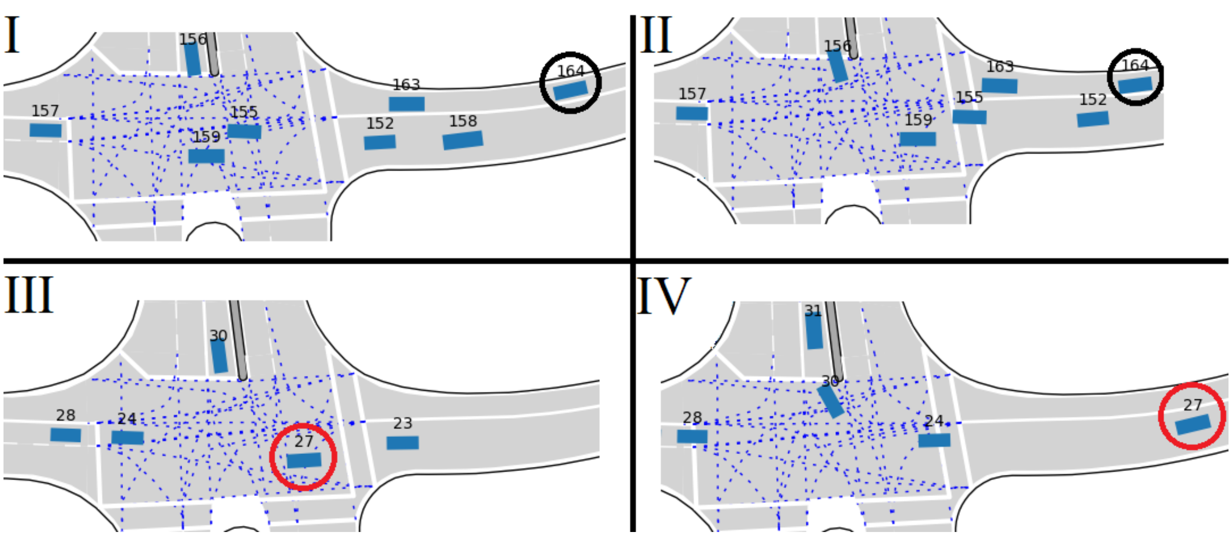


Figure 3.17: Two examples of a target speed with waypoint violation occurring (**red and black circles**).

Goal Waypoint With Target Speed Algorithm The provided algorithm (Goal Waypoint With Target Speed Algorithm) checks whether a vehicle has met the criterion of maintaining a target speed within the vicinity of a designated waypoint along its route through the intersection:

Goal Waypoint With Target Speed Violation Check: Checks to see if the goal waypoint and target speed requirements are satisfied.

Input Data: *waypoint_position*, Waypoint-epsilon (ϵ_W), *target_speed*, Target-speed-epsilon (ϵ_T), *speed_checking_mode*
Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), Lanelet map ($M$)
Result: Determines if the vehicle meets the goal waypoint and target speed requirements during its trip at any point.

list ← create an empty list
for *every state s at timestamp t* (s_t) *in vehicle trajectory* S **do**
    speed ← Get vehicle's speed in state s_t
    velocity_difference ← *speed* − *target_speed*;
    Δ ← determine the distance between the vehicle at state s_t and the *waypoint_position* **if** $\Delta < \epsilon_W$ **then**
        if *(|velocity_difference|* $< \epsilon_T$ *AND waypoint_checking_mode is set to check if the speed is within the target speed limit) AND (velocity_difference+* $\epsilon_T < 0$ *AND waypoint_checking_mode is set to check if the speed is surpassing a maximum speed limit)* **then**
            list ← append the current time s_t to the given list
return *list*

Limitations The approach's limitation lies in its frame-by-frame evaluation, making it necessary for users to conduct additional checks to ascertain the duration or consistency of rule compliance over multiple frames or as a ratio of the entire trip when assessing rule adherence.

3.2.8 Tailgating

Tailgating Detection The *tailgating* rule assesses whether instances of tailgating have occurred between vehicles during a trip.

Rule Role The minimum tailgating distance is determined using Intel's RSS.I formulation formulation:
$d_{min} = [v_r p + \frac{1}{2} a_{max,accel} p^2 + \frac{v_r + p a_{max,accel}}{2 a_{min,brake}} - \frac{v_f^2}{2 a_{max,brake}}]_x \; with[x]_x := max\{x, 0\}$ [3]
The constraints for vehicle tailgating, including response time (p), maximum braking acceleration for the front vehicle ($a_{max,brake}$), maximum acceleration for the rear vehicle ($a_{max,accel}$), and minimum braking acceleration for the rear vehicle ($a_{min,brake}$), are considered [3].

The tailgating detection process involves iterating over all moments a vehicle encounters during its trip through an intersection. Two modes of tailgating are checked:

1. **Moving Mode:** This mode checks for tailgating occurrences when a vehicle is in motion, relying on Intel's RSS.I formulation formulation.

2. **At Rest Mode:** This mode determines the distance between vehicles at rest, flagging tailgating if vehicles are within two meters of each other while stationary. Vehicle orientations are also considered to ensure they are facing each other when driving down the road; otherwise, tailgating is not flagged as having occurred for those vehicles.

Example The image (Figure 3.18) provides an example of a vehicle violating Intel's RSS.I formulation formulation and engaging in tailgating. In this scenario, Vehicle 95 is observed engaging in tailgating with another vehicle (Vehicle 94). Vehicle 95 is moving at a speed of approximately 9.48m/s, while Vehicle 94 is travelling at around 9.30m/s. Tailgating detection parameters, including a response time of 0.75s, maximum front vehicle braking acceleration of $7.85 m/s^2$, maximum rear vehicle acceleration of $9.81 m/s^2$, and minimum rear vehicle braking acceleration of $4.61 m/s^2$, are set for Intel's RSS.I formulation formulation.

As depicted in Figure 3.18), Vehicle 95 has a minimum tailgating distance (d_{min}) of roughly 5.81m. Notably, the front point of Vehicle 95 is well within the bounds of the rear point of Vehicle 94, indicating a tailgating violation according to Intel's RSS.I formulation formulation.

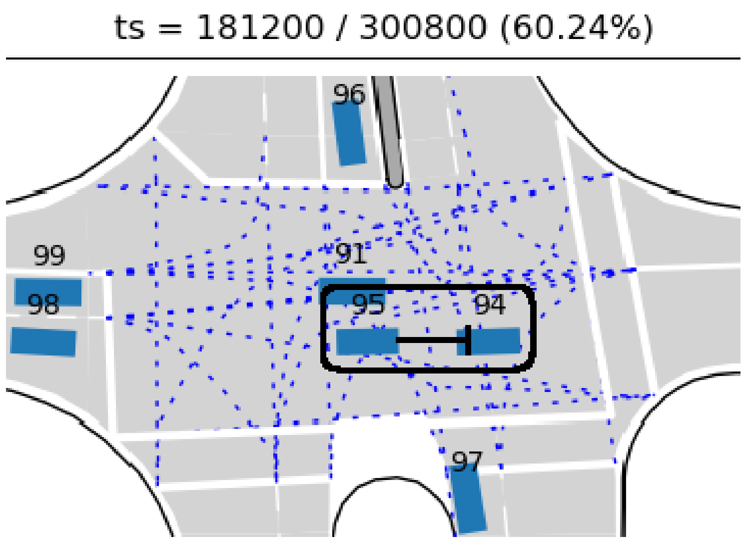


Figure 3.18: An example of a tailgating violation check (**black bar between both vehicles**) between two vehicles (**black square**) at a single frame.

Tailgating Algorithm The provided algorithm, known as the *Tailgating Algorithm*, is employed to ascertain whether a vehicle has been involved in instances of tailgating at any point during its journey. Key checks are incorporated into the algorithm:

- **Line 8:** examines whether a reference vehicle (state s_t) has come to a stop at any moment during its trip.

- **Lines 10 and 11:** iterates through all vehicles along the intersection, excluding the reference vehicle.
- **Line 12:** determines if the states of the other vehicle and the reference vehicle co-occur (during the same recorded timestamp) and assesses if the vehicle is positioned behind the other vehicle along the road.
- **Line 14:** performs a similar check for any other vehicle (state w_p).
- **Line 17:** ascertains if both vehicles are stopped and if the program is configured to check for either stopped vehicle tailgating instances or both stopped and moving vehicle tailgating instances.
- **Line 24:** checks if the program is set to assess either moving vehicle tailgating instances or both stopped and moving vehicle tailgating instances.
- **Line 26:** additionally verifies if the reference vehicle (state s_t) is within the d_{min} distance to another vehicle (state w_p).

The algorithm yields results in terms of the total time spent tailgating, expressed as a proportion of the total time covered by the evaluated time frames.

Tailgating Violations Check: Finds all moments a vehicle V engages in tailgating throughout its trip.

Input Data: Distance threshold (θ_{Δ}), Speed threshold (θ_{speed}), response time (p), maximum braking acceleration for front vehicle ($a_{max,brake}$), maximum acceleration for rear vehicle ($a_{max,accel}$), minimum braking acceleration for rear vehicle ($a_{min,brake}$), *tailgating_checking_mode*

Parameters: Vehicle trajectory ($S = s_{t_0}, s_{t_1}, \dots$), *vehicle_length*, Lanelet map (M), Vehicle recording ($R = (S_0, S_1, \dots)$)

Result: Obtains all moments vehicle (V) engaged in tailgating among all vehicles in Recording D

```
vehicle_stopped_one ← FALSE
vehicle_stopped_two ← FALSE
list ← create an empty list
// Counter for the number of frames where the vehicle is tailgating
tailgating_occurance ← 0
// Counter for the number of frames being checked
driving_state ← 0
for every state s at timestamp t (s_t) in vehicle trajectory S do
    v_r ← gets the speed of the vehicle at state s_t
    if v_r ≤ θ_speed then
        vehicle_stopped_one ← TRUE
    for every vehicle trajectory S_i in R where S_i is not S do
        for every state w at timestamp p (w_p) in vehicle trajectory S_i do
            if t = p AND the vehicle at state (s_t) is behind the vehicle at state (w_p) then
                v_f ← gets the speed of the other vehicle at state w_p
                if v_f ≤ θ_speed then
                    vehicle_stopped_two ← TRUE
                Δ ← the distance between the front of the reference vehicle at state s_t and the rear of
                  the other vehicle at state w_p
                // Checks for tailgating while stopped
                if (vehicle_stopped_one AND vehicle_stopped_two) AND (tailgating_checking_mode
                  is set to check both moving and stopped vehicles OR tailgating_checking_mode is set
                  to check stopped vehicles) then
                    driving_state ← 1
                    if Δ < θ_Δ then
                        if vehicles at states s_t and w_p are facing the same direction as indicated by
                          their yaw values then
                            tailgating_occurance ← add 1 to the counter
                d_min ← Intel's_RSS.I(v_r, v_f, p, a_max,accel, a_min,brake, a_max,brake)
                //Checks for tailgating while driving
                if tailgating_checking_mode is set to check both driving and stopped vehicles OR
                  tailgating_checking_mode is set to check moving vehicles then
                    driving_state ← 1
                    if Δ < d_min then
                        if vehicles at states s_t and w_p are facing the same direction as indicated by
                          their yaw values then
                            tailgating_occurance ← add 1 to the counter
            vehicle_stopped_two ← FALSE
    vehicle_stopped_one ← FALSE
return (tailgating_occurance / driving_state)
```

Limitations A limitation of the approach is that it relies on binary classification (true or false) for tailgating occurrences based on input coefficients for the Intel's RSS.I formulation. It might be more appropriate to determine tailgating events based on a range of coefficients, producing a set of results that can be further analyzed to assess tailgating severity.

Another limitation pertains to the determination of whether tailgating has occurred for a sufficient duration to qualify as a rule violation. One approach could involve calculating the total time a vehicle spends tailgating throughout its trip and expressing it as a proportion of the overall trip time. Establishing a threshold for a high prevalence of tailgating becomes challenging, as factors such as traffic conditions, congestion levels, and the context of the road need to be considered. Comparing a vehicle's behaviour to a baseline norm determined by other vehicles during the same time interval might offer insights into whether it exhibited a high enough prevalence of tailgating to be considered a rule-violating vehicle.

Chapter 4

Evaluation of Traffic Rule Checking System

4.1 System Implementation

In the subsequent section, we delve into the implementation, testing, evaluation, statistical results, and analysis of the traffic rule-checking system. Details regarding the implementation of the C++ program are elaborated in the *System Implementation* section. The *Testing and Evaluation Using Artificial and Naturalistic Test Cases* section provides instances from the testing code of the C++ program. The *Statistical Results and Analysis* section illustrates example rule violation cases, vehicle position data, rule violation data, and statistical outcomes for rule violations derived from the C++ program. The traffic rule-checking system is built as an extension of the Lanelet2 library.

Datasets The datasets used in the study are not flawless, but efforts were made to enhance their accuracy. The interaction dataset, focusing on uS-SSRIs and S-RIs, underwent manual verification to rectify inaccuracies [5]. The system incorporates two datasets, one for uS-SSRIs assessing traffic rule adherence in roundabouts and unsignalized intersections, and another for S-RIs utilizing Waterloo's Multi-Agent Traffic Dataset: Intersection, featuring drone recordings of an urban intersection in Waterloo, Canada [6] While the interaction dataset received manual validation, the Waterloo dataset, comprising thirteen databases with recorded traffic, lacks manual verification, potentially containing errors from sensor readings and detection inaccuracies.

Interaction Dataset Data Extraction The data extraction process involves retrieving information line-by-line from the interaction dataset and storing it in a structured format for each vehicle. The (x, y)-values representing the starting and ending points of the intersection recording, along with the starting and ending yaw for the recording, are recorded in a struct that includes the length and width of the vehicle. Intervals for each moment within the interaction dataset recording are registered, containing details such as the vehicle ID, current timestamp, current X- and Y-velocity component vectors, the vehicle's current yaw, and the current X- and Y-coordinate position.

Signal Regulated Intersection Database The S-RI database from the WISE Lab encompasses records from a bustling intersection in Waterloo, Canada [6]. Various tables within the S-RI database store information pertinent to traffic rule violation detection, including:

1. **Tracks:** This table contains details about agent tracks, providing information such as intersectional entry and exit times, entry gate, and exit gate.
2. **Traffic Lights:** This table is dedicated to storing information related to traffic lights, including their status and configurations, during specific time intervals.
3. **Traffic Regions Def:** This table holds spatial information defining the regions associated with traffic lights and gates.
4. **TRAJECTORIES_0<file_id>:** These tables store trajectory information for agents in the corresponding database. They include details such as the position, speed, angle, and acceleration of agents at specific time intervals from the beginning of the recording.

These tables collectively form a comprehensive dataset that facilitates the analysis of traffic dynamics, rule violations, and the performance of the traffic control system at the intersection.

The process of traffic rule detection involves iterating through time intervals for each vehicle Track ID and cross-referencing the recorded entry time at the gate with the current time interval. The determination of the light state at the entry gate traffic light is made by comparing the entry gate information from the tracks table with the spatial details provided in the *"Traffic Regions Def"* dataset. The Gate ID for the entry gate is then utilized to establish the traffic light state when the vehicle enters the intersection.

4.2 Testing and Evaluation Using Artificial and Naturalistic Test Cases

Included is a set of test cases depicting examples from the program's test case code.

4.2.1 Artificial Test Cases

The Artificial Test Cases evaluate traffic rules in an artificial environment, providing examples of rule compliance based on vehicle positions and corresponding lanelets. The test cases cover various intersectional rules, excluding permissive left turns and speed limit violations, as these require naturalistic driving data. The test cases are designed to assess rule compliance across intersections, particularly focusing on S-RI and uS-SSRI traffic rules.

The Artificial Test Cases are conducted in an artificial Lanelet2 map within a testing file, and they do not rely on naturalistic driving data. The map features a four-way intersection, and Lanelet2 elements such as points, linestrings, and lanelets are defined in the testing script to create the environment in a two-dimensional plane. Vehicles are positioned on the map using pre-defined points, representing their locations along the intersection. The points are defined with (x, y, 0) coordinates, indicating a flat, two-dimensional Lanelet2 four-way intersection.

The test cases cover a wide range of traffic rule scenarios associated with both uS-SSRIs and S-RIs. The rules include checks for vehicle turns, stop sign compliance, and traffic light adherence during single-frame encounters at intersections where right-of-way is regulated. For stop sign checks, vehicles are examined to determine if they need to stop at a stop sign before entering the intersection. In traffic light checks, vehicles are assessed to determine if they should proceed or stop at intersections. uS-SSRIs are evaluated to determine if a vehicle should continue for a turn after stopping, based on uS-SSRI right-of-way priority rules. The test cases also consider situations where other vehicles may be stopped along the intersection at the same frame as the reference vehicle, assessing which vehicle has the right-of-way during that frame.

For detailed implementation and code related to these test cases, the *lanelet2_traffic_rules.cpp* testing script associated with the thesis can be referenced.

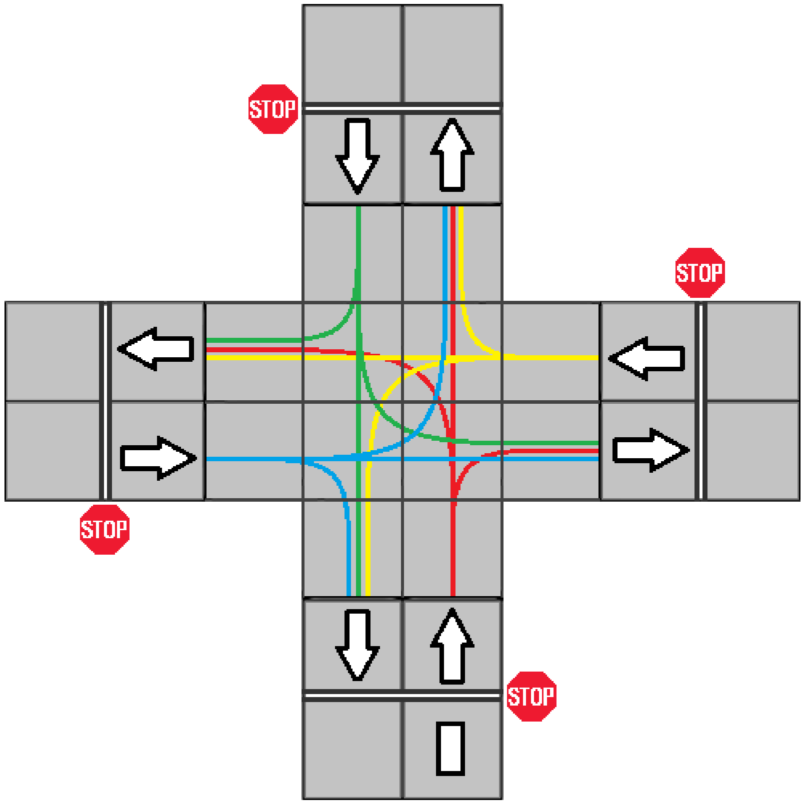


Figure 4.1: Artificial Test Case Representation.

Artificial Test Case Checks				
Traffic Rule Type	**Regulation Type**	**Traffic Rule**	**Test Case Implementation**	**No. Tests Implemented**
Intersectional Rules	Stop-Sign-Regulation	Stop Sign Checking Algorithm	Figure 4.2	8
Intersectional Rules	Stop-Sign-Regulation	Right Before Left Movement Priority Algorithm	Figure 4.5	12
Intersectional Rules	Stop-Sign-Regulation	Intersection Priority Algorithm	Figure 4.6	8
Intersectional Rules	Stop-Sign-Regulation	Left Turns Algorithm	Figure 4.7	10
Intersectional Rules	Traffic-Light-Regulation	Red Light Violation Algorithm	Figure 4.3	2
Intersectional Rules	Traffic-Light-Regulation	Yellow Light Violation Algorithm	Figure 4.4	5
Intersectional Rules	Traffic-Light-Regulation	Permissive Green Light Violation Algorithm	NA	NA
Non-Intersectional Rules	Traffic Sign Regulation	Speed Limits Algorithm	NA	NA
Non-Intersectional Rules	NA	Waypoints Algorithm	NA	NA
Non-Intersectional Rules	NA	Collisions Algorithm	NA	NA
Non-Intersectional Rules	NA	Target Speed Algorithm	NA	NA
Non-Intersectional Rules	NA	Offroad Violations Algorithm	NA	NA
Non-Intersectional Rules	NA	Goal Waypoint With Target Speed Algorithm	NA	NA
Non-Intersectional Rules	NA	Tailgating Algorithm	NA	NA

Table 4.1: Table of Implemented Artificial Test Cases.

Example One: Vehicle is Past Stop Line As seen in the scenario (Figure 4.2), the vehicle has crossed the stop line, indicating that it is no longer in close proximity ($\Delta \leq \theta_\Delta$) to the stop line. Consequently, the vehicle fails to meet the stipulated requirement of coming to a stop before the stop sign, unless it had already come to a stop prior to reaching the stop line.

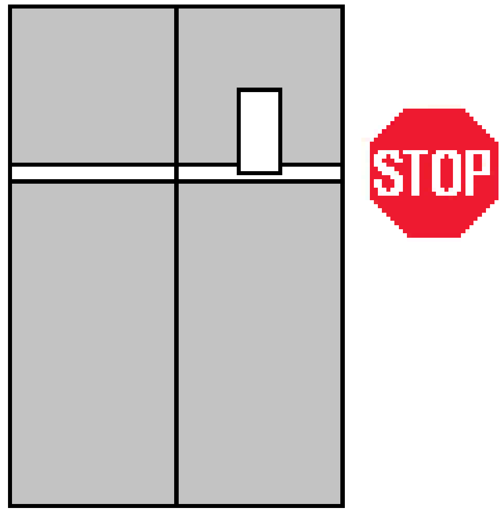


Figure 4.2: The Vehicle is Past the Point Where a Stop Would Satisfy a Stop Sign Check.

Example Two: Vehicle is Behind a Red Light As depicted in the image (Figure 4.3), the vehicle is positioned behind a traffic light displaying a red signal phase. It is essential to note that the vehicle cannot proceed through the intersection until the red signal phase concludes and transitions to a green signal. As long as the vehicle remains behind the stop line corresponding to the traffic lights, a violation related to running a red light is prevented.

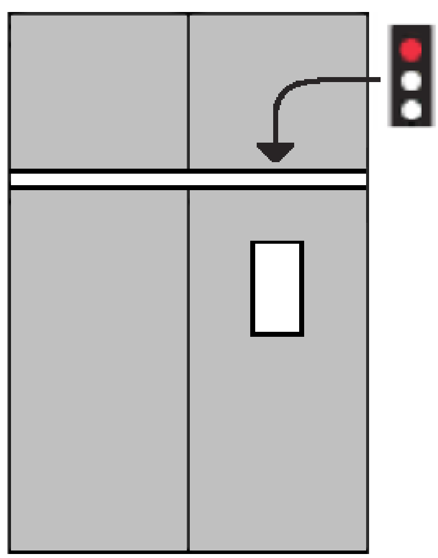


Figure 4.3: The Vehicle is not Violating the Red Light Unless it Crosses the Stop Line.

