FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGY CENTER
AUTOMATION AND SYSTEMS DEPARTMENT
UNDERGRADUATE COURSE IN CONTROL AND AUTOMATION ENGINEERING

Marcelo Vogt

Development of a road events simulator

Toulouse
2024

Marcelo Vogt

Development of a road events simulator

Final report of the subject DAS5511 (Course Final Project) as a Concluding Dissertation of the Undergraduate Course in Control and Automation Engineering of the Federal University of Santa Catarina.
Supervisor: Prof. Jérôme Ermont, Dr.
Co-supervisor: Alexis Rodet, Eng.

Toulouse
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Vogt, Marcelo
Desenvolvimento de um simulador de eventos urbanos / Marcelo Vogt ; orientador, Jérôme Ermont, coorientador, Alexis Rodet, 2024.
43 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2024.

Inclui referências.

1. Engenharia de Controle e Automação. 2. automotivo. 3. ADAS. 4. simulador. I. Ermont, Jérôme. II. Rodet, Alexis. III. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. IV. Título.

Marcelo Vogt

Development of a road events simulator

This dissertation was evaluated in the context of the subject DAS5511 (Course Final Project) and approved in its final form by the Undergraduate Course in Control and Automation Engineering

Toulouse, September 09th, 2024.

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Course Coordinator

Examining Board:

Alexis Rodet, Eng.
Supervisor
Bertrandt Company

Prof. Jérôme Ermont, Dr.
ENSEEIHT/SN/E

Prof. Katia Jaffrès-Runser, Dr.
IRIT

Prof. Martial Coulon, Dr.
ENSEEIHT/SN

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to the individuals who supported me throughout this project. First and foremost, I want to thank my internship supervisor, Alexis Rodet, for the opportunity to work on such a meaningful project and for providing guidance along the way. I also extend my appreciation to Jérôme Ermônt, my academic supervisor, for his formal support during this internship.

A special thanks to Abdelaziz Jelassi, my colleague at Bertrandt, whose professional support and insight were invaluable throughout my time here. His readiness to help and constant availability ensured that I had the assistance I needed to progress.

I would also like to thank the entire Bertrandt team for making this internship both professionally enriching and personally enjoyable. From playing sports at the Jeux de Sophia, to our regular table tennis matches during breaks, and the Friday lunches at Naga, these experiences made my time at Bertrandt unforgettable.

On a personal note, I am deeply grateful to my parents, whose unwavering support and guidance have shaped my journey from the beginning. Finally, I want to thank my sister for always being there, offering me encouragement and emotional support from afar, which has been a great comfort during this time.

DISCLAIMER

Toulouse, September 9-th, 2024.

As representative of the Bertrandt S.A.S. in which the present work was carried out, I declare this document to be exempt from any confidential or sensitive content regarding intellectual property, that may keep it from being published by the Federal University of Santa Catarina (UFSC) to the general public, including its online availability in the Institutional Repository of the University Library (BU). Furthermore, I attest knowledge of the obligation by the author, as a student of UFSC, to deposit this document in the said Institutional Repository, for being it a Final Program Dissertation (*"Trabalho de Conclusão de Curso"*), in accordance with the *Resolução Normativa n°126/2019/CUn*.

Alexis Rodet
Bertrandt S.A.S.

ABSTRACT

The development of Advanced Driver Assistance Systems (ADAS) requires robust testing methodologies to ensure their safety and reliability across diverse driving conditions. Traditional real-world testing methods face significant challenges, including high costs, logistical complexity, and safety risks, making it difficult to consistently reproduce and assess specific scenarios.

This project addresses these challenges by developing a specialized road events simulator designed for ADAS testing. The simulator offers a controlled, repeatable virtual environment capable of generating a variety of road scenarios using the CARLA simulator and OpenDRIVE format. This approach allows for the efficient and detailed evaluation of ADAS functionalities in dynamic conditions, overcoming the limitations of real-world testing.

Key benefits of the simulator include its capability to execute simulations faster than real-time and produce outputs optimized for analysis. It is adaptable across different ADAS projects, contributing to enhanced development processes and supporting the advancement of vehicle safety and autonomous driving technologies. Ultimately, this tool aims to reduce the time and cost of ADAS validation, facilitating progress toward safer and more intelligent vehicles.

Keywords: Automotive. ADAS. Simulator.

RESUMO

O desenvolvimento de Sistemas Avançados de Assistência ao Motorista (ADAS) exige metodologias de teste robustas para garantir sua segurança e confiabilidade em diversas condições de direção. Os métodos tradicionais métodos tradicionais de teste no mundo real enfrentam desafios significativos, incluindo altos custos, complexidade logística e riscos à segurança, o que dificulta a reproduzir e avaliar cenários específicos de forma consistente.