Example Three: Vehicle Cant Stop At Yellow Light In another example (Figure 4.4), we observe a scenario with a traffic light transitioning from a green to a yellow signal state. The stopping distance required to reach the traffic light stop line is two meters. If the vehicle's distance to the stop line is one meter, it can proceed through the yellow light without a violation, as it is impractical to come to a stop before reaching the stop line. However, if the distance to the stop line is ten meters, the vehicle must come to a complete stop before the traffic light stop line, as there is sufficient space to do so.

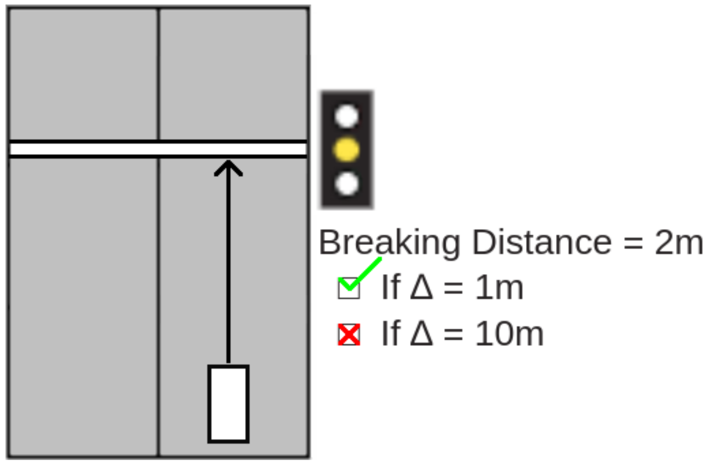


Figure 4.4: Traffic Light - Yellow Light.

Example Four: Illegal Right Before Left Vehicle Movement Priority The image (Figure 4.5) illustrates an example of a vehicle that should yield (bottom), as it is attempting to move after stopping at the same time as a vehicle on its right.

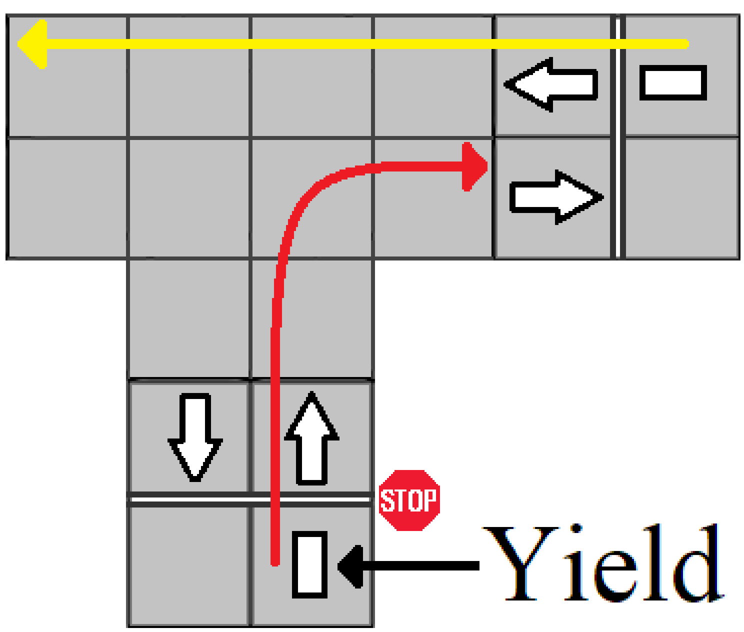


Figure 4.5: Lanelet Left Before Right Illustration.

Example Five: Illegal Lanelet Priority Movements The image (Figure 4.6) illustrates an example of a vehicle that should yield (right), as it does not have priority over the bottom vehicle.

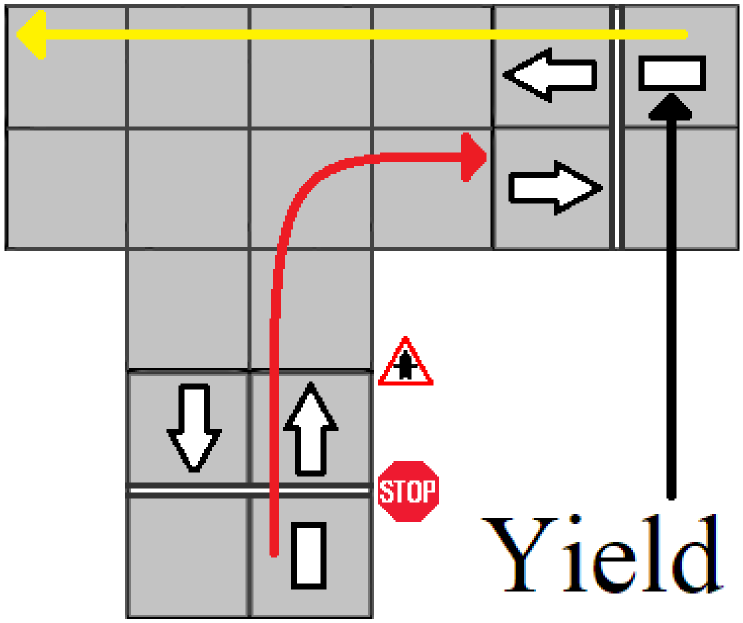


Figure 4.6: Lanelet Priority Illustration.

Example Six: Illegal Left Turning Priority Movements The image (Figure 4.7) illustrates an example of a vehicle that should yield (bottom) given oncoming traffic has the right-of-way over left-turning vehicles, with two vehicles stopping along the intersection simultaneously.

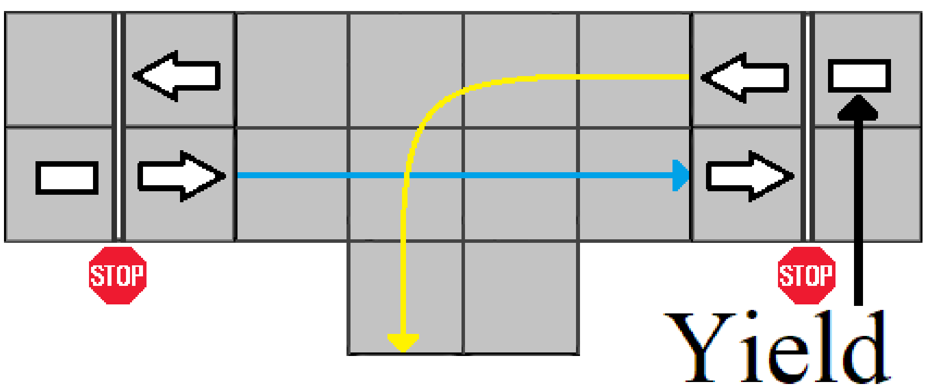


Figure 4.7: Lanelet Left Turn Illustration.

Example Seven: No Illegal Movement Rules Apply The image (Figure 4.8) illustrates an example where neither vehicle must yield to the other after both vehicles stop simultaneously, as no all-way-stop priority rules apply.

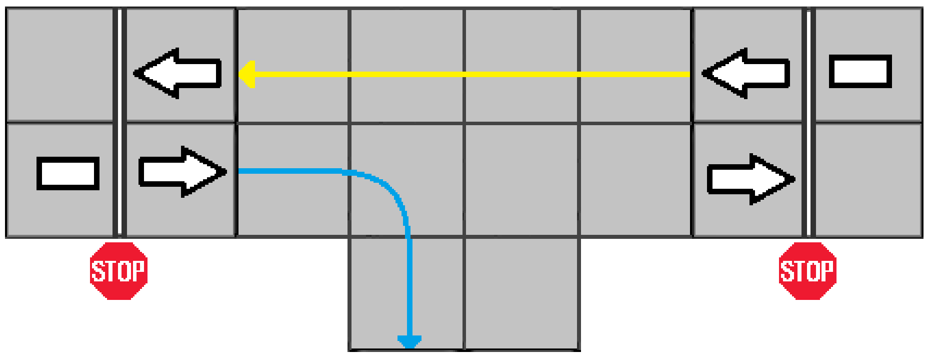


Figure 4.8: Neither Vehicle has Intersection Priority Over the Other.

4.2.2 Naturalistic Test Cases

Naturalistic Test Cases - Implementation

Naturalistic test cases are constructed using data from the interaction dataset, which captures the real-world movements of traffic participants in dynamic driving scenarios across different countries [5]. The dataset encompasses diverse driving scenarios such as roundabouts, uS-SSRIs, merging, lane changing, and S-RIs [5]. For the purposes of this thesis, the primary focus is on roundabouts and uS-SSRIs.

The dataset is comprised of CSV files containing time-dependent information for each track. This information includes track ID, vehicle type, size, position, velocity, and orientation, recorded at various time intervals.

In these test cases, the reference vehicle's coordinates are compared to positions on a Lanelet Map (M), and regulatory element (such as all way stop regulatory elements or right of way regulatory elements) are extracted along with associated stop lines. Intersectional rule violations and driving violations are assessed for each moment the vehicle is documented within the intersection in the interaction dataset.

To cross-reference vehicle coordinates at different timestamps, OpenStreetMap files are utilized, mapping them to specific lanelets associated with the intersection geometry. Comparisons are made at corresponding timestamp intervals, representing frames in the recording. Vehicle attributes like speed, position, and yaw are scrutinized concerning other vehicles or predefined thresholds to identify rule violations.

These test cases span various traffic junction geometries, encompassing three-way intersections, four-way intersections, and roundabouts.

Determining Stops with Interaction Dataset The position and velocity values of a vehicle for each time interval are extracted from the interaction dataset. The vehicle's position along a lanelet is analyzed to ascertain its distance from a stop line, if one is present. If the vehicle is within a specified distance threshold of the stop line and its velocity falls below a predetermined threshold, signifying that it has come to a stop, then the vehicle is considered to have stopped at the stop line. Conversely, if a vehicle never comes to a stop at a stop line within an intersection, it is deemed to have violated the requirement to stop at the stop sign.

Stopping Velocity Thresholds The image (Figure 4.9) illustrates an example of using a stopping velocity threshold. With a velocity threshold for stop detection set at 0.95 m/s, Vehicles 53 and 54 are identified as violating the Left-Before-Right rule with respect to Vehicle 50. Adjusting the velocity threshold to 1 m/s results in Vehicle 53 no longer being considered a violating vehicle. Alternatively, utilizing standstill detection instead of distance from the stop line to determine if a vehicle has started moving leads to Vehicle 54 no longer being considered a violating vehicle. The progression of time is depicted from the left to the right image. The potential offending vehicles, 53 and 54, are highlighted with red circles, while the victim vehicle, 50, is highlighted with a green circle.

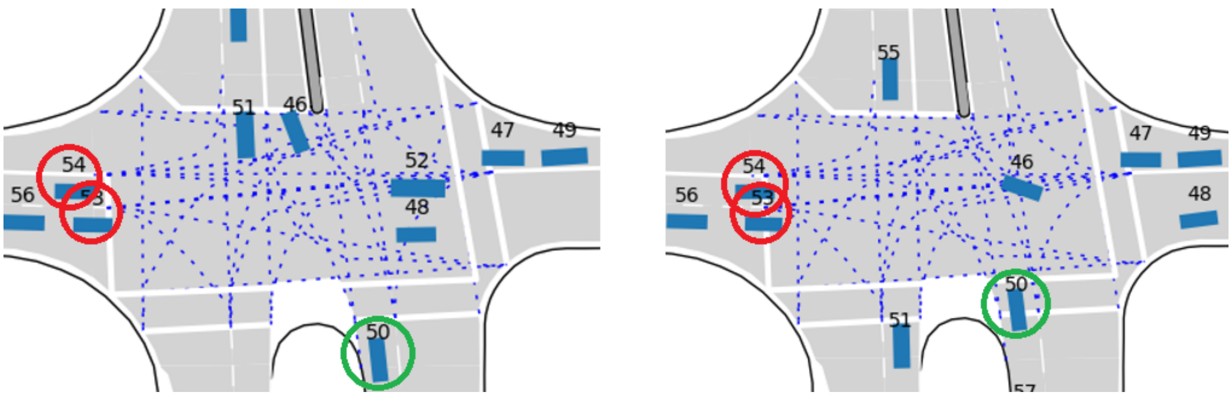


Figure 4.9: Stopping Velocity Threshold Example.

Stopping Distance Thresholds the irregularly shaped stop line introduces a challenge where the vehicle is deemed too far from the line to be considered as having successfully stopped at the sign. Specifically, when the θ_Δ is set to five meters, Vehicle 64 is classified as not having stopped. However, adjusting θ_Δ to six meters is adequate to recognize Vehicle 64 as having stopped at the stop sign. This example emphasizes that stopping distance threshold values are context-specific, varying based on intersection geometry. Different intersections may require higher stop thresholds to avoid false negatives in stop-sign rule violation detection. The red circle highlights Vehicle 64 in this scenario.

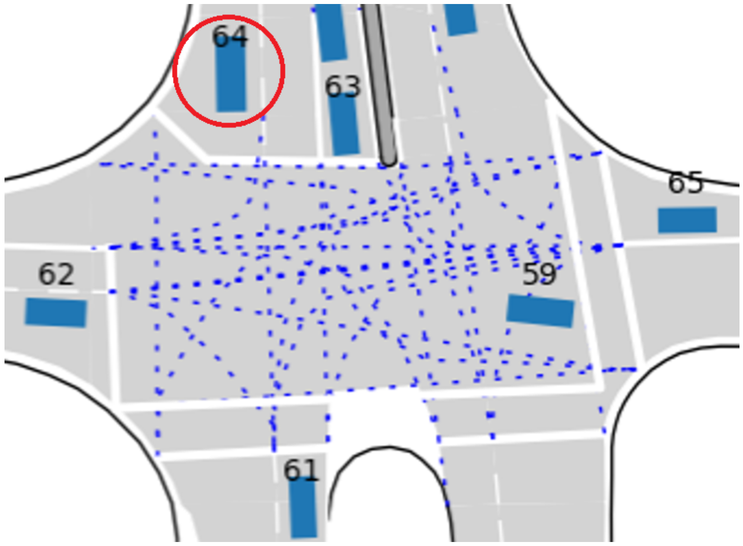


Figure 4.10: Stopping Distance Threshold Example.

Turns After Stops with Interaction Dataset Turns after stops are evaluated for uS-SSRI scenarios. In these situations, vehicles are required to come to a complete stop at a sign and subsequently proceed through an intersection not governed by traffic lights. When multiple vehicles come to a stop simultaneously, the rule mandates that a vehicle must yield to those on the right before proceeding. Additionally, vehicles with priority, if applicable, must be allowed to pass, and making left turns across the intersection that cut off oncoming traffic is prohibited. The right-of-way along the intersection is granted to the vehicle that arrives first.

Example Illegal Movement Violation One - Intersection Right Vehicle Priority The image (Figure 4.11) illustrates a violation of the Left-Before-Right vehicle priority rule at the intersection. In this situation, both Vehicle 22 and Vehicle 18 proceed straight. However, Vehicle 18 is positioned to the right of Vehicle 22, indicating that, considering their simultaneous arrival, Vehicle 18 should have the right-of-way. It's essential to note that time progresses from the left to the right in the image. The red circle represents vehicle 22 (the offender), while the green circle represents vehicle 18 (the victim).

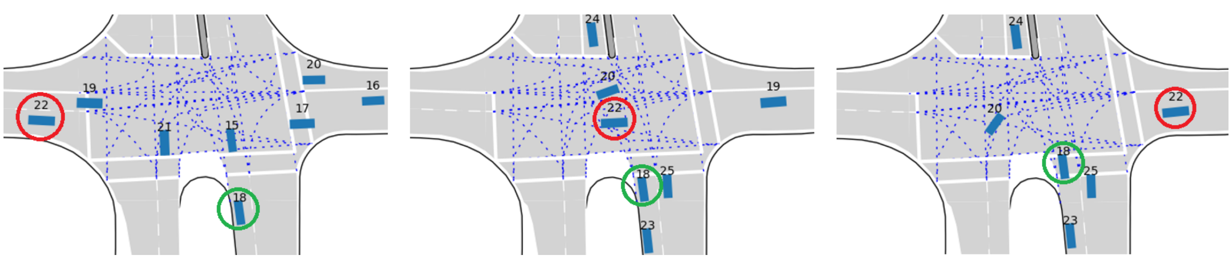


Figure 4.11: Right Vehicle Priority Example.

Example Illegal Movement Violation Two - Intersection Oncoming Vehicle Priority on Left Turn Violation Example The image in Figure 4.12 illustrates a violation of the Left-Turning vehicle priority rule at the intersection. It's important to note that time progresses from the left to the right in the image. Vehicle 64 turns left down the intersection, but Vehicle 62 arrives at roughly the same time (Vehicle 62 arrived just before Vehicle 64) and is attempting to go straight. Vehicle 64 cuts off Vehicle 62 by moving down the intersection first through a left turn. The red circle represents vehicle 64 (the offender), and the green circle represents vehicle 62 (the victim).

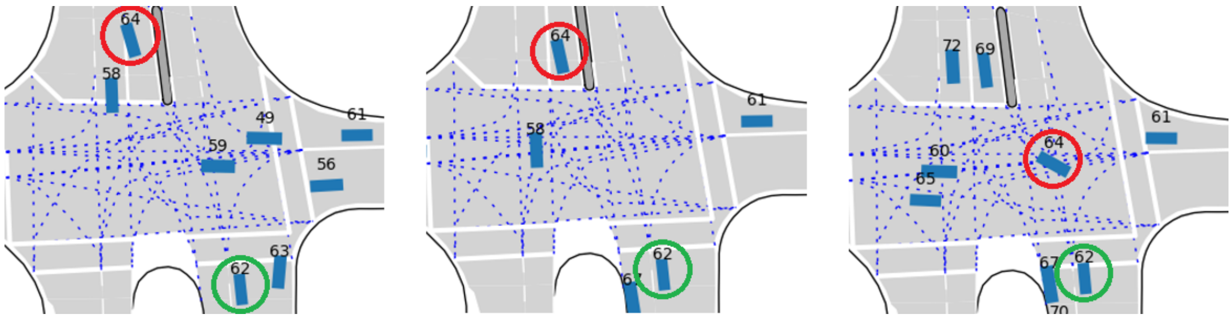


Figure 4.12: Intersection Oncoming Vehicle Priority on Left Turns Example.

Offroad Violation Boundary Mapping Examples In the provided image (Figure 4.13), the top left corner (area I) displays offroad violations identified using only the vehicle's centerpoint. The adjacent images to the left of the corner (above) showcase instances of offroad violations considering half (area II) or the full extent (area III) of the distance from the center of the vehicle to its corners. The two images on the bottom row illustrate offroad violations when utilizing 80% (area IV) or 90% (area V) of the distance from the center of the vehicle to its corners.

The black squares in the images represent a vehicle's centerpoint during the moment it was involved in an offroad violation. The horizontal and vertical bars on the black squares indicate the length and width of the respective evaluated vehicle. Different parameter settings for length and width from the centerpoint yield varying outcomes for detecting offroad violations:

- **Area I:** detects only those vehicles that are entirely offroad.
- **Area II:** identifies vehicles that are offroad or positioned too close to the road's curb to be entirely on the road.
- **Area IV:** detects vehicles that are driving too close to the curb, implying they are outside lane boundaries but not completely off the road.
- **Areas III and V:** detect numerous vehicles situated between two separate lanes, flagged as off the road because parts of the vehicles are not within any lane—they are in the space between lanes.

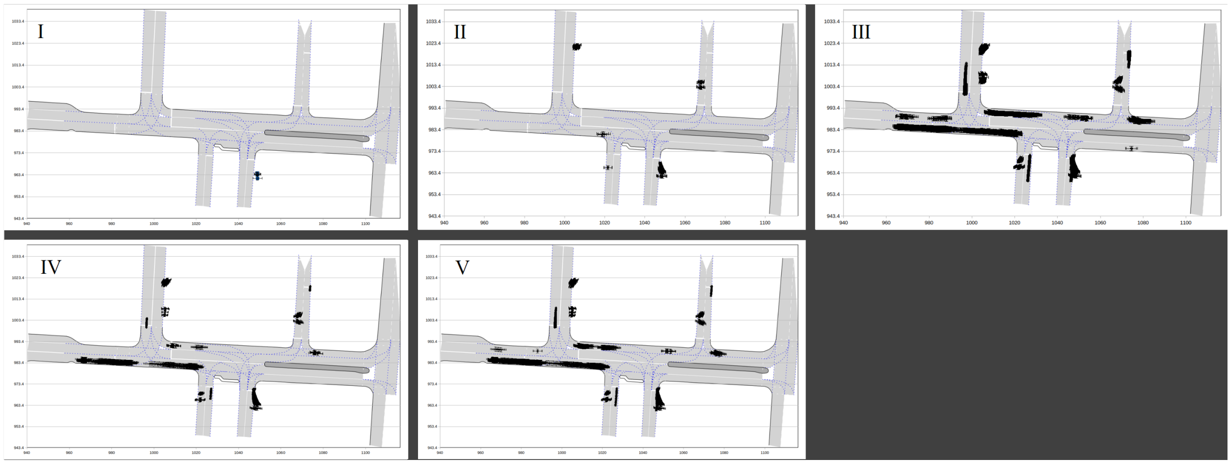


Figure 4.13: Illustration of Different Thresholds for Offroad Violation Mappings.

Tailgating Violation Instances Examples In the presented image (Figure 4.14), black squares depict occurrences of tailgating for vehicles in the *USA_Roundabout_EP* intersection using the *vehicle_tracks_000.csv* dataset. Notably, instances of tailgating are most frequently observed at the entry points of intersections, with a solitary occurrence identified along the roundabout. It's essential to mention that the default tailgating coefficients, as outlined in Table A.2, were employed for these detections.

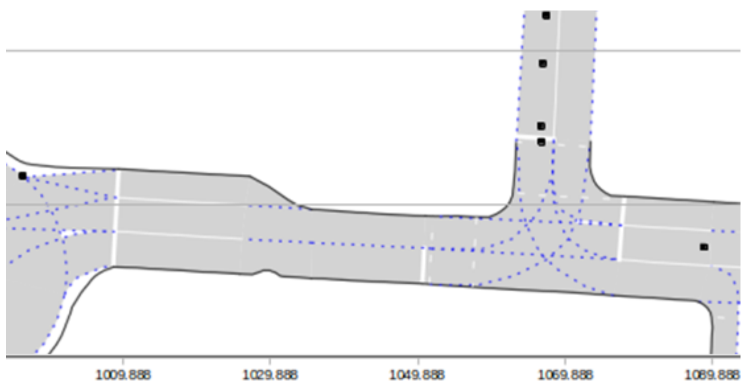


Figure 4.14: Illustration of Different Thresholds for Tailgating Violation Mappings.

In the presented image (Figure 4.15), black squares indicate instances of tailgating for vehicles using the *vehicle_tracks_000.csv* dataset in the *USA_Roundabout_EP* intersection. In quadrant I (top left), tailgating instances are shown with maximized rear vehicle acceleration (9.62 $\frac{m}{s^2}$). In quadrant II (top right), tailgating instances are depicted with maximized rear vehicle acceleration (10.81 $\frac{m}{s^2}$). Quadrant III (bottom left) displays instances of tailgating with maximized minimal rear vehicle braking acceleration (6.08 $\frac{m}{s^2}$), while quadrant IV (bottom right) illustrates instances of tailgating with a maximized vehicle response time (1 s).

Upon comparison with the tailgating results using the *default* tailgating coefficients (Table A.2) in Figure 4.14, some differences are noted. Quadrant III is missing a single instance of tailgating along the roundabout to the left of the image. Quadrant IV is missing an instance of tailgating along the entry intersection at the top of the image, and it has an additional instance of tailgating along the entry lane into the roundabout to the left of the center of the image. Quadrants I and II show identical results to those obtained with the *default* tailgating coefficients.

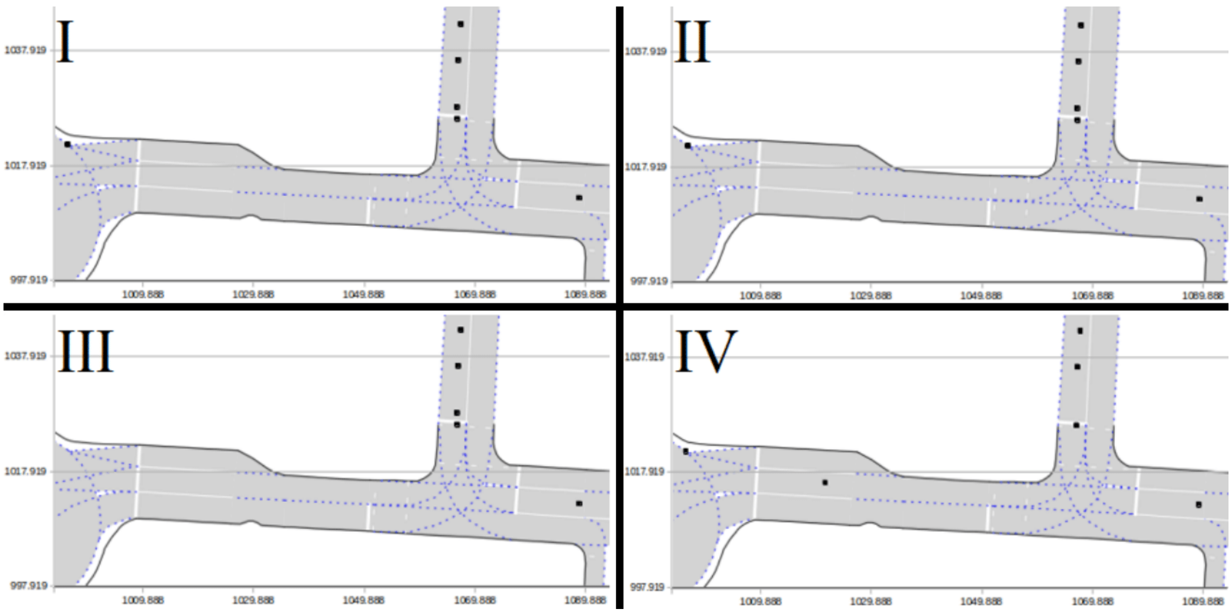


Figure 4.15: Illustration of Different Thresholds for Tailgating Violation Mappings.

In the presented image (Figure 4.16), black squares indicate instances of tailgating for vehicles using the *vehicle_tracks_000.csv* dataset in the *USA_Roundabout_EP* intersection. Quadrant I (top left) shows tailgating instances with a minimized front vehicle braking acceleration (6.08 $\frac{m}{s^2}$). Quadrant II (top right) displays tailgating instances with a minimized rear vehicle acceleration (8.81 $\frac{m}{s^2}$). Quadrant III (bottom left) illustrates instances of tailgating with a minimized minimal rear vehicle braking acceleration (3.14 $\frac{m}{s^2}$), while quadrant IV (bottom right) shows tailgating instances with a minimized vehicle response time (0.5 s).

Upon comparison with the tailgating results using the *default* tailgating coefficients (Table A.2) in Figure 4.14), some differences are noted. Quadrants I and IV are missing an instance of tailgating along the roundabout to the left of the image. Quadrants II and III exhibit identical results to those obtained with the *default* tailgating coefficients.

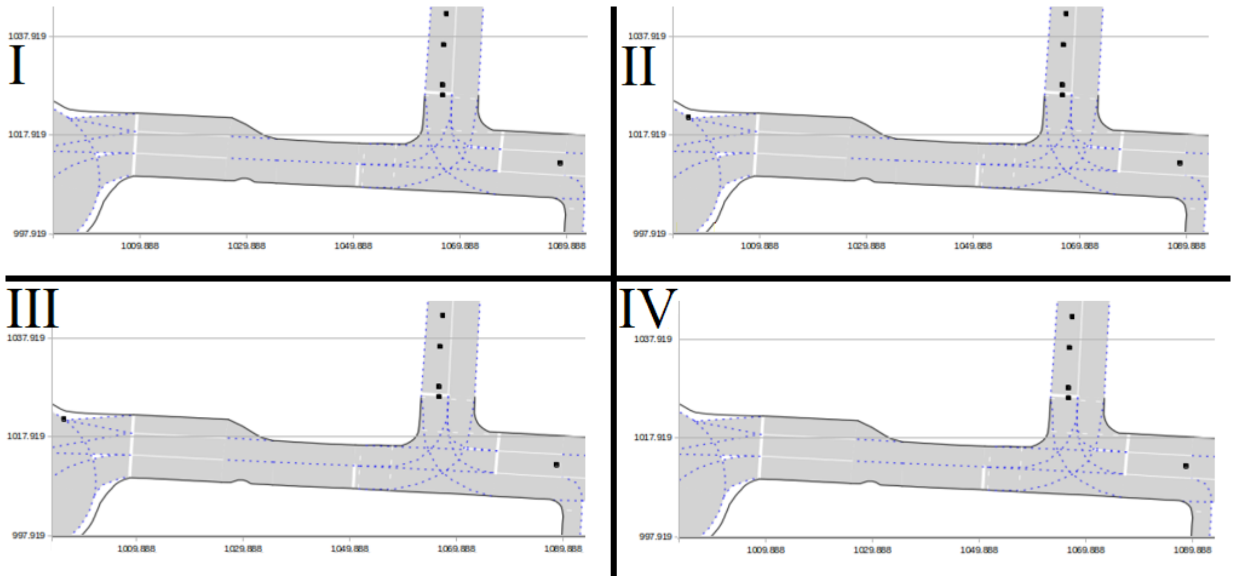


Figure 4.16: Illustration of Different Thresholds for Tailgating Violation Mappings.

In the presented image (Figure 4.17), squares represent instances of tailgating for vehicles using the *vehicle_tracks_001.csv* dataset at the *DR_USA_Intersection_MA* intersection. Notably, instances of tailgating are more prevalent at the entry points of intersections, with a smaller number occurring in the middle of an intersection. It is important to mention that the *default* tailgating coefficients (see Table A.2) were applied in this analysis.

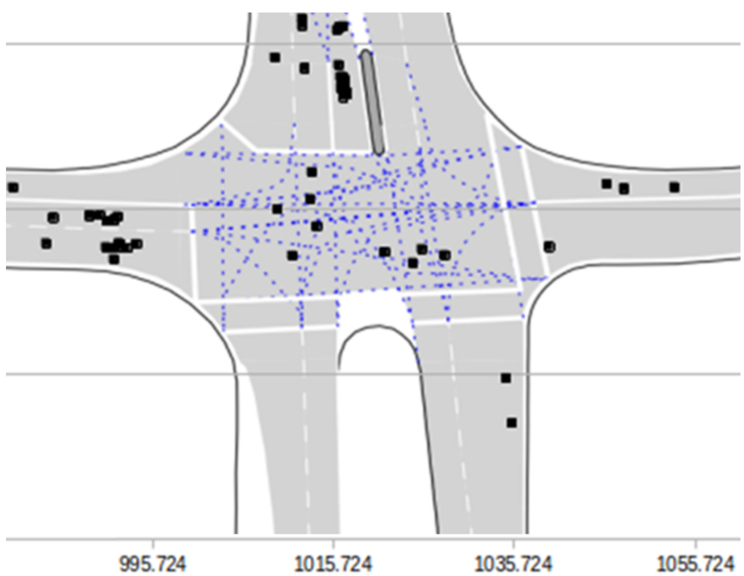


Figure 4.17: Illustration of Different Thresholds for Tailgating Violation Mappings.

In the provided image (Figure 4.18), squares represent instances of tailgating for vehicles using the *vehicle_tracks_001.csv* dataset at the *DR_USA_Intersection_MA* intersection. When comparing these tailgating results with the default parameters (Table A.2) in Figure 4.17, several observations can be made:

1. **Top Left-Hand Corner (Quadrant I):** Instances of tailgating with a maximized front vehicle braking acceleration setting (9.62 $\frac{m}{s^2}$) exhibit more occurrences at the center of the intersection compared to the default parameters in Figure 4.17.

2. **Top Right-Hand Corner (Quadrant II):** Instances of tailgating with a maximized rear vehicle acceleration (10.81 $\frac{m}{s^2}$) remain the same as the default parameters in Figure 4.17.

3. **Bottom Left-Hand Corner (Quadrant III):** Instances of tailgating with a maximized minimal rear vehicle braking acceleration (6.08 $\frac{m}{s^2}$) show fewer occurrences at the center of the intersection compared to the default parameters in Figure 4.17.

4. **Bottom Right-Hand Corner (Quadrant IV):** Instances of tailgating with a maximized vehicle response time (1 s) exhibit significantly more occurrences of tailgating, especially along the right entry lane of the intersection.

This comparison provides insights into how different tailgating coefficients can influence the detection of tailgating instances in various regions of the intersection.

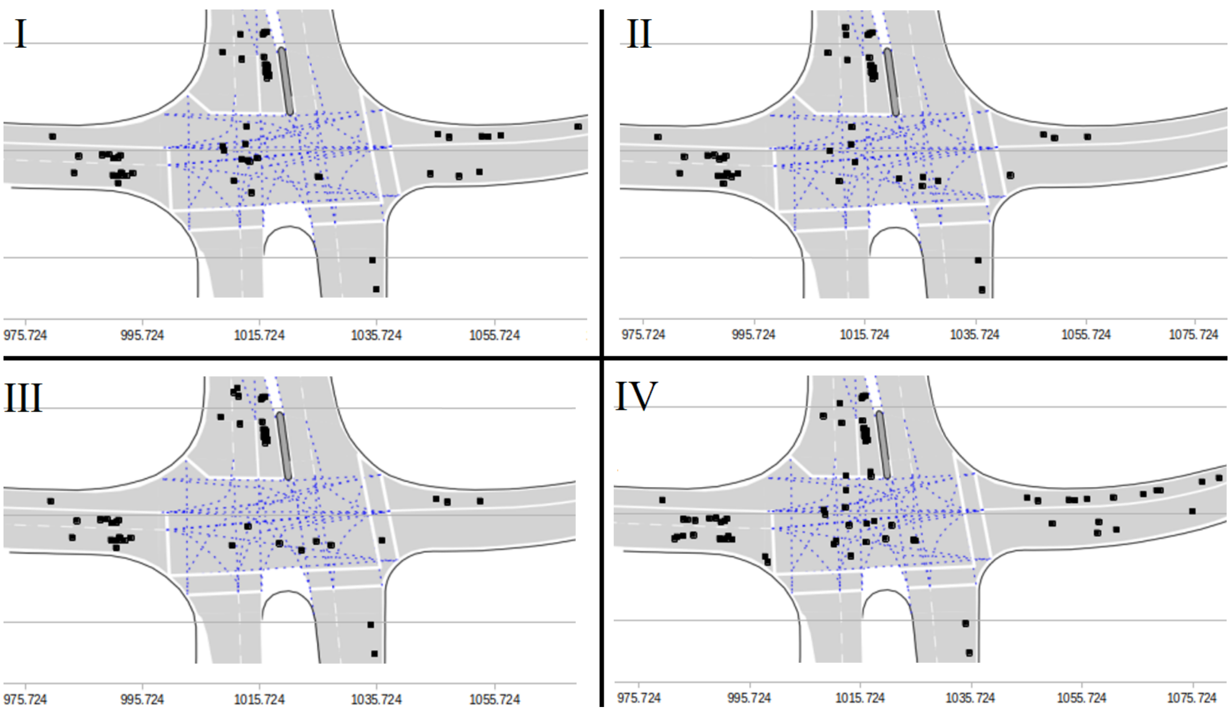


Figure 4.18: Illustration of Different Thresholds for Tailgating Violation Mappings.

In Figure 4.19, squares represent instances of tailgating for vehicles using the *vehicle_tracks_001.csv* dataset at the *DR_USA_Intersection_MA* intersection. A comparison with the *default* tailgating coefficients (Table A.2) in Figure 4.17 reveals the following:

1. **Top Left-Hand Corner (Quadrant I):** Instances of tailgating with a minimized front vehicle braking acceleration (6.08 $\frac{m}{s^2}$) result in fewer occurrences of tailgating along the center of the intersection compared to the default parameters in Figure 4.17.

2. **Top Right-Hand Corner (Quadrant II):** Instances of tailgating with a minimized rear vehicle acceleration (8.81 $\frac{m}{s^2}$) show increased instances of tailgating along the intersection's right entry lane compared to the default parameters in Figure 4.17.

3. **Bottom Left-Hand Corner (Quadrant III):** Instances of tailgating with a minimized minimal rear vehicle braking acceleration (3.14 $\frac{m}{s^2}$) exhibit similar results to the default tailgating coefficients in Figure 4.17.

4. **Bottom Right-Hand Corner (Quadrant IV):** Instances of tailgating with a minimized vehicle response time (0.5 s) show no instances of tailgating at the center of the intersection.

This comparison emphasizes how adjustments to tailgating coefficients can influence the detection of tailgating instances in specific regions of the intersection, providing insights into the sensitivity of the tailgating detection algorithm to parameter variations.

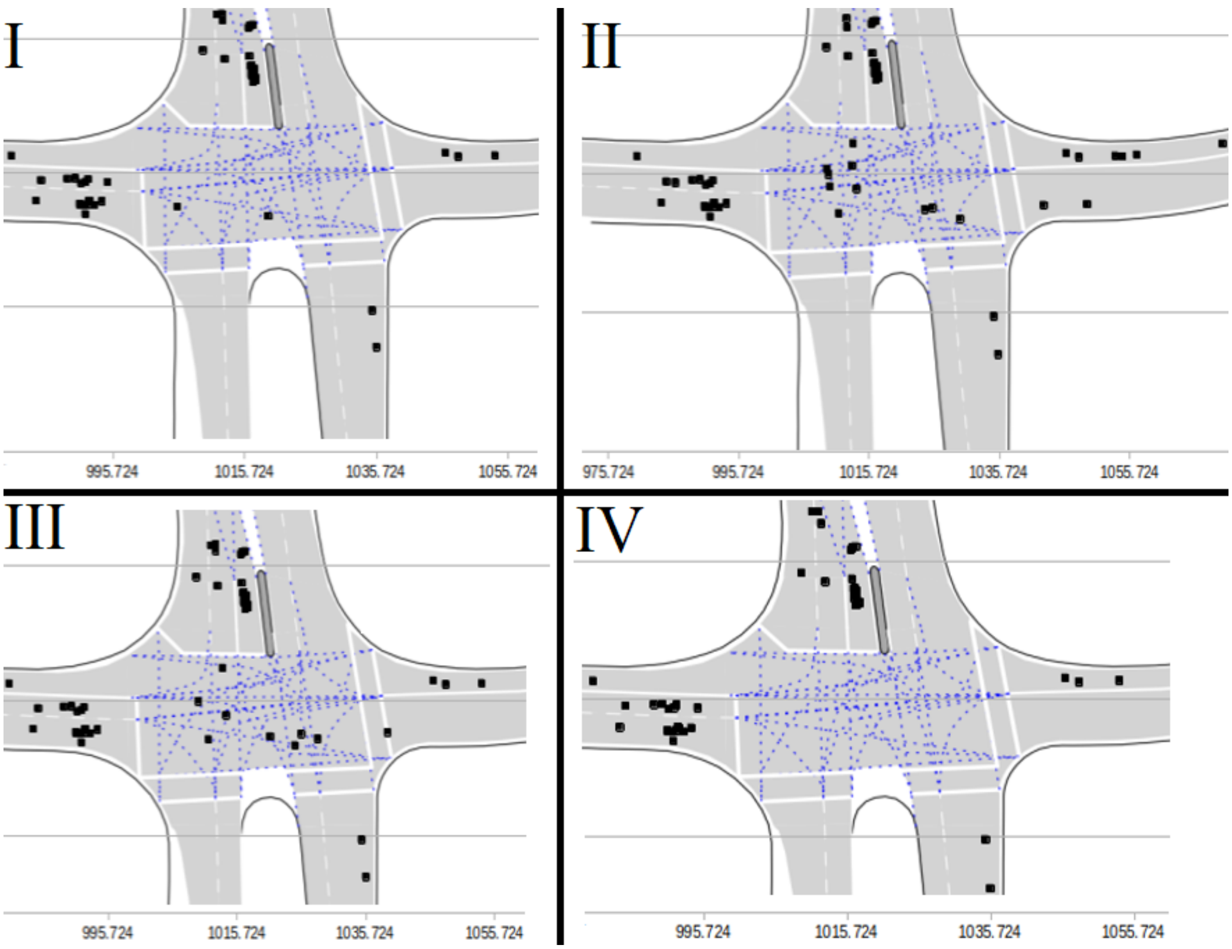


Figure 4.19: Illustration of Different Thresholds for Tailgating Violation Mappings.

Naturalistic Test Cases - Individual Rule Violation Checks

The testing code, detailed in Appendix B for reference, encompasses various test cases designed for the interaction dataset and its corresponding OSM maps. Contained within the Catkin Testing file named *lanelet2_traffic_rules.cpp*, the *"shouldStopTestStatic"* traffic rule vehicle test focuses on assessing vehicle stopping behaviour at a four-way intersection (*DR_USA_Intersection_MA*). The results indicate successful detection of stops for vehicles that come to a complete halt, while issues arise with rolling or running stop signs, leading to stop validation failures. Expanding the testing scope to intersections with multiple stop signs, an examination of an intersection with a connected roundabout (*dataPathUSARoundaboutEP*) is conducted, incorporating multi-stop checking. This extension results in failure flags for stop-checking if vehicles neglect to stop at a second stop line within a traffic junction. Additionally, standard I/O tests, including file reading and formatting tests (*fileReadTestOne*, *fileReadTestTwo*, and *fileReadTestThree*), showcase the correct processing of vehicle information from the interaction dataset. The efficacy of extracting angles to represent lanelet and vehicle orientations and determining turn directions from relative changes in vehicle yaw is demonstrated in the *"laneletGeometryTestStatic"* and *"turnDirectionTestStatic"* tests. These tests also highlight the ability to reference and compare linestrings in lanelets for orientation measurement. Furthermore, the code exhibits the capability to extract and determine the turn direction of a vehicle based on its change in yaw across an intersection. The *"frontPointStatic"* test illustrates the determination of the position and bounds of a vehicle, considering its centerpoint, length, and width.

The testing code includes comprehensive test cases for lanelet speed violations and user-set target speed violations, implemented in the functions *targetSpeedViolations* and *findSpeedViolations*. Waypoint violation detection is assessed by checking if vehicles deviate from proximity to a specific point for at least a single frame in the interaction dataset. Conversely, collision detection is determined by examining whether a vehicle comes close to a particular point for at least one frame. Offroad violations are identified by examining all possible timestamps in the interaction dataset recording and comparing the vehicle's position (using the vehicle's centerpoint or corners) with the closest lanelets. If no lanelets are found near a vehicle, it is considered offroad and outside any lanelet. Vehicle dimensions for length and width are adjusted based on factors of fifty, eighty, or ninety percent when checking the nearest lanelets. The *findNoOffroadViolations* test outputs the total proportion of time vehicles spend offroad to a CSV file. Vehicle compliance with stop signs without going offroad is verified by the *stopSignWithNoOffroadViolations* test, ensuring that vehicles remain on the road when passing a stop sign check. The *goalWayPointWithTargetSpeedViolations* test checks whether vehicles reach a specified target speed at the goal waypoint. Statistical tests, presented through the *statisticsStatic* test, offer mean, median, mode, and standard deviation values for positional (x, y) and yaw data. Tailgating is assessed using four tailgating checking functions, with the *tailgatingCheck* test evaluating instances of tailgating based on varying coefficient values derived from Intel's RSS.I formulation formulation. Nine tests, including one with default coefficients and eight with high and low variations, are conducted. The corresponding coefficient values are detailed in the provided table in Table A.1.

Tailgating detection test one and the second tailgating detection test are similar, with the second test exploring all possible combinations of coefficients. This involves mixing and maxing high, low, and default values for response time, maximum front vehicle braking acceleration, maximum rear vehicle acceleration, and minimum rear vehicle braking acceleration. The third tailgating checking test focuses on reporting all instances of tailgating within a given dataset based on tailgating coefficients. Rather than providing time-interval-based results for every moment a vehicle tailgates another, this test offers an overall assessment. The fourth tailgating test check adopts an alternative format, iterating through all combinations of low, default, and high coefficient values for tailgating detection. Lastly, the *tailgatingExtraction* test systematically examines all OSM files and their respective datasets. It determines the position within a lanelet where a vehicle was found tailgating and reports the results, detailing the overall time spent tailgating as a proportion of the vehicle's total driving time. The outcomes are stored in three distinct CSV files for each OSM file, consolidating all datasets associated with the OSM file into these three CSV files. The files categorize tailgating instances in the intersection entry lanes, the intersection's interior lanes, and the exit lanes, as well as lanes not part of the intersection.

Naturalistic Test Cases - Whole Dataset Rule Violation Checks

In the final evaluation, all traffic rules are systematically checked for an entire OSM file and its respective datasets (trip recordings). The results can be conveniently outputted to CSV files using the nine violation checking functions. Specifically, the tests named *threeWayIntersectionWithRoundabout*, *threeWayIntersectionShort*, *threeWayIntersection*, *rulesFourWay*, *rulesFourWayDiamond*, *roundaboutThreeWay*, *roundaboutSixWay*, *roundaboutFourWay*, and *roundaboutFiveWay* correspond to nine OSM maps. With the exception of *DR_DEU_Roundabout_OF.osm* (a German map) and *DR_CHN_Roundabout_LN.osm* (a Chinese map), all other maps are American OSM files. The rules are scrutinized for all vehicles present in the interaction datasets for each map, utilizing the maps as references for vehicle positions along lanelets. Speed limit violations are assessed as a proportion of the total time spent exceeding the lanelet's speed limit, plus a delta threshold, measured against the total time spent driving (excluding periods of vehicle stoppage). Turning rules, such as right vehicle turn priority and oncoming vehicle priority over left turns, are evaluated, particularly when multiple vehicles arrive at an uS-SSRI intersection simultaneously. Stop violation checking is executed for all vehicles in the datasets, encompassing scenarios with roundabouts that feature stop signs.

Instances of vehicle offroad violations, measured as a proportion of the total time spent driving, are systematically reported for five distinct margins around the vehicles. These margins include the vehicle's centerpoint, the vehicle's full dimensions, and lengths/widths at 50%, 80%, and 90% from the center of the vehicle. The assessment of vehicle tailgating covers all vehicles and is presented as a proportion of the overall time spent driving. Stationary vehicles are flagged for tailgating if they are within a two-meter proximity of one another. The count includes the number of vehicles encountered and those that adhere to all the specified rules. Rule-violating vehicles are identified if they exceed the speed limit (with a speed threshold (θ_{speed}) set to zero), engage in at least one instance of tailgating (using Intel's RSS.I formulation) with *default* coefficient values (see the Table A.1), violate offroad rules using a distance of 50% of the vehicle's length and width from the centerpoint, or fail to stop before a stop sign or transfer the right of way correctly in uS-SSRI scenarios where the vehicle stops approximately simultaneously with another vehicle.

Naturalistic Test Cases - Signal Regulated Intersection Tests

A distinct program is dedicated to checking rules related to traffic light priority in S-RIs. Thirteen databases containing information for an S-RI in the Waterloo area undergo analysis. The evaluation includes a review of red light violations, specifically examining instances of left turns on red lights to verify the occurrence of valid permissive left turns. Permissive left turns are scrutinized from the moment a vehicle enters the intersection until it exits, assessing whether any vehicle needs to adjust its initial velocity in response to the reference vehicle executing the permissive left turn. Yellow light braking is assessed at the onset of the yellow light to determine if a vehicle could have come to a stop before reaching the intersection; crossing the intersection under these conditions is considered a rule violation. Factors such as the driver's reaction time are taken into account for permissive left-turn and yellow light-stopping checks. The program records the number of legal and illegal passes during green and yellow lights for each database, along with the count of illegal red-light passes. Additionally, it tracks the number of legal yellow light passes based on vehicle speed and stopping distance, as well as legal red light passes facilitated by permissive left turns.

4.3 Statistical Results and Analysis

4.3.1 Vehicle Positional Data

The following section depicts vehicle position data for a sample of the maps evaluated.

Yaw and Velocity One The accompanying diagram (see Figure 4.20) provides a visualization of the median velocity (represented by bubble size) relative to the median position for each vehicle in the dataset. The upper diagram illustrates the median velocity of vehicles over their median positions using distinct coloured circles, where each colour corresponds to a different vehicle. The size of the circles varies, with larger circles indicating a higher velocity magnitude. In the lower diagram, filled circles represent positive yaws, while hollow circles represent negative yaws. Larger circles indicate greater magnitudes for the corresponding yaw values.

Examining the top figure, it is evident that the median position for a vehicle is concentrated at the center of the intersection, even though many vehicles spend a considerable amount of time stopping along the entry gates. Vehicles positioned over the entry gates (as observed along the top and right entry gates) have smaller circle sizes, suggesting lower median velocities. Conversely, vehicles with median positions inside the exit gates (notably along the right exit gate) exhibit larger bubbles, indicating higher speeds. Generally, the proximity of a vehicle's median position to the exit gate correlates with larger circle sizes, signifying faster speeds, or less time spent stopping along the intersection. Conversely, circles closer to the entry gate are smaller, indicating slower speeds, possibly due to traffic congestion or frequent stops.

In the lower figure, hollow circles are noticeable along the right and top intersection entry gates, indicating that vehicles along these gates were more likely to have a negative yaw, implying they were facing downward during their trips. On the left entry gate, this suggests that many vehicles made left turns, causing them to face downwards. On the top entry gate, numerous vehicles maintained a straight trajectory, leading to a consistent negative yaw.

Along the bottom entry gate, filled circles dominate and are coloured orange, indicating that vehicles mostly had a positive yaw and a front-facing orientation. Vehicles on the bottom lane had limited options to shift to negative yaw, except for making a U-turn, as left, straight, and right turns all resulted in maintaining a general positive yaw.

At the center of the intersection, numerous black dots are present, indicating a negative yaw with a minimal magnitude. This suggests that many vehicles along the left entry gate turned left at the intersection, transitioning to a negative yaw. Near these tiny black dots, a small number of orange-filled circles can be observed, indicating trips where vehicles along the left entry gate turned right toward the top exit gate. In summary, vehicles entering from the left entry gate predominantly turn left, those entering from the bottom gate turn right, those from the top gate proceed straight, and those from the right gate turn right.

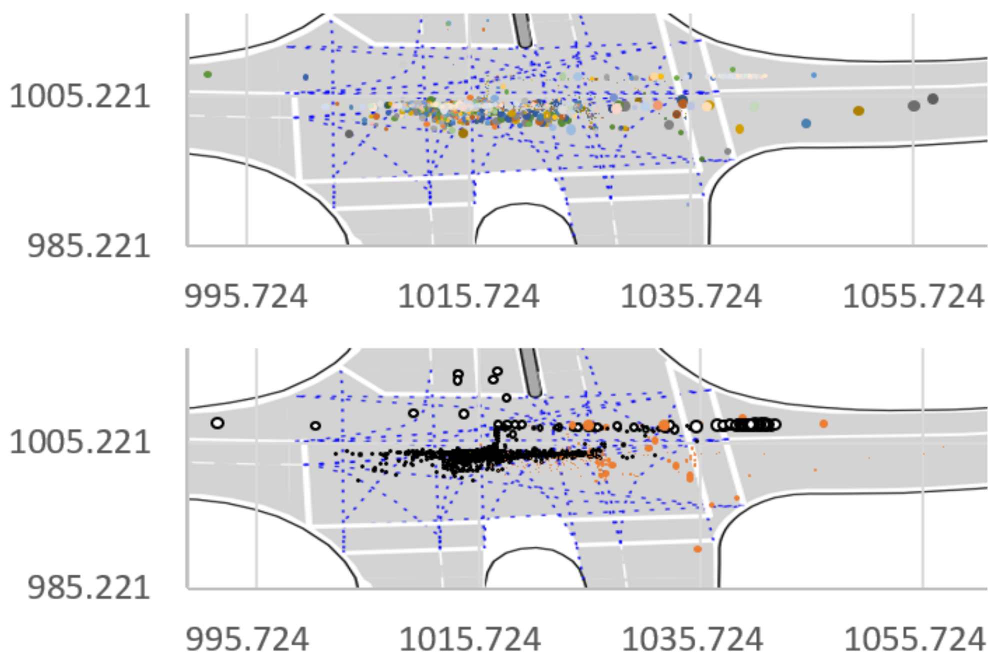


Figure 4.20: Median Yaw (Bottom Diagram) and Median Velocity (Top Diagram) Magnitudes Over Median Position Illustrations.

Yaw and Velocity Two In the diagram presented in Figure 4.21, the average velocity is visualized over the median position for each vehicle in the dataset. The top diagram illustrates the median velocity of vehicles over their median positions, utilizing various coloured circles where each colour corresponds to a different vehicle. The size of the circles varies, with larger circles indicating a higher velocity magnitude. The bottom diagram represents filled circles for positive yaws and black dots for negative yaws (where the magnitude is too low to distinguish a hollow interior). Larger circles denote greater yaw magnitudes.

Analyzing the top diagram reveals that the majority of vehicles have a median position along the intersection entry-gate stop line, while the remaining vehicles are positioned along the center of the intersection. Vehicles located along the center exhibit a smaller average velocity, indicating slower movement, likely due to congestion in that area. In contrast, vehicles positioned at the intersection entry gate show a larger average velocity, suggesting they were not stuck in traffic. The time spent stopped at the entry gate is counteracted by their driving velocity when calculating the median velocity.

Examining the bottom diagram, vehicles with positive median yaws (coloured circles) are primarily situated along the intersection entry gate. Since the entry gate offers only two options—moving straight or turning right—it is logical that no vehicles have a median negative yaw (indicating facing downward). Black dots represent negative yaws, likely indicating vehicles that turned left down the intersection entry gates at the bottom of the intersection.

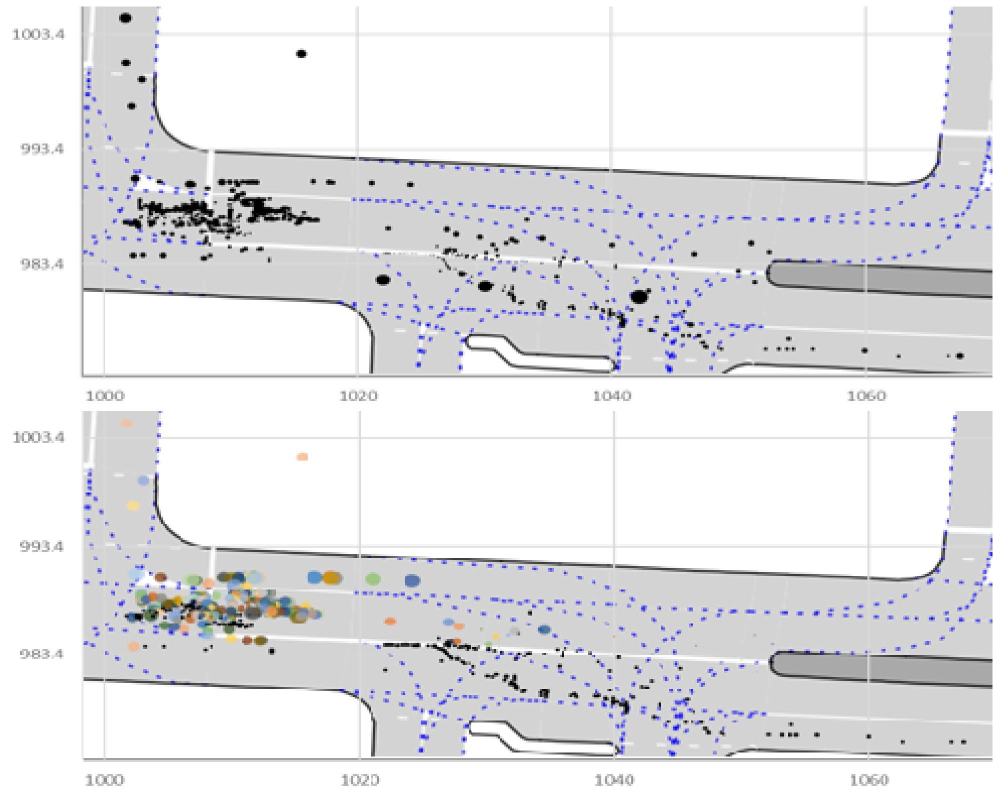


Figure 4.21: Yaw (Bottom Diagram) and Average Velocity (Top Diagram) Magnitudes Over Median Position Illustrations.

Vehicle Position Mapping In Figure 4.22, the visualization depicts black squares representing the median position, red diamonds representing the mean position, and yellow triangles representing the mode position of vehicles. On average, vehicles tend to cluster along the center of the intersection, with specific regions experiencing higher utilization than others.

In the top image, mode positions (yellow) are infrequent at the entry gates of intersections but are more prevalent in the middle of intersections. This suggests that vehicles are more likely to encounter traffic congestion rather than coming to a complete stop at a sign when navigating through the intersection. Mean positions (orange) are common at entry gates, indicating that vehicles often spend significant time around the entry gate of an intersection but are not entirely stationary across frames (unlike the yellow markers). Vehicles predominantly linger at entry gates but become stationary only when trapped in traffic jams along the center of the intersection. Mean positions (black) tend to occur around the center-right of the intersection, signifying that traffic is more frequently observed in those regions.

In the bottom image, mode positions (yellow) are prevalent in areas such as the middle of the three-way junction to the left or the exit gate to the bottom right, as these are common locations for traffic congestion to cause traffic to come to a standstill. Median positions (orange) are concentrated along the entry gate at the center of the intersection, indicating that many vehicles wait in this area before proceeding. Mean positions (black) are common either at the entry gate or the center of the intersection, highlighting these regions as the most heavily trafficked areas.

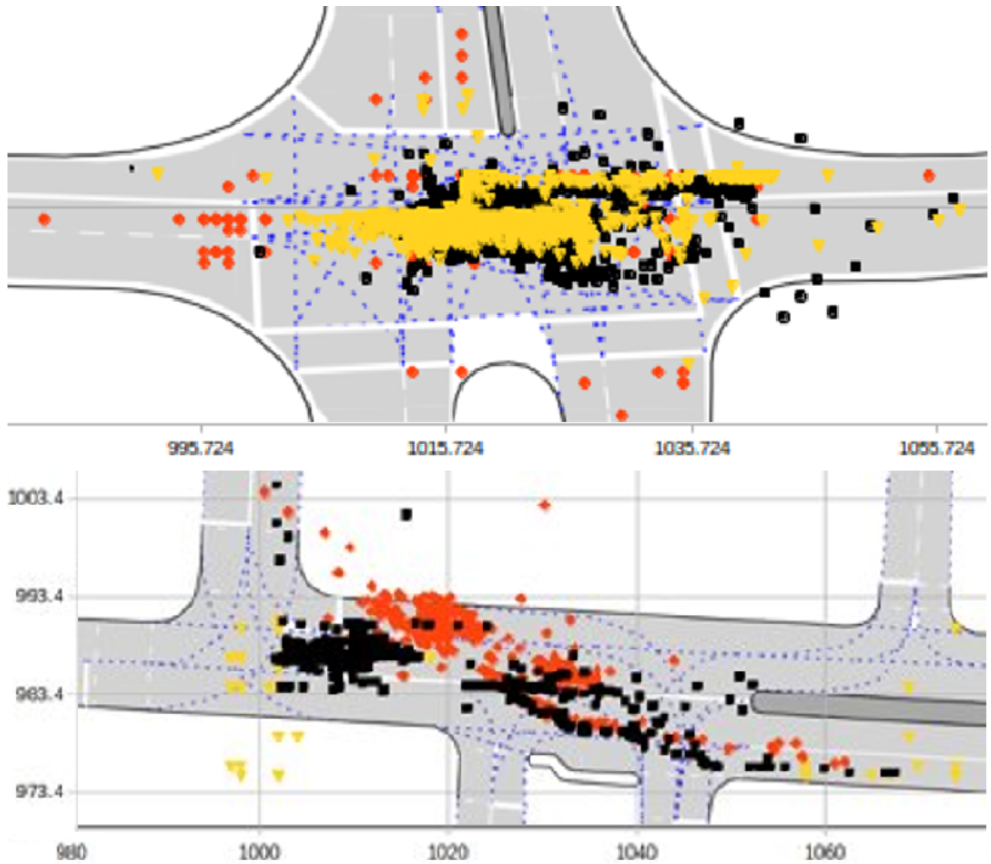


Figure 4.22: Vehicle Position Mapping for *DR_ USA_ Intersection_ MA* (Top Diagram) and *DR_ USA_ Intersection_ EP1.osm* (Bottom Diagram) Intersection Using Mean (red diamonds), Median (black squares), and Mode Values (yellow triangles).

4.3.2 Rule Violation Data - Box Plots

The figures in the following section (Figure 4.23, Figure 4.24, Figure 4.25, Figure 4.26, Figure 4.27, Figure 4.28) provide an overview of the aggregated results derived from the interaction dataset's nine folders, encompassing all associated recordings [5]. These folders consist of *DR_ USA_ Roundabout_ SR*, *DR_ CHN_ Roundabout_ LN*, *DR_ DEU_ Roundabout_ OF*, *DR_ USA_ Intersection_ MA*, *DR_ USA_ Roundabout_ EP1*, *DR_ USA_ Intersection_ EP0*, *DR_ USA_ Roundabout_ FT*, *DR_ USA_ Intersection_ GL*, and *DR_ USA_ Intersection_ EP*. The figure presents a box plot, where the bottom and top whiskers represent the minimum and maximum values, respectively. Values outside the whiskers are considered outliers. The box itself illustrates the first and third quartiles, denoted by the bottom and top edges. The central line within the box represents the median value. Dots to the left of each plot indicate individual data points used for generating the plots. Outliers are excluded based on Tukey's fences for outliers. The interquartile range

(IQR), the range between the first (Q1) and third (Q3) quartiles, is delineated by the tops and bottoms of the boxes [12]. The whiskers' minimum and maximum values are calculated as (Q1 - 1.5IQR) and (Q3 + 1.5IQR), respectively [13]. Images with a dotted line-shaped diamond figure over the rectangular boxes represent the standard deviation of the data. The dotted horizontal line in the image represents the mean value for the data in the distribution, and the top and bottom vertexes of the diamond represent the mean plus or minus one standard deviation of the population.

Boundary Violation Detection Rate Using Distance from Vehicle Center As can be seen in Figure 4.23, the rate of off-road vehicle encounters, expressed as a proportion of total driving time, is influenced by the consideration of the distance from the vehicle centerpoint to its corners, as observed in the interaction dataset. The findings highlight that extreme results, such as outliers with exceptionally high rates of offroad violations, are reduced when taking into account only half of the vehicle's length and width. Additionally, the occurrence of exceptionally high rates of offroad violation detections experiences a notable reduction when incorporating either half or none of the vehicle's length and width from the centerpoint in the calculations. However, when using only the centerpoint, significantly fewer instances of offroad violations are encountered compared with half the distance from the centerpoint, indicating that half the distance from the centerpoint is a more effective metric for offroad violation detection than just the centerpoint.

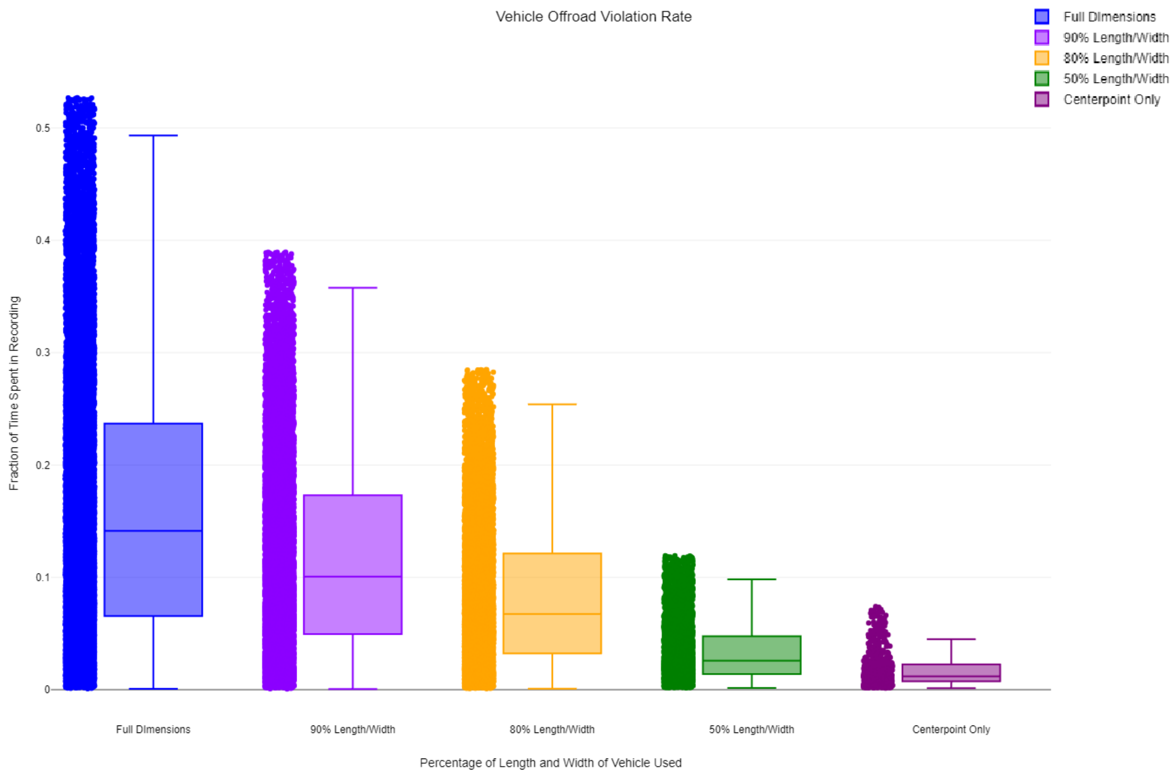


Figure 4.23: Offroad Violation Rate for Different Vehicle Dimensions as a Fraction of Total Time Spent in the Intersection Recording.

Speed Limit Violation Detection Rate From Speeding Delta Value As can be seen in Figure 4.24, the assessment of the proportion of time spent driving appropriately versus the time spent speeding can be influenced by the choice of delta values, which determine when a vehicle is deemed to have exceeded the speed limit. The analysis reveals that employing a delta value of five kilometres per hour leads to more consistent results with speeding detection. Interestingly, increasing the delta to ten kilometres per hour does not result in a substantial change in the detection rate, meaning it is too high to be necessary for detecting speed limit violations.

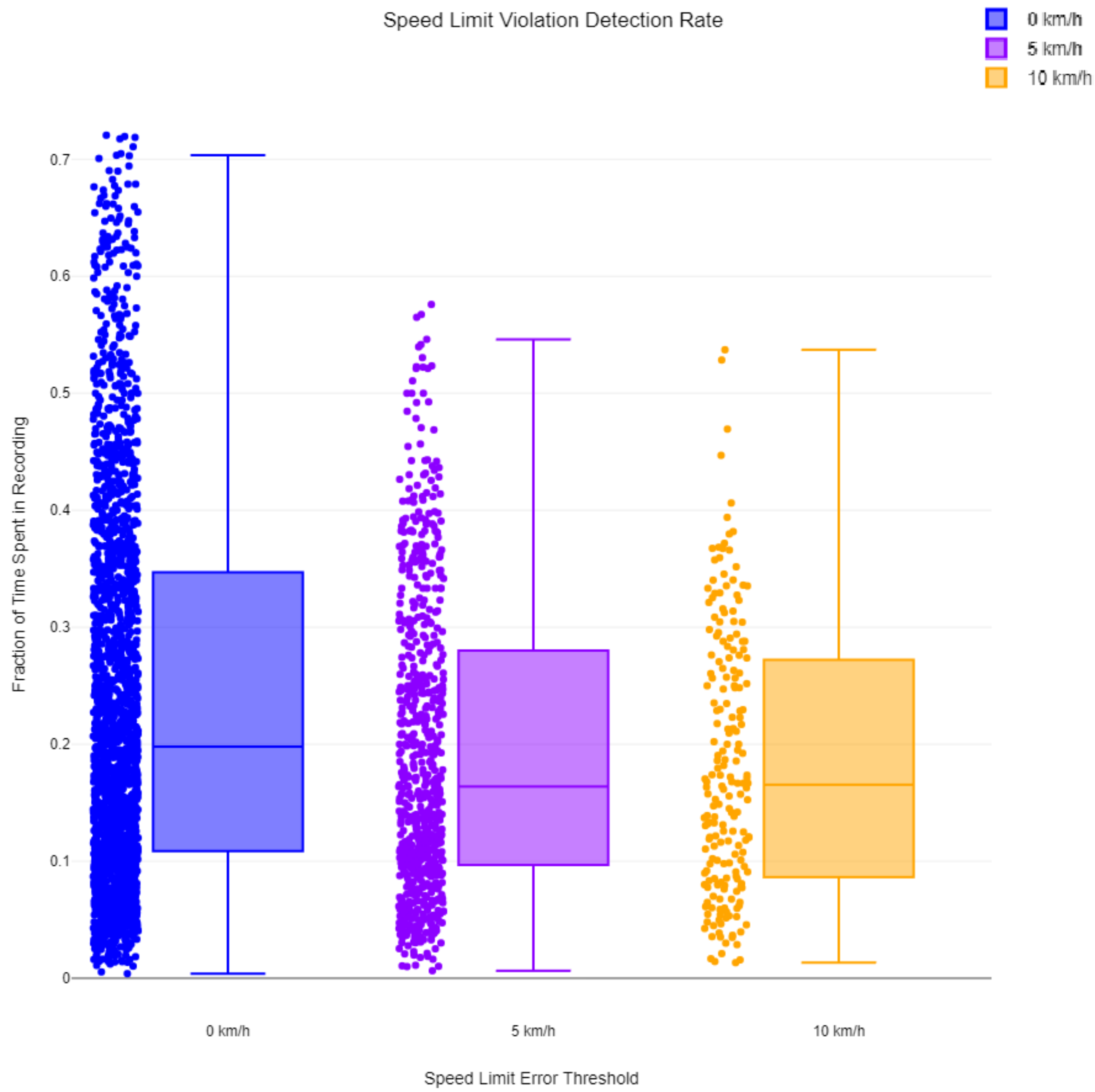


Figure 4.24: Speed Limit Violation Detection Rate Per Delta Value as a Fraction of Total Time Spent Driving.

Turning Direction After Stopping Along Unsignalized Stop-Sign Regulated Intersection (uS-SSRI) As can be seen in Figure 4.25, the direction a vehicle turns at an intersection slightly changes how long it waits at the stop sign. Vehicles that turn left do not wait nearly as long as vehicles that turn right or go straight. This difference in waiting times may be attributed to factors such as visibility over the road and the ability to assess oncoming traffic more efficiently when making a left turn.

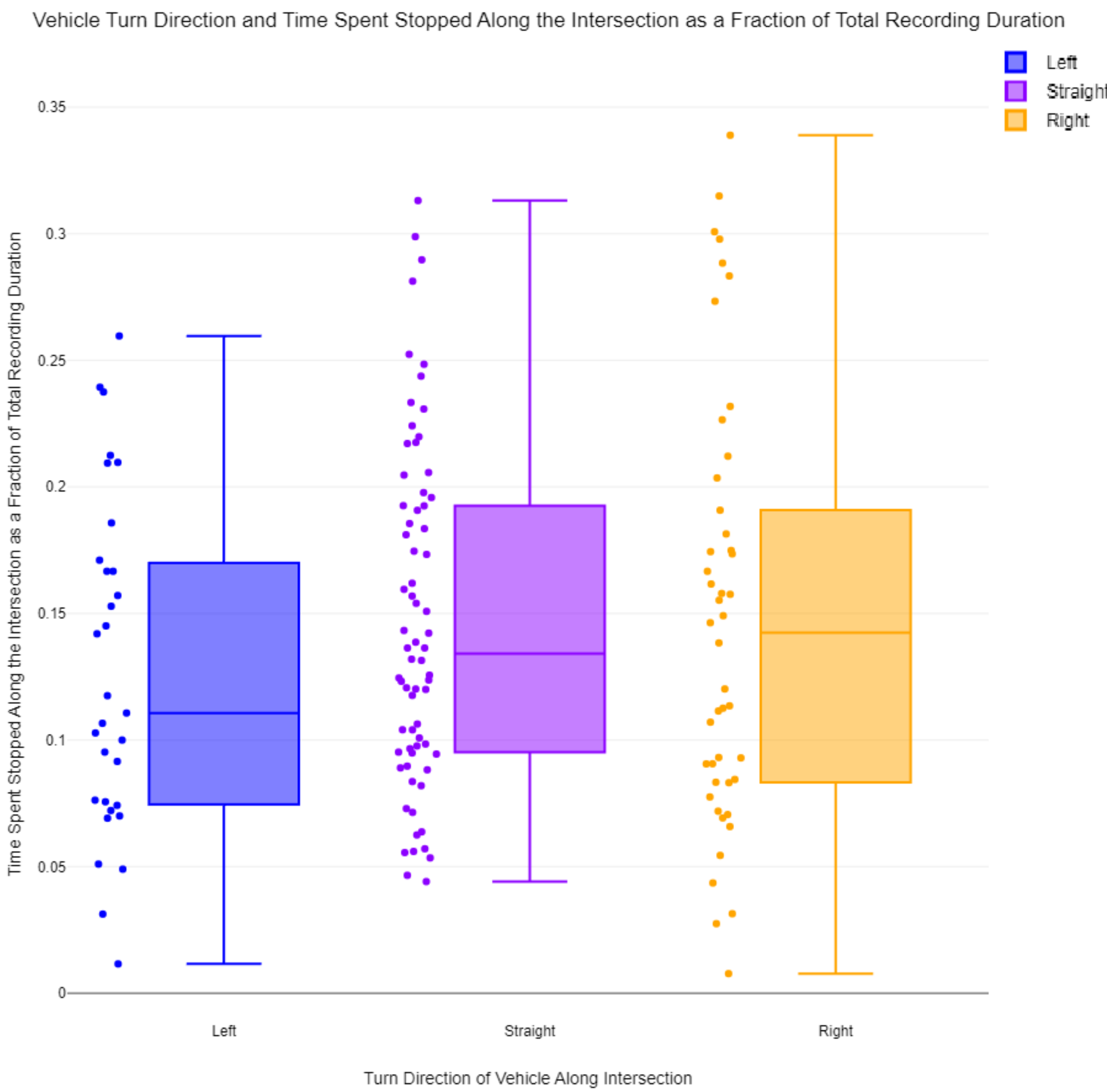


Figure 4.25: The relationship between the turn direction of a vehicle and the time spent stopped along the intersection is depicted as a fraction of the total time of the recording.

Rate of Tailgating Detection as a Proportion of Total Time Driving as a Fraction of Tailgating Detection Coefficient Values As can be seen in Figure 4.26, the rate of tailgating as a proportion of the total time spent driving is most sensitive to changes in vehicle response time. Changes in rear vehicle maximum acceleration have the lowest overall impact on tailgating detection rates. Specifically, a low maximum rear vehicle acceleration, a high minimum rear vehicle braking acceleration, and a high maximum rear vehicle maximum acceleration all result in approximately the same rate of tailgating occurrences as the default coefficient values. Variations in response time were shown to have a great impact on tailgating violation detection. Furthermore, a low maximum front vehicle vehicle braking acceleration was shown to have a relatively small impact on tailgating violation detection results.

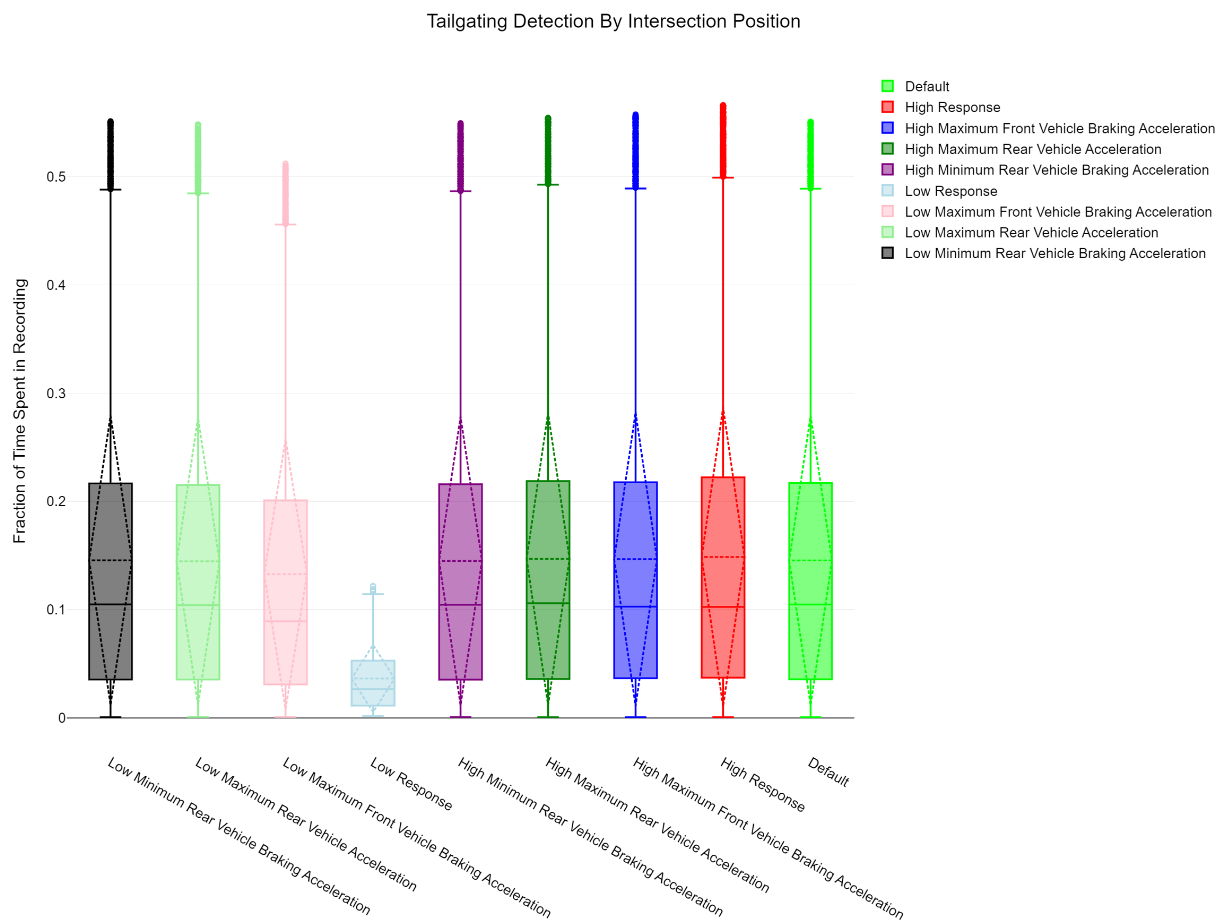


Figure 4.26: Tailgating Detection Rate per Tailgating Coefficient Value as a Fraction of Total Time Spent in the Intersection Recording.

Tailgating Detection Based Off Vehicle Position Along Intersections As can be seen in Figure 4.27, vehicles can be positioned along the entrance, inside an intersection, or along the exit lanes and lanes away from the intersection. It is observed that vehicles are most likely to engage in tailgating along the exit and other segments of an intersection while tailgating inside an intersection is relatively rare. Vehicles at the entry of intersections tend to tailgate at almost the same rate as those that exiting the intersection or are along some other segment of the intersection.

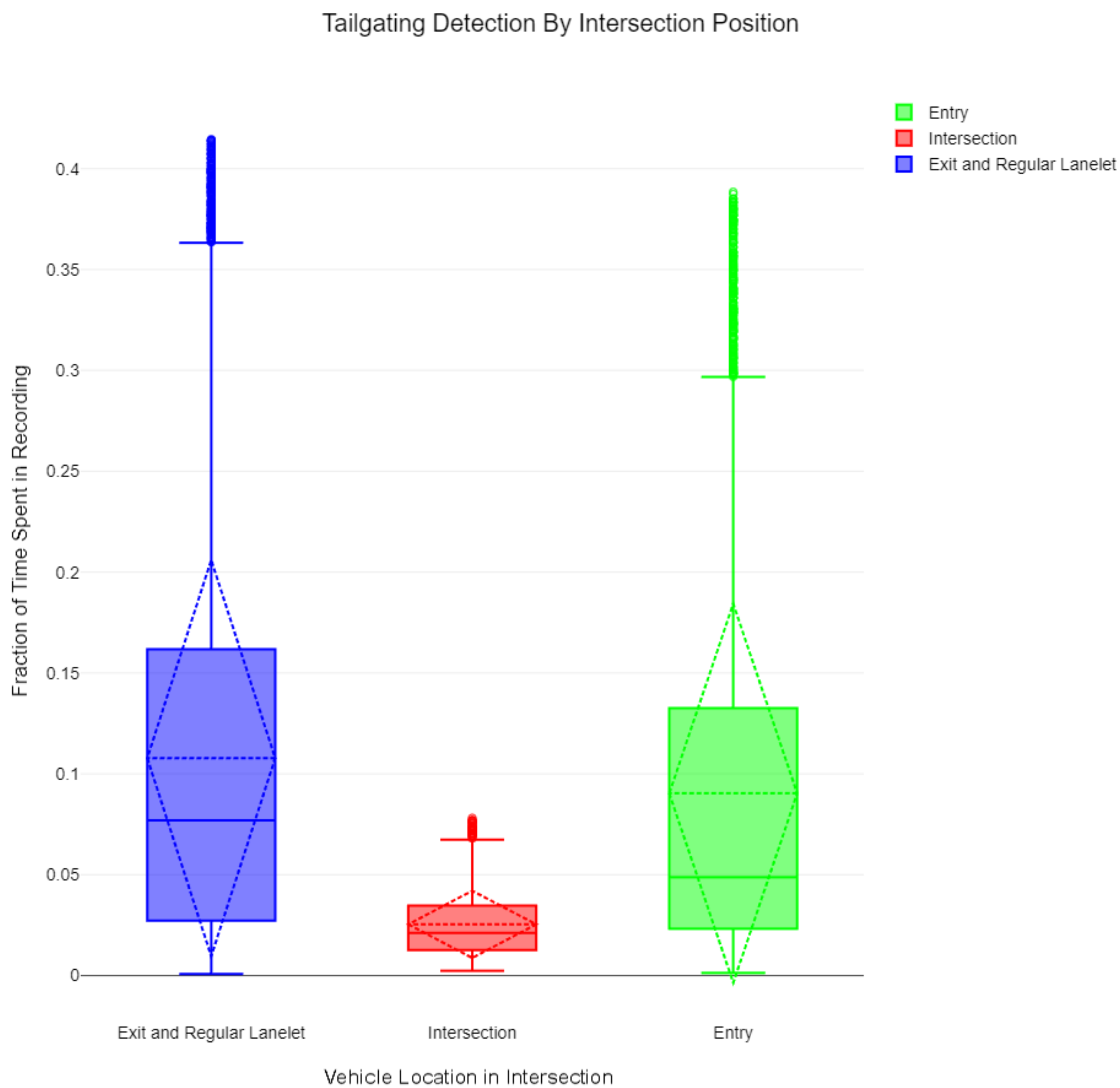


Figure 4.27: Tailgating Detection Rate per Position in Intersection as a Fraction of Total Time Spend in the Intersection Recording.

Rate of Time Waiting at Stop Sign as a Fraction of Threshold Values for Stop Detection As can be seen in Figure 4.28, threshold values for the speed of a vehicle and the distance of a vehicle to a stop line can be employed to assess whether a vehicle has come to a stop before a stop sign appropriately. Adjusting these threshold values can influence the results, with a low-speed threshold or a high-distance threshold tightening extreme outcomes. Conversely, a high-speed threshold or a low-distance threshold may lead to an increased rate of stop detection. The choice of threshold values plays a crucial role in determining the effectiveness and sensitivity of the stop detection algorithm.

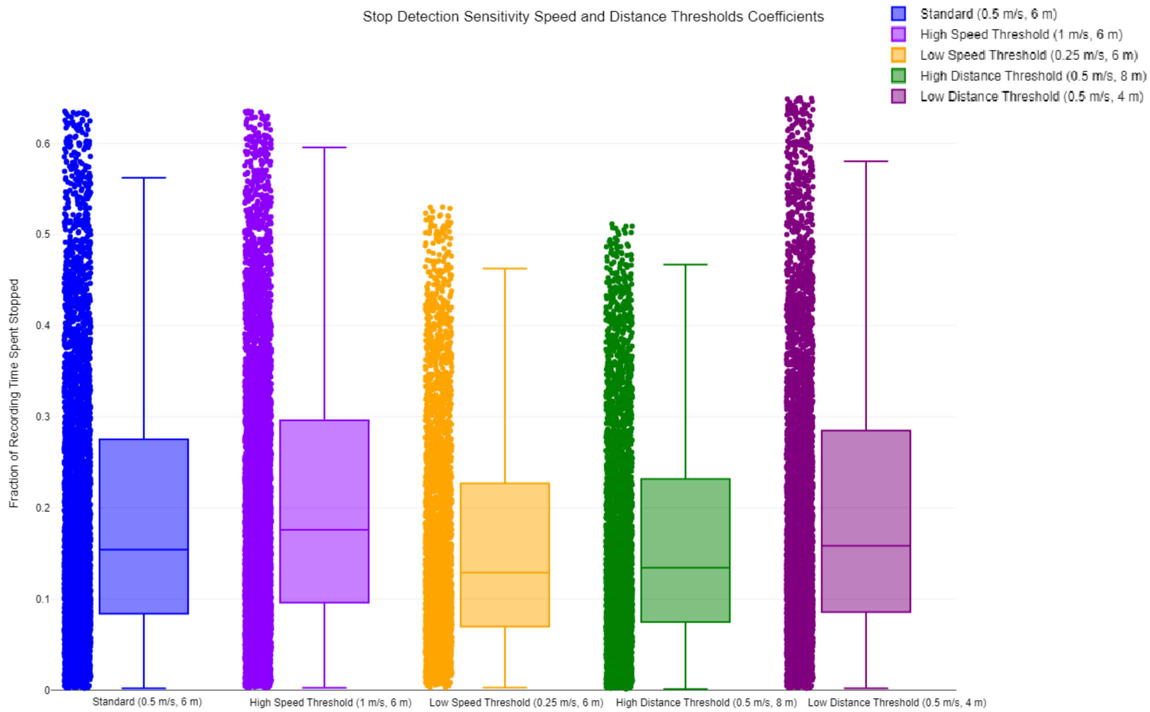


Figure 4.28: Fraction of Time Stopped in an Intersection by Stop-line Detection Coefficients.

Vehicle Speed During Stop-Sign Violations As shown in Figure 4.29, the speed of vehicles at the moment of a stop-sign violation varies based on whether the violation occurred due to a rolling stop or not. Stop-sign violation checks for vehicles utilize standard threshold values for speed (0.5m/s) and distance (6m) from the stop line. According to the findings of Wen et al. (2021), vehicles can be categorized into five different clusters of violations. Cluster one includes vehicles that come to a full stop (0.01m/s or less), cluster two involves vehicles performing partial rolling stops (0.05m/s to 0.96m/s), cluster three consists of vehicles executing rolling stops (0.96m/s and 1.95m/s), cluster four comprises vehicles slowing down without stopping (1.95m/s and 3.30m/s), and cluster five includes vehicles running through the stop sign (equal to or greater than 3.31m/s). In Figure 4.29, a minimum speed limit violation threshold of 0.5m/s is used, rather than the 0.01m/s used in the article, and cluster one is not included in the box-plot depicted under Figure 4.29.

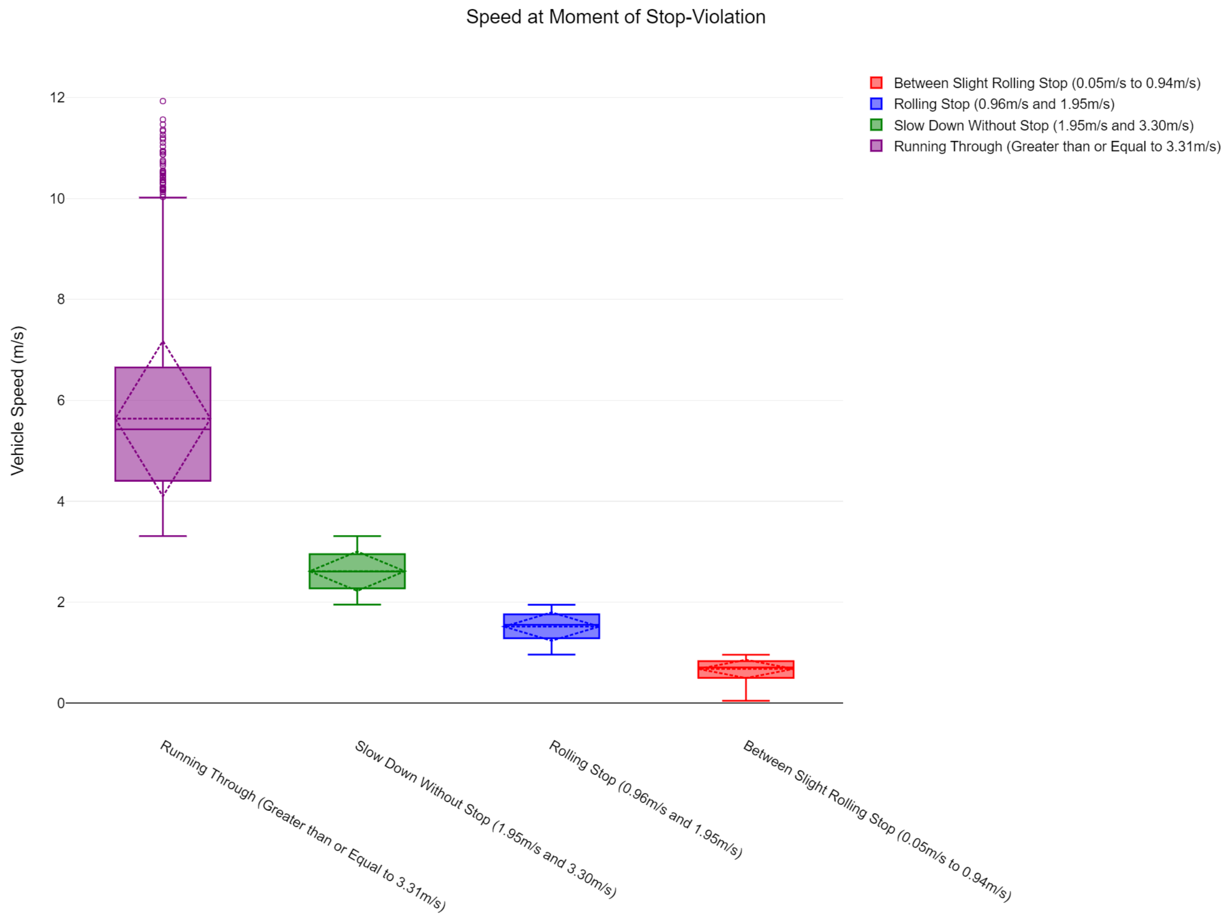


Figure 4.29: Vehicle Speed at Moment of Stop-Sign Violation Given Standard ($\theta_{speed} = 0.5\text{m/s}$, $\theta_{\Delta} = 6\text{m}$) Stop Violation Coefficients.

4.3.3 Rule Violation Data - Results

The following subsection depicts results for rule violation checking along with statistical results for rule violations.

Tailgating Detection Heatmaps The provided heatmaps, illustrated in Figure 4.30, Figure 4.31, Figure 4.32, and Figure 4.33, depict the occurrence rate of tailgating based on variations in vehicle coefficients, specifically, the vehicle response time, maximum front vehicle braking acceleration, maximum rear vehicle acceleration, and minimum rear vehicle braking acceleration. The first heatmap shows tailgating instances as a result of changes in two coefficients from their *"default"* values, while the second heatmap illustrates tailgating instances resulting from a single coefficient deviating from its *"default"* value. The *"default"* column in the heatmaps (Figure 4.31 and Figure 4.33) represents the duration of tailgating occurrences in the recording when all coefficients are set to the *"default"* values as indicated in Table A.1.

In Figure 4.30 and Figure 4.32, tailgating violations are presented as a percentage difference from the default parameter settings (639.9s in Figure 4.30 and 1064.1s in Figure 4.32 of tailgating detected across all vehicles). Each row and column signify a modification in one of the four coefficients utilized for tailgating violation detection. The default row and column denote no change from the default coefficient values. The *"Frames Encountered"* column/row in the heatmap represents the duration (in seconds) of tailgating violations detected in the recording when coefficients are set to the values indicated in the table when only a single variable (the one highlighted under the row or column) is changed (see Table A.1 for details).

The following images (Figure 4.30 and Figure 4.31) are for the three-way-intersection named *DR_ USA_ Intersection_ EP1.osm*.

DR_USA_Intersection_EP1/vehicle_tracks_004.csv	Frames Encountered (s)	Default	High Response	High Maximum Front Vehicle Braking Acceleration	High Maximum Rear Vehicle Acceleration	High Minimum Rear Vehicle Braking Acceleration	Low Response	Low Maximum Front Vehicle Braking Acceleration	Low Maximum Rear Vehicle Acceleration	Low Minimum Rear Vehicle Braking Acceleration
Frames Encountered (s)		639.900	731.700	668.600	653.200	637.600	11.700	466.200	633.800	644.100
Default	639.900	1.000								
High Response	731.700	1.143	1.143							
High Maximum Front Vehicle Braking Acceleration	668.600	1.045	1.148	1.045						
High Maximum Rear Vehicle Acceleration	653.200	1.021	1.146	1.049	1.021					
High Minimum Rear Vehicle Braking Acceleration	637.600	0.996	1.143	1.044	1.017	0.996				
Low Response	11.700	0.018	NA	0.027	0.027	0.016	0.018			
Low Maximum Front Vehicle Braking Acceleration	466.200	0.729	1.024	NA	0.762	0.725	0.005	0.729		
Low Maximum Rear Vehicle Acceleration	633.800	0.990	1.143	1.042	NA	0.987	0.016	0.723	0.990	
Low Minimum Rear Vehicle Braking Acceleration	644.100	1.007	1.144	1.046	1.025	NA	0.020	0.736	0.996	1.007

Figure 4.30: Tailgating Violations are Measured in Terms of their Percent Difference from the *default* Parameter Settings in the *DR_ USA_ Intersection_ EP1.osm* Dataset.

FILE	Frames Encountered (s)	Default	High Response	High Maximum Front Vehicle Braking Acceleration	High Maximum Rear Vehicle Acceleration	High Minimum Rear Vehicle Braking Acceleration	Low Response	Low Maximum Front Vehicle Braking Acceleration	Low Maximum Rear Vehicle Acceleration	Low Minimum Rear Vehicle Braking Acceleration
./Data/DR_USA_Intersection_EP1/vehicle_tracks_004.csv	639.9	1.000	1.143	1.045	1.021	0.996	0.018	0.729	0.990	1.007
./Data/DR_USA_Intersection_EP1/vehicle_tracks_005.csv	648.9	1.000	1.062	1.050	1.022	0.994	0.008	0.740	0.984	1.010
./Data/DR_USA_Intersection_EP1/vehicle_tracks_000.csv	355.9	1.000	1.096	1.049	1.017	0.995	0.000	0.799	0.992	1.004
./Data/DR_USA_Intersection_EP1/vehicle_tracks_003.csv	272.7	1.000	1.098	1.074	1.041	0.992	0.005	0.693	0.987	1.016
./Data/DR_USA_Intersection_EP1/vehicle_tracks_001.csv	188.9	1.000	1.084	1.060	1.026	0.998	0.000	0.689	0.993	1.010
./Data/DR_USA_Intersection_EP1/vehicle_tracks_002.csv	181.5	1.000	1.165	1.112	1.063	0.988	0.000	0.677	0.917	1.012
ALL FILES	2545.100	1.000	1.104	1.057	1.027	0.994	0.008	0.731	0.983	1.009

Figure 4.31: Total Recorded Tailgating Violation Duration in Seconds for Vehicle Recordings in the *DR_ USA_ Intersection_ EP1.osm* Dataset as Percent of the *Default* Coefficient Value.

The following images (Figure 4.32 and Figure 4.33) are for the four-way-intersection named *DR_ USA_ Intersection_ MA*.

DR_USA_Intersection_MA/vehicle_tracks_001.csv	Frames Encountered (s)	Default	High Response	High Maximum Front Vehicle Braking Acceleration	High Maximum Rear Vehicle Acceleration	High Minimum Rear Vehicle Braking Acceleration	Low Response	Low Maximum Front Vehicle Braking Acceleration	Low Maximum Rear Vehicle Acceleration	Low Minimum Rear Vehicle Braking Acceleration
Frames Encountered (s)		1064.100	1181.200	1123.900	1082.200	1058.800	5.000	874.400	1058.800	1072.100
Default	1064.100	1.000								
High Response	1181.200	1.110	1.110							
High Maximum Front Vehicle Braking Acceleration	1123.900	1.056	1.140	1.056						
High Maximum Rear Vehicle Acceleration	1082.200	1.017	1.118	1.067	1.017					
High Minimum Rear Vehicle Braking Acceleration	1058.800	0.995	1.108	1.054	1.009	0.995				
Low Response	5.000	0.005	NA	0.014	0.011	0.004	0.005			
Low Maximum Front Vehicle Braking Acceleration	874.400	0.822	1.022	NA	0.849	0.817	0.003	0.822		
Low Maximum Rear Vehicle Acceleration	1058.800	0.995	1.107	1.053	NA	0.992	0.004	0.816	0.995	
Low Minimum Rear Vehicle Braking Acceleration	1072.100	1.008	1.113	1.059	1.024	NA	0.008	0.831	1.001	1.008

Figure 4.32: Tailgating Violations are Measured in Terms of their Percent Difference from the *default* Parameter Settings in the *DR_ USA_ Intersection_ MA* Dataset.

FILE	Frames Encountered (s)	Default	High Response	High Maximum Front Vehicle Braking Acceleration	High Maximum Rear Vehicle Acceleration	High Minimum Rear Vehicle Braking Acceleration	Low Response	Low Maximum Front Vehicle Braking Acceleration	Low Maximum Rear Vehicle Acceleration	Low Minimum Rear Vehicle Braking Acceleration
./Data/DR_USA_Intersection_MA/vehicle_tracks_001.csv	1064.100	1.000	1.110	1.056	1.017	0.995	0.005	0.822	0.995	1.008
./Data/DR_USA_Intersection_MA/vehicle_tracks_002.csv	1031.200	1.000	1.097	1.074	1.019	0.995	0.002	0.831	0.994	1.008
./Data/DR_USA_Intersection_MA/vehicle_tracks_016.csv	889.800	1.000	1.140	1.080	1.030	0.996	0.000	0.773	0.991	1.005
./Data/DR_USA_Intersection_MA/vehicle_tracks_014.csv	592.100	1.000	1.131	1.087	1.022	0.995	0.008	0.811	0.992	1.009
./Data/DR_USA_Intersection_MA/vehicle_tracks_003.csv	570.600	1.000	1.100	1.065	1.019	0.995	0.002	0.775	0.994	1.007
./Data/DR_USA_Intersection_MA/vehicle_tracks_008.csv	487.800	1.000	1.092	1.057	1.008	0.997	0.006	0.837	0.996	1.003
./Data/DR_USA_Intersection_MA/vehicle_tracks_000.csv	481.400	1.000	1.161	1.107	1.025	0.996	0.004	0.741	0.994	1.013
./Data/DR_USA_Intersection_MA/vehicle_tracks_011.csv	477.600	1.000	1.148	1.101	1.026	0.994	0.011	0.811	0.992	1.011
./Data/DR_USA_Intersection_MA/vehicle_tracks_005.csv	429.700	1.000	1.175	1.118	1.031	0.995	0.012	0.801	0.995	1.015
./Data/DR_USA_Intersection_MA/vehicle_tracks_015.csv	428.600	1.000	1.104	1.078	1.013	0.996	0.000	0.803	0.996	1.009
./Data/DR_USA_Intersection_MA/vehicle_tracks_020.csv	401.300	1.000	1.067	1.048	1.012	0.996	0.013	0.817	0.995	1.005
./Data/DR_USA_Intersection_MA/vehicle_tracks_013.csv	381.600	1.000	1.095	1.080	1.017	0.995	0.000	0.795	0.993	1.007
./Data/DR_USA_Intersection_MA/vehicle_tracks_012.csv	361.700	1.000	1.057	1.046	1.007	0.998	0.000	0.854	0.997	1.003
./Data/DR_USA_Intersection_MA/vehicle_tracks_009.csv	347.100	1.000	1.114	1.083	1.014	0.997	0.014	0.845	0.994	1.005
./Data/DR_USA_Intersection_MA/vehicle_tracks_018.csv	328.300	1.000	1.217	1.163	1.028	0.991	0.002	0.748	0.983	1.015
./Data/DR_USA_Intersection_MA/vehicle_tracks_006.csv	259.400	1.000	1.153	1.123	1.024	0.994	0.013	0.794	0.992	1.007
./Data/DR_USA_Intersection_MA/vehicle_tracks_019.csv	215.200	1.000	1.190	1.120	1.032	0.992	0.009	0.759	0.991	1.016
./Data/DR_USA_Intersection_MA/vehicle_tracks_007.csv	206.300	1.000	1.137	1.089	1.030	0.995	0.001	0.797	0.995	1.012
./Data/DR_USA_Intersection_MA/vehicle_tracks_010.csv	200.900	1.000	1.069	1.031	1.008	0.999	0.005	0.868	0.999	1.002
./Data/DR_USA_Intersection_MA/vehicle_tracks_021.csv	149.900	1.000	1.217	1.142	1.023	0.987	0.000	0.665	0.986	1.012
./Data/DR_USA_Intersection_MA/vehicle_tracks_004.csv	144.600	1.000	1.120	1.100	1.015	0.996	0.001	0.786	0.997	1.008
./Data/DR_USA_Intersection_MA/vehicle_tracks_017.csv	130.200	1.000	1.167	1.085	1.015	0.992	0.000	0.764	0.994	1.012
ALL FILES	9579.400	1.000	1.123	1.082	1.020	0.995	0.005	0.802	0.993	1.008

Figure 4.33: Total Recorded Tailgating Violation Duration in Seconds for Vehicle Recordings in the *DR_ USA_ Intersection_ MA* Dataset as Percent of the *Default* Coefficient Value.

Rule Violation Statistics In Table 4.2, a comprehensive compilation of rule violations is presented, stemming from nine distinct OSM driving scenarios and spanning a total of one hundred and sixty-three vehicle track recordings. The data is categorized into two groups: one for maps featuring stop-regulated intersections and another for maps incorporating roundabouts. Notably, it is observed that numerous vehicles failed to come to a complete stop at stop signs situated at traffic junctions with roundabouts, despite the physical presence of stop signs at these roundabout junctions. Additionally, there are instances of overlap across certain maps. For instance, the *"USA_ Roundabout_ EP"* intersection map includes a roundabout with a uS-SSRI and satisfies the criteria for both categories outlined in Table 4.2. Offroad violations are assessed for vehicles by considering half their length and width from their centerpoints along their recorded trips. Speed limits are evaluated using a speed limit violation threshold value of zero kilometres per hour. The total count of instances for *Left Before Right Turns Needed* and *Left Turn Violators*

Needed reflects the number of occasions when a vehicle was required to yield to another vehicle at an intersection after both vehicles had come to a stop at approximately the same time. For the following figure, tailgating was assessed using Intel's RSS.I formulation for vehicle following behaviour only.

	Recording Type		
	Has Roundabout	Has Intersection	All
Total Vehicles	10446	15589	15589
Non-Rule-Violating Vehicles	1561	9594	9594
Stop-Violators	8313	4090	11977
Stop-Sign-Encounters	9472	6873	16345
Left Before Right Violators	21	155	155
Left Before Right Turns Needed	67	609	609
Left Turn Violators	11	60	60
Left Turn Violators Needed	14	111	111
Vehicle Tailgating Violators	791	1893	1572
Speed Limit Violators	1119	829	1355
Offroad Violators	1740	578	578

Table 4.2: Rule Violation Statistics for Intersection Rules.

Stop Sign Violation Rates As presented in Table 4.3, stop sign stop violations were observed at a rate of approximately 73.28% across all traffic junctions and 59.51% specifically at intersections. A notable number of vehicles were found to commit stop sign violations, often involving rolling stops at intersections. At roundabouts, around 87.86% of vehicles were observed engaging in stop sign violations, with many neglecting to adhere to stop signs when no other vehicles were present within the roundabout. The overall results for all tailgating occurrences at intersections were recorded at 59.51%, a figure comparable to a study (67.6%) that analyzed intersection traffic across 528 hours of recordings at 142 sites with 31212 vehicles observed [15]. Additional findings from another study indicated that only 20.2% of vehicles comply with stop signs, which is closer to the overall stop sign violation rate (73.28%) reported in Table 4.3 for both intersections and roundabouts [16]. More so, a study demonstrated that the stop sign violation rate varies with roadway traffic volume, ranging from a 75% violation rate on roads with around 1000 to 2000 vehicles per day to less than 50% violation rates on roads with over 5000 vehicles per day [17]. The traffic light violation rate for vehicles in the thesis falls within the range of 59.51% for intersections and 73.28% for both intersections and roundabouts, aligning with the results from the study. Finally, according to a study referenced by Wen et al. (2021), a field investigation in the US disclosed that 52% of the 25,660 drivers did not come to a stop at stop signs along five S-RIs, closely aligning with the 59.51% error rate reported in the thesis. Notably, the study by the authors themselves presented conflicting findings compared to the aforementioned literature sources, suggesting that only 11% of vehicles executed full stops (with a speed threshold of 0.01m/s), while 37% performed slight rolling stops (with a speed threshold between 0.01m/s and 0.96m/s) [14].

Dataset	Stop Sign Violations	Stop Sign Encounters	Rule Violation Rate
DR_USA_Intersection_MA	1092	2887	0.3782
DR_CHN_Roundabout_LN	120	120	1.0000
DR_USA_Roundabout_EP	426	667	0.6387
DR_DEU_Roundabout_OF	831	845	0.9834
DR_USA_Intersection_EP1	297	522	0.5690
DR_USA_Intersection_GL	1894	2827	0.6700
DR_USA_Roundabout_SR	814	832	0.9784
DR_USA_Intersection_EP0	381	637	0.5981
DR_USA_Roundabout_FT	6122	7008	0.8736
ALL DATASETS	11977	16345	0.7328
All Roundabouts	8313	9472	0.8776
All Intersections	4090	6873	0.5951

Table 4.3: Stop Sign Stop Violation Rate for Vehicles at Traffic Junctions.

When classifying stop-sign violations into four categories, similar to the five mentioned by Wen et al. (2021), the results are presented in Table 4.4. Out of 16,345 stop-sign encounters found in Table 4.3, approximately 2.07%, 8.52%, 14.93%, and 47.3% of vehicles were observed to engage in slight rolling stops, rolling stops, slowdowns without stopping, and direct runs through the stop sign, respectively. Since not all speeds for vehicles were recorded due to the ending times of recordings cutting off the last frame of the video, only 11,905 violator speeds were recorded, contrasting with the total 11,977 stop-sign violators found in Table 4.3.

The stop-sign violation rates differ from the findings of Wen et al. (2021), who reported that 5% of vehicles ran through stop signs, 14% of vehicles slowed down without stopping, 33% of vehicles made a rolling stop, 37% of vehicles made slight rolling stops, and 11% of vehicles made full stops. Considering that Wen et al. (2021) observed differences in the total number of vehicles making full stops before the stop line compared to other literature sources such as Pietrucha (1989), Liu and Zhang (2022), Mounce (1981), it may be attributed to variations in the datasets used.

The median speeds in meters per second for slight rolling stops were 0.71, for rolling stops were 1.55, for slowdowns without stops were 2.61, and for running through stop signs were 5.43. In contrast, the findings by Wen et al. (2021) indicated median speeds of 0.53, 1.38, 2.44, and 3.94 for the corresponding clusters. While the speeds for slight rolling stops, rolling stops, and slowdowns without stops were roughly similar, the speeds for full stops were much higher in the thesis compared to the paper, suggesting potential differences in traffic conditions between the two driving settings.

Clustering	Approach Speed at Stop Sign (m/s)								Sample size (n)
	Minimum	Maximum	Median	Q1	Q3	Mean	Standard Deviation	Mode	
Between Slight Rolling Stop (0.05m/s to 0.96m/s)	0.0461	0.9584	0.7051	0.4997	0.8317	0.6774	0.1813, 95% CI [0.1693,0.1950]	0.4737,0.4820,0.4891,0.4892,0.5816,0.6369,0.6671,0.6673,0.7116,0.7661,0.8543,0.8898,0.9308,0.9562 (appears 2 times)	338
Rolling Stop (0.96m/s and 1.95m/s)	0.9603	1.9490	1.5515	1.2853	1.7600	1.5168	0.2810, 95% CI [0.2710,0.2919]	1.3654,1.5393,1.8708,1.8867 (appears 3 times)	1393
Slow Down Without Stop (1.95m/s and 3.30m/s)	1.9520	3.3095	2.6081	2.2742	2.9514	2.6137	0.3901, 95% CI [0.3795,0.4015]	2.503 (appears 4 times)	2441
Running Through (Greater than or Equal to 3.31m/s)	3.3104	11.9302	5.4259	4.4053	6.6520	5.6374	1.5384, 95% CI [1.5146,1.5631]	4.4403 (appears 5 times)	7733

Table 4.4: Stop Sign Stop Violation Rate for Vehicles at Traffic Junctions.

Stop Sign Yield Rates As depicted in Table 4.5, vehicles exhibit yield violations at uS-SSRIs when multiple vehicles reach the intersection simultaneously. The data shows that stop sign yield violations occurred at a rate of 25.45% for yields where a vehicle was required to yield to another vehicle stopped to its right at an intersection. Additionally, yield violations were observed at a rate of around 54.05% for vehicles that needed to yield to a vehicle in an oncoming lane before executing a left turn across the intersection. As of the thesis writing, no literature source providing a basis for comparing the results of yield violations at stop-regulated intersections was identified.

Dataset	Left Before Right Violators	Left Before Right Turns Needed	Fraction of Left Before Right Turns as Violations	Fraction of Left Before Right Turns as Violations	Fraction of Left Before Right Turns as Violations	Fraction of Left Before Right Turns as Violations
DR_USA_Intersection_MA	62	362	0.1713	31	64	0.4844
DR_CHN_Roundabout_LN	NA	NA	NA	NA	NA	NA
DR_USA_Roundabout_EP	21	67	0.3134	11	14	0.7857
DR_DEU_Roundabout_OF	NA	NA	NA	NA	NA	NA
DR_USA_Intersection_EP1	33	88	0.4045	8	17	0.4706
DR_USA_Intersection_GL	3	3	1.0000	0	0	0.0000
DR_USA_Roundabout_SR	NA	NA	NA	NA	NA	NA
DR_USA_Intersection_EP0	36	89	0.4045	10	16	0.6250
DR_USA_Roundabout_FT	NA	NA	NA	NA	NA	NA
ALL DATASETS	155	609	0.2545	60	111	0.5405

Table 4.5: Stop Sign Yield Violation Rate for Vehicles at Traffic Junctions.

Speed Limit Violation Statistics Detailed information regarding speed limit violations, encompassing violation thresholds of zero, five, and ten kilometres per hour, along with mean, median, mode data, and minimum/maximum values, is available in Table 4.6. The results reveal that drivers who do violate speed limits do so by exceeding the speed limit by zero, five, or ten kilometres per hour do so for a median of 20.22%, 16.67%, and 16.67% of their total trip duration, respectively. This contrasts with previous findings by

Perez et al. (2021), indicating median driver speeds (for all drivers - not just those who engage in speeding) of approximately 35%, 20%, and 7% for speed limit thresholds of 0-mph, 5-mph, and 10-mph. Furthermore, Haus et al. (2022) estimated that the percentage of time all drivers engaged in speeding violations was around 44 for speed limit thresholds in the [5, 10)-mph range. Notably, the literature suggests slightly higher instances of speeding compared to the program's results when speed limit violators are only the vehicles taken into consideration. When we look at all vehicles, including those that did not break any speed limit rules, we can see that the results show that speeding is very rare in the data when compared with data in the literature.

	Parameter	Speed Limit - 0 km/h Speed Limit Exceedance	Speed Limit - 5 km/h Speed Limit Exceedance	Speed Limit - 10 km/h Speed Limit Exceedance
Total Duration of Trip Recordings	Total Duration of all Vehicle Trips (Seconds)	454648.8000		
	Total Duration of Moving Vehicles (Seconds)	294050.7000		
	Total Duration of Speed Limit Violations (Seconds)	7159.7000	2112.3000	567.7000
Fraction the of Duration of the Trip Violating Speed Limits - Average of All Vehicles - Outliers Included	Sample Size (n)	25327	25327	25327
	Min	0.0000	0.0000	0.0000
	Q1	0.0000	0.0000	0.0000
	Median	0.0000	0.0000	0.0000
	Q3	0.0000	0.0000	0.0000
	Max	1.0000	1.0000	1.0000
	Mean	0.0016	0.0058	0.0200
	Mode	0 (appears 25119 times)	0 (appears 24620 times)	0 (appears 23307 times)
Fraction the of Duration of the Trip Violating Speed Limits - Average of Speed Limit Violating Vehicles Only - Outliers Included	Sample Size (n)	2020	707	208
	Min	0.0039	0.0064	0.0133
	Q1	0.1111	0.0986	0.0885
	Median	0.2022	0.1667	0.1667
	Q3	0.3556	0.2918	0.2762
	Max	1.0000	1.0000	1.0000
	Mean	0.2502	0.2065	0.1927
	Mode	1 (appears 12 times)	0.3333 (appears 6 times)	0.166667 (appears 3 times)
	Population Standard Deviation (σ) and Standard Error (SE)	σ = 0.18, SE = 0.004	σ = 0.15, SE = 0.0056	σ = 0.13, SE = 0.0092
Fraction the of Duration of the Trip Violating Speed Limits - Average of Speed Limit Violating Vehicles Only - Outliers Excluded	Sample Size (n)	1985	693	204
	Min	0.0039	0.0064	0.0133
	Q1	0.1088	0.0968	0.0864
	Median	0.1979	0.1638	0.1654
	Q3	0.3470	0.2801	0.2722
	Max	0.7206	0.5759	0.5372
	Mean	0.2388	0.1956	0.1830
	Mode	0.3333 (appears 11 times)	0.3333 (appears 6 times)	0.1667 (appears 3 times)
	Population Standard Deviation (σ) and Standard Error (SE)	σ = 0.16, SE = 0.0036	σ = 0.13, SE = 0.0048	σ = 0.11, SE = 0.0078

Table 4.6: Rule Violation Statistics for Intersection Rules.

Tailgating Statistics The comprehensive statistics for vehicle tailgating behaviour are presented in Table 4.7, which includes data for tailgating when vehicles use following other vehicles (as indicated by a violation of Intel's RSS.I formulation) only as a metric for detection, when they use distance from other vehicles while stopped only as a metric, and finally, when they use both following other vehicles and distance from other vehicles while stopped as metrics. When outliers are included, and all vehicles (including those that did not engage in tailgating) are considered, the mean duration of tailgating is roughly 8.05% of the trip duration when checking both following and stopped vehicles. When only vehicles that engaged in tailgating are considered, and outliers are excluded, the median duration of tailgating for all vehicles is 11.03% of the overall duration of the trip, and the mean is 15.17%. Moreover, it is evident that out of a total sample size of 25,327 vehicles, 11,210 vehicles were involved in tailgating—11042 while following and 1073 while stopped. This implies that 44.26% of vehicles engaged in tailgating (43.61% while following and 4.24% while stopped) for at least one frame of the recording. In a study examining tailgating behaviour on urban highways, it was found that 61.2% of vehicles engage in tailgating during rush hours, and 39.2% engage in tailgating during non-rush hours [20]. The tailgating rates observed in the thesis (43.61%) are comparable to the rates during non-rush hours in the study. It is essential to note that the thesis primarily focuses on intersectional traffic, while the study concentrates on urban highways. It's important to note that, due to a small error margin for a vehicle to be considered stopped, some instances of tailgating while a vehicle is stopped also count as instances of tailgating while a vehicle is following another. Consequently, the *Total Duration of Tailgating Violations (Seconds)* for *Tailgating - All* is lower than the duration of both *Tailgating - Following Only* and *Tailgating - Stopped Only*.

	Parameter	Tailgating - All	Tailgating - Following Only	Tailgating - Stopped Only
Total Duration of Trip Recordings	Total Duration of all Vehicle Trips (Seconds)	454648.8	454648.8	454648.8
	Total Duration of Tailgating Violations (Seconds)	52681.0	49782.0	6799.0
Fraction the of Duration of the Trip Engaged in Instances of Tailgating - Average of All Vehicles - Outliers Included	Sample Size (n)	25327	25327	25327
	Min	0.0000	0.0000	0.0000
	Q1	0.0000	0.0000	0.0000
	Median	0.0000	0.0000	0.0000
	Q3	0.1020	0.0943	0.0000
	Max	1.0000	1.0000	0.9287
	Mean	0.0805	0.0774	0.0069
	Mode	0 (appears 13644 times)	0 (appears 13774 times)	0 (appears 24198 times)
Fraction the of Duration of the Trip Violating Speed Limits - Average of Speed Limit Violating Vehicles Only - Outliers Excluded	Sample Size (n)	11210	11046	1073
	Min	0.0008	0.0008	0.0009
	Q1	0.0377	0.0357	0.0508
	Median	0.1103	0.1048	0.1055
	Q3	0.2276	0.2170	0.1964
	Max	0.5678	0.5506	0.4492
	Mean	0.1517	0.1455	0.1332
	Mode	0.1111 (appears 27 times)	0.1111,0.1667 (appears 26 times)	0.2 (appears 5 times)

Table 4.7: Tailgating Statistics for Traffic in the Interaction Dataset.

Signal Regulated Intersection Results The comprehensive statistics for vehicle behaviour at S-RI are presented in Table 4.8, Table 4.9, and Table 4.10. The data includes instances where vehicles pass through traffic lights during red, yellow, or green signals. Additionally, the statistics cover the utilization of permissive left turns at a green light without slowing an oncoming vehicle and passing through a yellow light without rule infringement when it is not feasible to come to a stop before the traffic light stop line.

Database Number	769	770	771	775	776	777	778	779	780	781	782	783	784	ALL
Red Light Encounters	53	60	55	73	84	47	60	47	65	79	66	45	27	761
Stops Before Red Lights	40	46	41	57	60	32	45	34	44	64	53	32	16	564
Red Light Passes	13	14	14	16	24	15	15	13	21	15	13	13	11	197
Red Light Violations	1	3	3	1	4	3	1	2	7	3	0	2	1	31
Right Turn on Red Light With Stop to Yield to Traffic	4	4	4	6	7	6	4	6	6	3	8	7	4	69
Right Turn on Red Lights Without Stopping	8	7	7	9	13	6	10	5	8	9	5	4	6	97
Right Turns on Red Lights that Slowed Down Stright Moving Vehicles from the Right	3	1	4	11	7	7	6	1	3	7	4	4	4	62
Fraction of Right Turns on Red Lights to Slow Traffic	0.3	0.1	0.4	0.7	0.4	0.6	0.4	0.1	0.2	0.6	0.3	0.4	0.4	0.37
Red Light Violation Rate as a Propotion of Red Light Encounters	0.17	0.17	0.18	0.14	0.20	0.19	0.18	0.15	0.23	0.15	0.08	0.13	0.26	0.17

Table 4.8: Signal Regulated Intersection Database Results - Red Lights.

Database Number	769	770	771	775	776	777	778	779	780	781	782	783	784	ALL
Yellow Light Encounters	1	8	7	12	17	13	6	9	3	9	17	7	6	115
Total Yellow Light Passes	1	5	6	11	12	7	4	8	2	7	13	4	4	84
Stops Before Yellow Lights	0	3	1	1	5	6	2	1	1	2	4	3	2	31
Illigal Yellow Light Passes	1	2	5	10	7	1	2	7	1	5	9	1	2	53
Legal Yellow Light Passes due to Vehicle Speed	0	2	0	0	3	3	2	0	1	1	2	2	1	17
Yellow Light Violation Rate as a Proportion of Yellow Light Encounters	1.00	0.25	0.71	0.83	0.41	0.08	0.33	0.78	0.33	0.56	0.53	0.14	0.33	0.46

Table 4.9: Signal Regulated Intersection Database Results - Yellow Lights.

Database Number	769	770	771	775	776	777	778	779	780	781	782	783	784	ALL
Green Light Encounter	197	195	160	218	191	141	185	204	191	222	218	142	183	2447
Green Light Passes	188	184	150	206	185	136	172	195	180	218	208	131	174	2327
Permissive Left Turns Causing Oncomign Traffic Delay	4	9	8	12	3	4	10	7	9	4	6	5	4	85
Legal Permissive Left Turns	5	2	2	0	3	1	3	2	2	0	4	6	5	35
Total Permissive Left Turns	9	11	10	12	6	5	13	9	11	4	10	11	9	120
Proportion of Permissive Left Turns that Cause Traffic Delays	0.44	0.8	0.80	1	0.50	0.8	0.77	0.8	0.82	1	0.60	0.5	0.44	0.71

Table 4.10: Signal Regulated Intersection Database Results - Green Lights.

In Table 4.8, there were a total of 761 encountered red lights, with 31 instances of running red lights, resulting in a violation rate of 4.07%. However, when incorporating instances of right turns at red lights causing slowdowns for incoming straight-moving vehicles, the overall red light violation rate increases to 17%. Specifically, 37% of right turns at red lights contribute to traffic slowdowns. The violation rates for yellow lights and permissive left turns on green lights are 46% and 71%, respectively. Yellow light violations were analyzed using a vehicle braking acceleration of $10f/s^2$ (roughly 0.31g or $3m/s^2$) [21]. Vehicles correctly stopped at red lights 69 times but failed to do so 97 times, leading to a 58% violation rate for right turn stop violations on red lights. A study by Choupani (2020) found that out of 1085 drivers, 51% ran yellow lights when restrictive yellow light laws were applied, and 12% ran red lights. Another study by Zegeer and Cynecki (1985) reported that right turns on red lights violated other vehicles and pedestrians 23.4% of the time, with 56.9% of vehicles failing to stop before turning on the red. While the rates of yellow and red light violations in the literature data align with the results, and the rates of right turn stop violations in the literature data are very similar to the results, the rates of right turns on red lights causing slowdowns are higher in the results when compared with the study.

Chapter 5

Conclusions

In addressing the challenges of developing traffic rule verification systems for traffic junctions and evaluating them using naturalistic datasets, various obstacles were confronted. These challenges encompassed the need for formulating comprehensive and detailed traffic rules, as well as defining and formalizing ontologies to effectively model the complex dynamics of the vehicle situation within the traffic junctions.

1. The operationalization of rule violation detection was realized through the development of algorithms and code implementation, complemented by data extraction procedures for various traffic rule concepts. The implemented algorithms covered a diverse range of scenarios, encompassing vehicle halting at stop signs (Stop Sign Checking Algorithm), identification of traffic light violations, including permissive-green, red, and yellow light infractions (Red Light Violation Algorithm, Yellow Light Violation Algorithm, and Permissive Green Light Violation Algorithm), assessments of illegal movements after stops at signal-regulated intersections (SSRIs) with considerations for illegal right-before-left movement priority, illegal intersection priority movements, and illegal left turns (Right Before Left Movement Priority Algorithm, Intersection Priority Algorithm, and Left Turns Algorithm). Additionally, the algorithms addressed turn directions (Turn Direction Algorithm), speed limits (Speed Limits Algorithm), waypoints (Waypoints Algorithm), collisions (Collisions Algorithm), target speeds (Target Speed Algorithm), offroad violations (Offroad Violations Algorithm), and tailgating violations (Tailgating Algorithm). Each algorithm was designed to extract relevant data, contributing to a comprehensive system for the detection and assessment of rule violations in diverse traffic scenarios.

2. The traffic rule checking system was evaluated through a range of methods, including artificial and naturalistic test cases. Further, the severity of rule violations in naturalistic data was analyzed using histograms, box plots, and statistics detailing rule violations in relation to their associated threshold values. Notably, rates of vehicle violations concerning offroad occurrences were deemed most realistic when the vehicle boundaries were defined as half the vehicle's length and width from its centerpoint. However, challenges were acknowledged, particularly with point-to-lanelet matching

in the Lanelet2 interface, which has known issues with false negatives, possibly leading to instances of detected tailgating being false positives. Detailed numbers of speed limit violations, offroad violations, tailgating violations, stop sign violations, and illegal intersectional movement right-of-way violations are provided in Table 4.2. Traffic light violation results, outlined in Table 4.8, Table 4.9, and Table 4.10, reveal that the majority of yellow light passes were considered illegal for vehicles with a braking acceleration of 0.31g, and most attempted green-light permissive-left turns were deemed unlawful as they interfered with oncoming traffic [21].

3. Threshold values for rule violation parameters have been determined to be valid based on the geometry of intersections. In the test cases examined, a vehicle stop speed error threshold of half a meter per second and a distance from an intersectional stop line threshold of six meters resulted in no observed false positives or negatives for either stop violation detection or stop-regulated intersectional illegal movement violation detection at signal-regulated intersections (SSRI). The results for different thresholds for stopping velocity and stopping distance in stop sign violation detection are presented in Figure 4.9 and Figure 4.10. For tailgating thresholds, Figure 4.26, Figure 4.32, and Figure 4.33 illustrate that the default tailgating coefficient distribution (refer to Table A.1) results in a more moderate rate of tailgating detection compared to the high and low parameter settings. These findings highlight the importance of selecting appropriate threshold values for different rule violation parameters based on specific intersection geometries.

The analysis of results reveals optimal thresholds for the detection of violations while minimizing outliers, covering various scenarios such as Signal Regulated Intersections (S-RIs) and Stop-Sign-Regulated Intersections (uS-SSRIs). The dataset encompasses instances of speed limit violations, offroad occurrences, tailgating, and violations related to intersectional stop signs. Notably, data from uS-SSRIs indicates that a significant portion of vehicles tends to violate at least one rule at intersections, with brief occurrences of tailgating and stop sign violations at roundabouts being prominent.

Potential future developments include the comparison of vehicles to assess the justification of speed limit violations by comparing a vehicle's behaviour to others on the road. Lane merging behaviour could be examined to identify rule violations resulting from unsafe lane changes. Future research may explore different junction setups, incorporating elements such as speed limit signs, yield signs, lane closures, and construction rules. Additionally, there is potential for integrating the proposed system with vehicle planners to determine valid plans through trajectory validators.

Implementing a violation detection system in real-time computing allows for the continuous monitoring of intersectional violations. Additionally, the integration of vehicle data generators into a big-data system enables the collection of extensive vehicle data. This comprehensive system has the potential to determine optimal rule violation thresholds across various traffic junction geometries and setups, providing valuable insights into traffic behaviour and aiding in the refinement of traffic rule enforcement strategies.

The implementation has certain limitations, such as assuming standard operational junctions and lacking consideration for potential collision sites or construction zones. It also assumes non-impaired drivers and doesn't account for lane geometry beyond the OpenStreetMap (OSM) file. Offroad violations may include cases where vehicles enter parking lots on the map. Challenges in the implementation include difficulties in distinguishing turn directions and their boundaries, as well as the reliance on traffic junction geometry for determining turn directions. Distinguishing left, right, or straight vehicle movements involve more than geometric angles or trajectories, extending to considerations of both vehicle trajectories and traffic junction directions. Furthermore, for Signal Regulated Intersections (S-RIs), there is a lack of visualization tools or comparison methods to validate and cross-reference the data within the WISE Labs databases for S-RIs, posing challenges in result interpretation and verification.

The implementation demonstrates successful features, including naturalistic driving tools for analyzing vehicle speed limits, offroad violations, and tailgating detection systems that effectively report vehicle offences. It accurately registers traffic violations such as stop and intersectional illegal movement right-of-way violations, as well as Signal Regulated Intersection (S-RI) traffic light violations. Configurable parameters, including tailgating, stopping, and speeding delta threshold configurations, provide flexibility. The implemented tests can generate box plots or roadway diagrams, enabling visualization of vehicle behaviours and the analysis of various coefficients for vehicle rule violations.

The implemented rule violation detection system has various potential use cases. It can be employed to determine optimal coefficients for rule violations, aiding in the identification of values that optimize detection for different traffic junction setups. The system is valuable for assessing rule violations within a dataset, offering a quantitative measure of the rate at which drivers violate traffic laws. Additionally, the rule violation detection system is applicable for evaluating user-generated trajectories to detect potential rule violations in specific driving scenarios.

References

[1] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving, November 2018. URL https://github.com/fzi-forschungszentrum-informatik/Lanelet2. https://github.com/fzi-forschungszentrum-informatik/Lanelet2.

[2] Sebastian Maierhofer, Paul Moosbrugger, and Matthias Althoff. Formalization of intersection traffic rules in temporal logic. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, page 1135–1144. IEEE Press, 2022. doi: 10.1109/IV51971.2022.9827153. URL https://doi.org/10.1109/IV51971.2022.9827153.

[3] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars, 2018.

[4] Rodrigo Queiroz, Divit Sharma, Ricardo Caldas, Krzysztof Czarnecki, Sergio García, Thorsten Berger, and Patrizio Pelliccione. A driver-vehicle model for ads scenario-based testing, 2022.

[5] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. *arXiv:1910.03088 [cs, eess]*, sep 2019.

[6] WISE Lab. Waterloo multi-agent traffic dataset: intersection, Sep 2022. URL https://uwaterloo.ca/waterloo-intelligent-systems-engineering-lab/datasets/waterloo-multi-agent-traffic-dataset-intersection. https://uwaterloo.ca/waterloo-intelligent-systems-engineering-lab/datasets/waterloo-multi-agent-traffic-dataset-intersection.

[7] Colin Shea-Blymyer and Houssam Abbas. A deontic logic analysis of autonomous systems' safety. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*. ACM, apr 2020. doi: 10.1145/3365365.3382203. URL https://doi.org/10.1145%2F3365365.3382203.

[8] Rodrigo Queiroz, Thorsten Berger, and Krzysztof Czarnecki. Geoscenario: An open dsl for autonomous driving scenario representation. In *2019 IEEE Intelligent Vehicles*

Symposium (IV), page 287–294. IEEE Press, 2019. doi: 10.1109/IVS.2019.8814107. URL https://doi.org/10.1109/IVS.2019.8814107.

[9] Frédéric Bouchard, Sean Sedwards, and Krzysztof Czarnecki. A rule-based behaviour planner for autonomous driving. In *Rules and Reasoning: 6th International Joint Conference on Rules and Reasoning, RuleML+RR 2022, Berlin, Germany, September 26–28, 2022, Proceedings*, page 263–279, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-21540-7. doi: 10.1007/978-3-031-21541-4_17. URL https://doi.org/10.1007/978-3-031-21541-4_17.

[10] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, page 719–726. IEEE Press, 2017. doi: 10.1109/IVS.2017.7995802. URL https://doi.org/10.1109/IVS.2017.7995802.

[11] Hart Solomon, Alireza Hadayeghi, Brian Malone, Shervin Shahriari, Brian Applebee, and Russell Brownlee. *Operational Practice*, page 29–29. Ministry of Transportation, 2012.

[12] Alexander B. Holmes, Barbara Illowsky, and Susan L. Dean. *Introductory business statistics*. OpenStax, Rice University, 2018.

[13] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[14] Xiamei Wen, Liping Fu, Ting Fu, Jessica Keung, and Ming Zhong. Driver behavior classification at stop-controlled intersections using video-based trajectory data. *Sustainability*, 13(3):1404, 2021. doi: 10.3390/su13031404.

[15] Martin Pietrucha. *Motorist compliance with Standard Traffic Control Devices*. U.S. Dept. of Transportation, Federal Highway Administration, Research, Development, and Technology, 1989.

[16] Chenhui Liu and Wei Zhang. Exploring the stop sign running at all-way stop-controlled intersections with the shrp2 naturalistic driving data. *Journal of Safety Research*, 81: 190–196, 2022. doi: 10.1016/j.jsr.2022.02.010.

[17] John M. Mounce. *Driver Compliance with Stop-Sign Control at Low-Volume Intersections*, page 30–37. National Academy of sciences, 1981.

[18] Miguel A. Perez, Edie Sears, Jacob T. Valente, Wenyan Huang, and Jeremy Sudweeks. Factors modifying the likelihood of speeding behaviors based on naturalistic driving data. *Accident Analysis & Prevention*, 159:106267, 2021. doi: 10.1016/j.aap.2021.106267.

[19] Samantha H. Haus, Pnina Gershon, Bruce Mehler, and Bryan Reimer. Speeding behavior when using automation: A descriptive analysis of naturalistic driving data. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 66(1): 1471–1475, 2022. doi: 10.1177/1071181322661226.

[20] Miao Song, Jyh-Hone Wang, and Valerie Maier-Speredelozzi. Tailgating on urban highways and possible means to mitigate tailgating behavior. *Journal of Transportation of the Institute of Transportation Engineers*, 2(1):71–89, Oct 2011.

[21] Yaping Zhang, Chuanyun Fu, and Liwei Hu. Yellow light dilemma zone researches: a review. *Journal of Traffic and Transportation Engineering (English Edition)*, 1(5):338–352, 2014. ISSN 2095-7564. doi: https://doi.org/10.1016/S2095-7564(15)30280-4. URL https://www.sciencedirect.com/science/article/pii/S2095756415302804.

[22] Abdoul-Ahad Choupani. Assessing drivers' compliance with restrictive yellow traffic lights in a developing country. *Transportation Research Record: Journal of the Transportation Research Board*, 2675(6):38–50, 2020. doi: 10.1177/0361198120973659.

[23] Charles V. Zegeer and Michael J. Cynecki. Determination of motorist violations and pedestrian-related countermeasures related to right-turn-on-red. *Transportation research record*, 1010:16–28, 1985. ISSN 0361-1981. URL http://dx.doi.org/.

[24] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, page 1672–1679. IEEE Press, 2018. ISBN 978-1-7281-0321-1. doi: 10.1109/ITSC.2018.8569929. URL https://doi.org/10.1109/ITSC.2018.8569929.

[25] Sushmitha Ramireddy, Vineethreddy Ala, Ravishankar KVR, and Arpan Mehar. Acceleration and deceleration rates of various vehicle categories at signalized intersections in mixed traffic conditions. *Periodica Polytechnica Transportation Engineering*, 49(4): 324–332, 2021. doi: 10.3311/pptr.16796.

[26] Daniel V. McGehee, Elizabeth N. Mazzae, and G.H. Scott Baldwin. Driver reaction time in crash avoidance research: Validation of a driving simulator study on a test track. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 44 (20), 2000. doi: 10.1177/154193120004402026.

[27] Joyoung Lee, Byungkyu (Brian) Park, Kristin Malakorn, and Jaehyun (Jason) So. Sustainability assessments of cooperative vehicle intersection control at an urban corridor. *Transportation Research Part C: Emerging Technologies*, 32:193–206, 2013. doi: 10.1016/j.trc.2012.09.004.

[28] R. G. Mortimer, L. Sagel, J. D. Campbell, C. M. Jorgeson, and R. W. Murphy. *Brake Force requirement study: Driver-vehicle braking performance as a function of brake system design variables*. National Highway Safety Bureau, 1970.

[29] 2019. URL https://www.nsc.org/getmedia/a46d07cb-faf1-4572-8317-661e7f77ef7a/instructor-admin-reference-guide.pdf.

[30] Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry Yershov, Scott Pendleton, James Fu, and Emilio Frazzoli. Liability, ethics, and culture-aware behavior specification using rulebooks, 2019.

[31] Miguel A. Perez, Edie Sears, Jacob T. Valente, Wenyan Huang, and Jeremy Sudweeks. Factors modifying the likelihood of speeding behaviors based on naturalistic driving data. *Accident Analysis & Prevention*, 159:106267, 2021. ISSN 0001-4575. doi: https://doi.org/10.1016/j.aap.2021.106267. URL https://www.sciencedirect.com/science/article/pii/S0001457521002980.

[32] Xuesong Wang, Minming Yang, and David Hurwitz. Analysis of cut-in behavior based on naturalistic driving data. *Accident Analysis & Prevention*, 124:127–137, 2019. ISSN 0001-4575. doi: https://doi.org/10.1016/j.aap.2019.01.006. URL https://www.sciencedirect.com/science/article/pii/S0001457519300296.

[33] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566, 2023. doi: 10.1109/ICRA48891.2023.10161463.

[34] Deepak Kumar Shrestha. *Modeling and empirical analysis of tailgating behavior of drivers*. Proquest, Umi Dissertation Publishing, September 2011.

[35] Anne E. Dickerson, Timothy A. Reistetter, Stacey Burhans, and Katie Apple. Typical brake reaction times across the life span. *Occupational Therapy In Health Care*, 30 (2):115–123, 2015. doi: 10.3109/07380577.2015.1059971.

[36] 2023. URL https://ev-database.org/cheatsheet/acceleration-electric-car.

APPENDICES

Appendix A

Different Tailgating Coefficient Values for Tailgating Detection

Table A.1: Standard Coefficient Values for Tailgating Detection.

Test	Response-Time	Maximum Front Vehicle Braking Acceleration	Maximum Rear Vehicle Acceleration	Minimum Rear Vehicle Braking Acceleration
Default	2.3s [26]	4.6 m/s^2 [27]	2.0 m/s^2 [25]	3.9 m/s^2 [25]
High-Response	3.0s [29]	-	-	-
High-Maximum-Front-Vehicle-Braking-Acceleration	-	6.1 m/s^2 [28]	-	-
High-Maximum-Rear-Vehicle-Acceleration	-	-	5.4 m/s^2 [36]	-
High-Minimum-Rear-Vehicle-Braking-Acceleration	-	-	-	4.9 m/s^2 [25]
Low-Response	0.5s [35]	-	-	-
Low-Maximum-Front-Vehicle-Braking-Acceleration	-	3.0 m/s^2 [27, 21]	-	-
Low-Maximum-Rear-Vehicle-Acceleration	-	-	1.0 m/s^2 [25]	-
Low-Minimum-Rear-Vehicle-Braking-Acceleration	-	-	-	2.9 m/s^2 [25]

Table A.2: Example Test Case Coefficient Values for Tailgating Position Illustration.

Test	Response-Time	Maximum Front Vehicle Breaking Acceleration	Maximum Rear Vehicle Acceleration	Minimum Rear Vehicle Breaking Acceleration
Default	0.75s	7.85 m/s^2	9.81 m/s^2	4.61 m/s^2
High-Response	1.0s	-	-	-
High-Maximum-Front-Vehicle-Breaking-Acceleration	-	9.62 m/s^2	-	-
High-Maximum-Rear-Vehicle-Acceleration	-	-	10.81 m/s^2	-
High-Minimum-Rear-Vehicle-Breaking-Acceleration	-	-	-	6.08 m/s^2
Low-Response	0.5s	-	-	-
Low-Maximum-Front-Vehicle-Breaking-Acceleration	-	6.08 m/s^2	-	-
Low-Maximum-Rear-Vehicle-Acceleration	-	-	8.81 m/s^2	-
Low-Minimum-Rear-Vehicle-Breaking-Acceleration	-	-	-	3.14 m/s^2

Appendix B

Run-time Results

Please note that in the results, speeding violations are assessed with a speed limit exceedance threshold of zero. Offroad violations are evaluated using half the length and width from the vehicle's centerpoint, and tailgating violations are examined solely for vehicle-following behaviour (using Intel's RSS.I formulation), with no consideration for stopped vehicles.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	128	6	113	114	0	0	0	0	0	73	12	49
vehicle_tracks_001.csv	102	5	90	91	0	0	0	0	0	60	11	36
vehicle_tracks_002.csv	122	6	104	106	0	0	0	0	0	83	2	39
vehicle_tracks_003.csv	89	6	73	76	0	0	0	0	0	59	3	25
vehicle_tracks_004.csv	22	5	7	8	0	0	0	0	0	11	0	6
vehicle_tracks_005.csv	116	2	105	106	0	0	0	0	0	72	10	31
vehicle_tracks_006.csv	129	7	110	117	0	0	0	0	0	97	3	36
vehicle_tracks_007.csv	103	2	93	95	0	0	0	0	0	58	5	32
vehicle_tracks_008.csv	102	4	93	93	0	0	0	0	0	61	8	28
vehicle_tracks_009.csv	32	4	26	26	0	0	0	0	0	18	3	7
All Files	945	47	814	832	0	0	0	0	0	592	57	289

Table B.1: Test Results for resultsLogRoundaboutFourWay.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	36	5	22	22	0	0	0	0	0	9	19	15
vehicle_tracks_001.csv	50	12	28	28	0	0	0	0	0	14	26	19
vehicle_tracks_002.csv	42	9	22	22	0	0	0	0	0	12	17	15
vehicle_tracks_003.csv	59	9	31	31	0	0	0	0	0	28	20	23
vehicle_tracks_004.csv	36	5	17	17	0	0	0	0	0	9	19	17
All Files	223	40	120	120	0	0	0	0	0	72	101	89

Table B.2: Test Results for resultsLogRoundaboutFiveWay.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	74	5	43	63	5	0	0	9	0	23	53	3
vehicle_tracks_001.csv	85	10	42	72	2	0	2	11	3	43	57	3
vehicle_tracks_002.csv	92	4	55	86	3	0	0	9	0	30	73	6
vehicle_tracks_003.csv	87	5	49	79	2	0	2	7	4	36	49	9
vehicle_tracks_004.csv	101	10	57	92	8	0	2	14	3	44	55	8
vehicle_tracks_005.csv	115	16	38	92	10	0	3	23	5	70	52	12
vehicle_tracks_006.csv	105	12	55	94	4	0	1	13	1	29	56	9
vehicle_tracks_007.csv	73	11	42	59	2	0	0	3	0	22	45	7
All Files	732	73	381	637	36	0	10	89	16	297	440	57

Table B.3: Test Results for resultsLogThreeWayIntersectionShort.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	56	10	22	47	1	0	2	9	3	28	20	11
vehicle_tracks_001.csv	91	4	56	89	3	0	0	5	0	45	45	41
vehicle_tracks_002.csv	86	8	49	80	3	0	1	11	2	33	47	34
vehicle_tracks_003.csv	83	7	56	79	1	0	1	7	1	33	37	23
vehicle_tracks_004.csv	87	3	54	84	3	0	2	9	2	36	52	34
vehicle_tracks_005.csv	103	8	61	92	2	0	2	7	3	48	46	51
vehicle_tracks_006.csv	101	3	70	96	3	0	2	7	2	48	50	49
vehicle_tracks_007.csv	101	5	58	100	5	0	1	12	1	51	47	53
All Files	708	48	426	667	21	0	11	67	14	322	344	296

Table B.4: Test Results for resultsLogThreeWayIntersectionWithRoundabout.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	196	105	39	55	0	0	0	0	0	62	2	8
vehicle_tracks_001.csv	227	109	43	67	0	0	0	0	0	94	0	14
vehicle_tracks_002.csv	52	30	13	16	0	0	0	0	0	15	0	2
vehicle_tracks_003.csv	243	117	37	66	0	0	0	0	0	106	0	14
vehicle_tracks_004.csv	227	99	41	65	0	0	0	0	0	99	0	15
vehicle_tracks_005.csv	221	114	33	58	1	0	0	1	0	88	0	11
vehicle_tracks_006.csv	208	110	32	48	0	0	0	0	0	75	0	15
vehicle_tracks_007.csv	203	95	41	66	0	0	0	0	0	80	0	12
vehicle_tracks_008.csv	197	108	28	41	0	0	0	0	0	65	1	20
vehicle_tracks_009.csv	206	117	33	58	0	0	0	0	0	62	0	11
vehicle_tracks_010.csv	85	48	12	20	0	0	0	0	0	30	0	6
vehicle_tracks_011.csv	205	90	39	55	0	0	0	0	0	90	0	19
vehicle_tracks_012.csv	218	91	39	56	0	0	0	0	0	102	0	16
vehicle_tracks_013.csv	212	107	39	54	0	0	0	0	0	80	0	11
vehicle_tracks_014.csv	112	58	17	29	0	0	0	0	0	37	1	7
vehicle_tracks_015.csv	207	102	31	48	0	0	0	0	0	84	0	15
vehicle_tracks_016.csv	201	113	30	44	0	0	0	0	0	61	0	14
vehicle_tracks_017.csv	132	78	19	32	0	0	0	0	0	34	0	7
vehicle_tracks_018.csv	188	110	25	39	0	0	0	0	0	53	0	11
vehicle_tracks_019.csv	207	112	30	52	0	0	0	0	0	67	0	14
vehicle_tracks_020.csv	200	103	43	59	1	0	0	1	0	63	0	9
vehicle_tracks_021.csv	180	91	34	47	0	0	0	0	0	60	0	12
vehicle_tracks_022.csv	107	58	27	33	0	0	0	0	0	29	0	2
vehicle_tracks_023.csv	163	87	33	51	0	0	0	0	0	57	1	7
vehicle_tracks_024.csv	218	117	20	35	0	0	0	0	0	79	0	17
vehicle_tracks_025.csv	205	83	20	34	0	0	0	0	0	111	0	11
vehicle_tracks_026.csv	202	85	37	57	0	0	0	0	0	97	0	12
vehicle_tracks_027.csv	174	88	42	53	0	0	0	0	0	52	0	8
vehicle_tracks_028.csv	194	113	31	50	0	0	0	0	0	52	1	15
vehicle_tracks_029.csv	175	109	25	35	0	0	0	0	0	40	0	8
vehicle_tracks_030.csv	196	101	28	40	0	0	0	0	0	77	0	12
vehicle_tracks_031.csv	63	33	19	27	0	0	0	0	0	14	0	2
vehicle_tracks_032.csv	172	96	30	44	0	0	0	0	0	48	1	12
vehicle_tracks_033.csv	203	103	39	62	0	0	0	0	0	79	0	12
vehicle_tracks_034.csv	195	95	49	66	0	0	0	0	0	65	0	12
vehicle_tracks_035.csv	202	101	45	69	0	0	0	0	0	75	0	12
vehicle_tracks_036.csv	206	89	66	83	0	0	0	0	0	80	0	15
vehicle_tracks_037.csv	148	65	36	57	0	0	0	0	0	73	0	2
vehicle_tracks_038.csv	225	116	30	49	0	0	0	0	0	88	0	10
vehicle_tracks_039.csv	77	37	11	19	0	0	0	0	0	30	1	7
vehicle_tracks_040.csv	212	108	47	63	0	0	0	0	0	77	0	7
vehicle_tracks_041.csv	209	107	33	52	0	0	0	0	0	83	0	10
vehicle_tracks_042.csv	168	75	29	47	0	0	0	0	0	77	0	6
vehicle_tracks_043.csv	201	95	52	77	0	0	0	0	0	91	0	5
vehicle_tracks_044.csv	74	25	30	36	0	0	0	0	0	36	0	6
vehicle_tracks_045.csv	249	86	41	64	0	0	0	0	0	136	0	19
vehicle_tracks_046.csv	163	105	21	39	0	0	0	0	0	44	0	5
vehicle_tracks_047.csv	207	94	42	59	0	0	0	0	0	89	0	8
vehicle_tracks_048.csv	218	114	35	60	0	0	0	0	0	86	0	8
vehicle_tracks_049.csv	102	47	19	27	0	0	0	0	0	42	0	6
vehicle_tracks_050.csv	212	93	46	68	0	0	0	0	0	93	0	17
vehicle_tracks_051.csv	60	31	12	20	0	0	0	0	0	27	0	1
vehicle_tracks_052.csv	187	118	24	42	1	0	0	1	0	52	0	7
vehicle_tracks_053.csv	175	93	41	53	0	0	0	0	0	45	1	10
vehicle_tracks_054.csv	62	37	6	12	0	0	0	0	0	19	0	6
vehicle_tracks_055.csv	174	89	31	45	0	0	0	0	0	69	0	9
vehicle_tracks_056.csv	167	81	35	40	0	0	0	0	0	67	0	8
vehicle_tracks_057.csv	146	92	25	32	0	0	0	0	0	32	0	7
vehicle_tracks_058.csv	171	103	22	32	0	0	0	0	0	52	0	9
vehicle_tracks_059.csv	101	61	17	20	0	0	0	0	0	22	0	5
All Files	10510	5337	1894	2827	3	0	0	3	0	3892	9	598

Table B.5: Test Results for FourWayDiamond.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	151	12	129	144	0	0	0	0	0	78	6	56
vehicle_tracks_001.csv	134	12	116	124	0	0	0	0	0	37	7	46
vehicle_tracks_002.csv	161	12	134	148	0	0	0	0	0	72	7	61
vehicle_tracks_003.csv	136	9	119	129	0	0	0	0	0	66	6	39
vehicle_tracks_004.csv	167	13	133	160	0	0	0	0	0	87	7	52
vehicle_tracks_005.csv	170	12	146	163	0	0	0	0	0	80	8	47
vehicle_tracks_006.csv	180	13	150	172	0	0	0	0	0	88	10	46
vehicle_tracks_007.csv	176	16	146	167	0	0	0	0	0	104	12	57
vehicle_tracks_008.csv	197	7	163	186	0	0	0	0	0	116	11	117
vehicle_tracks_009.csv	195	11	162	185	0	0	0	0	0	108	25	62
vehicle_tracks_010.csv	193	18	163	181	0	0	0	0	0	96	12	52
vehicle_tracks_011.csv	170	10	148	165	0	0	0	0	0	91	17	50
vehicle_tracks_012.csv	184	17	155	172	0	0	0	0	0	96	13	54
vehicle_tracks_013.csv	68	4	57	60	0	0	0	0	0	41	3	19
vehicle_tracks_014.csv	194	13	163	188	0	0	0	0	0	104	26	77
vehicle_tracks_015.csv	198	16	158	182	0	0	0	0	0	109	17	68
vehicle_tracks_016.csv	94	11	72	86	0	0	0	0	0	53	10	32
vehicle_tracks_017.csv	205	20	155	194	0	0	0	0	0	123	15	68
vehicle_tracks_018.csv	197	7	161	186	0	0	0	0	0	128	18	66
vehicle_tracks_019.csv	106	13	77	94	0	0	0	0	0	62	11	30
vehicle_tracks_020.csv	163	8	140	154	0	0	0	0	0	88	13	60
vehicle_tracks_021.csv	71	8	59	63	0	0	0	0	0	29	6	23
vehicle_tracks_022.csv	170	6	148	162	0	0	0	0	0	79	13	63
vehicle_tracks_023.csv	196	20	158	179	0	0	0	0	0	104	8	57
vehicle_tracks_024.csv	170	16	138	156	0	0	0	0	0	78	11	46
vehicle_tracks_025.csv	174	16	137	163	0	0	0	0	0	106	22	54
vehicle_tracks_026.csv	199	15	165	190	0	0	0	0	0	116	23	64
vehicle_tracks_027.csv	177	13	154	172	0	0	0	0	0	90	26	62
vehicle_tracks_028.csv	51	8	33	41	0	0	0	0	0	20	3	15
vehicle_tracks_029.csv	193	9	163	183	0	0	0	0	0	112	19	68
vehicle_tracks_030.csv	181	18	141	167	0	0	0	0	0	102	17	52
vehicle_tracks_031.csv	170	9	136	158	0	0	0	0	0	95	20	58
vehicle_tracks_032.csv	179	12	137	170	0	0	0	0	0	100	18	51
vehicle_tracks_033.csv	90	5	69	85	0	0	0	0	0	55	9	35
vehicle_tracks_034.csv	184	9	154	176	0	0	0	0	0	105	28	96
vehicle_tracks_035.csv	200	12	161	184	0	0	0	0	0	118	27	104
vehicle_tracks_036.csv	192	10	153	180	0	0	0	0	0	122	31	100
vehicle_tracks_037.csv	174	6	145	163	0	0	0	0	0	104	18	149
vehicle_tracks_038.csv	190	9	155	182	0	0	0	0	0	121	19	88
vehicle_tracks_039.csv	182	10	145	164	0	0	0	0	0	106	16	82
vehicle_tracks_040.csv	209	21	158	195	0	0	0	0	0	134	18	83
vehicle_tracks_041.csv	211	22	164	191	0	0	0	0	0	133	24	109
vehicle_tracks_042.csv	131	13	94	109	0	0	0	0	0	85	14	41
vehicle_tracks_043.csv	116	10	103	109	0	0	0	0	0	44	20	41
vehicle_tracks_044.csv	140	9	123	131	0	0	0	0	0	71	13	43
vehicle_tracks_045.csv	104	9	82	95	0	0	0	0	0	58	12	36
All Files	7493	549	6122	7008	0	0	0	0	0	4114	689	2779

Table B.6: Test Results for resultsLogRoundaboutSixWay.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	99	17	72	72	0	0	0	0	0	12	0	22
vehicle_tracks_001.csv	123	19	90	91	0	0	0	0	0	32	0	18
vehicle_tracks_002.csv	107	18	83	87	0	0	0	0	0	13	0	9
vehicle_tracks_003.csv	114	28	86	86	0	0	0	0	0	17	0	25
vehicle_tracks_004.csv	76	17	56	57	0	0	0	0	0	15	0	20
vehicle_tracks_005.csv	94	18	72	73	0	0	0	0	0	11	0	22
vehicle_tracks_006.csv	110	16	91	93	0	0	0	0	0	21	0	15
vehicle_tracks_007.csv	37	4	32	32	0	0	0	0	0	4	0	7
vehicle_tracks_008.csv	78	15	61	62	0	0	0	0	0	11	0	23
vehicle_tracks_009.csv	82	7	67	68	0	0	0	0	0	28	0	15
vehicle_tracks_010.csv	95	18	71	73	0	0	0	0	0	33	0	11
vehicle_tracks_011.csv	62	10	50	51	0	0	0	0	0	20	0	8
All Files	1077	187	831	845	0	0	0	0	0	217	0	195

Table B.7: Test Results for resultsLogRoundaboutThreeWay.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	156	40	43	145	3	0	4	15	4	102	1	0
vehicle_tracks_001.csv	184	29	38	177	9	0	2	46	5	140	0	9
vehicle_tracks_002.csv	163	33	37	157	4	0	0	30	2	121	0	5
vehicle_tracks_003.csv	147	36	45	142	8	0	0	33	6	86	0	5
vehicle_tracks_004.csv	100	33	47	98	4	0	1	8	4	38	0	1
vehicle_tracks_005.csv	139	27	62	134	3	0	1	12	2	85	2	0
vehicle_tracks_006.csv	115	27	55	113	0	0	1	8	1	58	0	5
vehicle_tracks_007.csv	130	28	71	125	0	0	1	5	1	65	0	3
vehicle_tracks_008.csv	142	48	39	136	2	0	2	14	2	73	0	3
vehicle_tracks_009.csv	141	42	41	136	1	0	2	21	5	80	0	3
vehicle_tracks_010.csv	93	19	48	88	3	0	0	6	0	49	0	1
vehicle_tracks_011.csv	135	36	46	133	1	0	1	8	3	81	0	2
vehicle_tracks_012.csv	142	43	31	140	5	0	1	26	5	73	0	6
vehicle_tracks_013.csv	142	27	58	141	5	0	2	16	3	84	0	1
vehicle_tracks_014.csv	148	36	45	143	5	0	2	28	2	95	0	4
vehicle_tracks_015.csv	157	45	50	151	1	0	4	16	5	91	0	3
vehicle_tracks_016.csv	122	27	52	118	1	0	0	17	1	63	0	6
vehicle_tracks_017.csv	106	18	66	104	1	0	1	11	1	46	0	4
vehicle_tracks_018.csv	138	40	47	131	3	0	1	15	3	71	0	8
vehicle_tracks_019.csv	133	31	68	131	0	0	1	9	2	67	1	4
vehicle_tracks_020.csv	139	35	40	136	3	0	3	12	4	82	0	7
vehicle_tracks_021.csv	110	27	63	108	0	0	1	6	3	56	0	2
All Files	2982	727	1092	2887	62	0	31	362	64	1706	4	82

Table B.8: Test Results for resultsLogRulesFourWay.

File	Total Vehicles	Non-Rule-Violating Vehicles	Stop-Violators	Stops Needed	Left Before Right Violators	Lanelet Priority Violators	Should Turn Left Violators	Yields One Needed	Yields Two Needed	Vehicle Tailgating Violations	Speed Limit	No Offroad
vehicle_tracks_000.csv	123	8	58	98	5	0	0	14	2	68	74	15
vehicle_tracks_001.csv	113	10	51	92	2	0	2	11	4	53	69	14
vehicle_tracks_002.csv	100	2	55	84	7	0	1	12	3	45	59	15
vehicle_tracks_003.csv	80	7	40	61	2	0	0	5	0	34	47	6
vehicle_tracks_004.csv	127	8	53	106	7	0	2	23	5	75	67	23
vehicle_tracks_005.csv	114	15	40	81	10	0	3	23	3	66	60	13
All Files	657	50	297	522	33	0	8	88	17	341	376	86

Table B.9: Test Results for resultsLogThreeWayIntersection.

Glossary

all way stop regulatory element All-way-stop regulatory elements regulate uS-SSRI by order of arrival, with all lanelets in the intersection being potentially lanelets that must yield the right-of-way [1]. All-way stop regulatory elements signal intersections where all junctions are stop sign regulated, with yield lanelets representing lanelets that have to yield [1]. A ref_line element directs where vehicles need to yield to stop, and a refers element refers to the traffic signs regulating the rule [1]. 6, 70

Intel's RSS.I formulation A formulation used to determine the d_{min} between two vehicles driving along a road. The formula is modeled using $d_{min}= [v_r p + \frac{1}{2} a_{max,accel} p^2 + \frac{v_r + p a_{max,accel}}{2 a_{min,brake}} - \frac{v_f^2}{2 a_{max,brake}}]_x \; with [x]_x := max\{x, 0\}$ [3]. The speed of the front (v_f) and rear (v_r) vehicles are taken into consideration with their reaction times (p) when determining d_{min}. In addition, the maximum braking acceleration for the front vehicle ($a_{max,brake}$), the maximum acceleration for the rear vehicle ($a_{max,accel}$), and the minimum breaking acceleration for the rear vehicle ($a_{min,brake}$) are used when calculating d_{min} between two vehicles. 57, 58, 60, 61, 79, 81, 96, 116

lanelet Lanelets define atomic sections of the map in which directed motion occurs and are defined as a one-line string on the left and exactly one on the right border [24]. viii, ix, 3–10, 12–15, 26, 31, 34, 35, 38, 39, 42, 44–47, 54, 55, 63, 64, 68–70, 79, 80

Lanelet2 Lanelet2 uses a predefined framework with map representations and a C++ library and acts as a HD Map representation which provides accurate information about a vehicle's surroundings. 2–9, 14–16, 54, 62, 64, 107

point Points represent vertical structures and are an essential element of the map, and only elements with position information and other elements directly or indirectly compose points [24]. Vehicle position can be represented as moving points along a lanelet map, with the vehicle's centerpoint being used to represent its position. ix, 4, 5, 50, 51, 54, 55, 57, 62, 64, 67, 79

regulatory element Regulatory elements express traffic rules applied to the lanelet [24]. Regulatory elements define traffic rules, including speed limits, priority rules, or traffic lights and are referenced by one or more lanelets or areas for which they apply [24]. Due to different types of traffic rules, the exact structure of regulatory elements can be very different [24]. 4–6, 70

right of way regulatory element When stop-signs are present along a traffic junction, right-of-way regulatory elements signal uS-SSRI [1]. Stop sign regulatory elements divide lanelets into yielding and right-of-way lanelets [1]. The yielding lanelets give their right-of-way to the right-of-way lanelets since they have priority over yielding lanelets [1]. A ref_line element defines the lines where vehicles crossing a yield lanelet must stop [1]. 6, 70