Este projeto aborda esses desafios ao desenvolver um simulador especializado em eventos rodoviários projetado para testes de ADAS. O simulador oferece um ambiente virtual controlado e repetível capaz de gerar uma variedade de cenários rodoviários usando o simulador CARLA e o formato OpenDRIVE. Essa abordagem permite a avaliação eficiente e detalhada das funcionalidades do ADAS em condições dinâmicas, superando as limitações dos testes no mundo real. testes reais.

Os principais benefícios do simulador incluem sua capacidade de executar simulações mais rapidamente do que em tempo real e produzir resultados otimizados para análise. Ele é adaptável em diferentes projetos ADAS, contribuindo para processos de desenvolvimento aprimorados e apoiando o avanço da segurança do veículo e das tecnologias de direção autônoma autônoma. Em última análise, essa ferramenta tem como objetivo reduzir o tempo e o custo da e o custo da validação do ADAS, facilitando o progresso rumo a veículos mais seguros e veículos mais seguros e inteligentes.

Palavras-chave: Automotivo. ADAS. Simulador.

RÉSUMÉ

Le développement de systèmes avancés d'aide à la conduite (ADAS) nécessite des méthodologies d'essai robustes pour garantir leur sécurité et leur fiabilité dans diverses conditions de conduite. sécurité et leur fiabilité dans diverses conditions de conduite. Les méthodes traditionnelles Les méthodes traditionnelles d'essai en conditions réelles sont confrontées à des défis importants, notamment des coûts élevés, une complexité logistique et des risques pour la sécurité. coûts élevés, la complexité logistique et les risques pour la sécurité, ce qui rend difficile la reproduction et l'évaluation de scénarios spécifiques. Il est donc difficile de reproduire et d'évaluer des scénarios spécifiques de manière cohérente.

Ce projet relève ces défis en développant un simulateur d'événements routiers spécialisé conçu pour les essais ADAS. Le simulateur offre un environnement virtuel contrôlé et reproductible capable de générer une variété de scénarios routiers en utilisant le simulateur CARLA et le format OpenDRIVE. Cette approche permet l'évaluation efficace et détaillée des fonctionnalités ADAS dans des conditions dynamiques, en surmontant les limites des essais en du monde réel.

Les principaux avantages du simulateur sont sa capacité à exécuter des simulations simulations plus rapidement qu'en temps réel et de produire des résultats optimisés pour l'analyse. Il est adaptable à différents projets ADAS, contribuant à l'amélioration des processus de développement et soutenant l'avancement de la sécurité des véhicules et des technologies de conduite autonomes. En fin de compte, cet outil vise à réduire le temps et le coût de la validation des systèmes d'aide à la conduite. et le coût de la validation des systèmes d'aide à la conduite, facilitant ainsi les progrès vers des véhicules plus plus sûrs et plus intelligents.

Mots-clés : Automobile. ADAS. Simulateur.

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

Acc	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
ADASIS	ADAS Interface Specification
ADASISv2	ADASIS Version 2
CAN	Controller Area Network
CC	Cruise Control
CI/CD	Continuous Integration and Continuous Delivery
EHP	Electronic Horizon Provider
RHT	Right-Hand Traffic
SAS	Speed Adaptation System

CONTENTS

1 INTRODUCTION

ADAS represent a significant leap forward in automotive technology, offering a range of features designed to enhance vehicle safety and improve the driving experience. These systems, which include functionalities such as lane departure warnings, adaptive cruise control, and automated emergency braking, are integral to the progression towards fully autonomous vehicles. As vehicles become increasingly connected and intelligent, the role of ADAS in reducing traffic accidents and supporting drivers under various conditions becomes ever more critical. There are multiple levels of automation in ADAS, as illustrated by Figure 1, starting from level 0, without any automation, all the way to level 5 with a fully autonomous vehicle.

Testing ADAS features in real-world settings presents substantial challenges. The unpredictability of live road conditions makes it difficult to reproduce specific scenarios consistently, which is crucial for thoroughly validating system responses. Additionally, real-world testing carries inherent safety risks, especially when assessing systems designed to intervene in dangerous situations. Moreover, the logistical and financial burden of conducting extensive on-road tests across different environments and conditions can be prohibitive. These challenges underscore the need for reliable, controlled, and repeatable testing environments where ADAS functionalities can be evaluated rigorously before they are deployed in real-world vehicles.

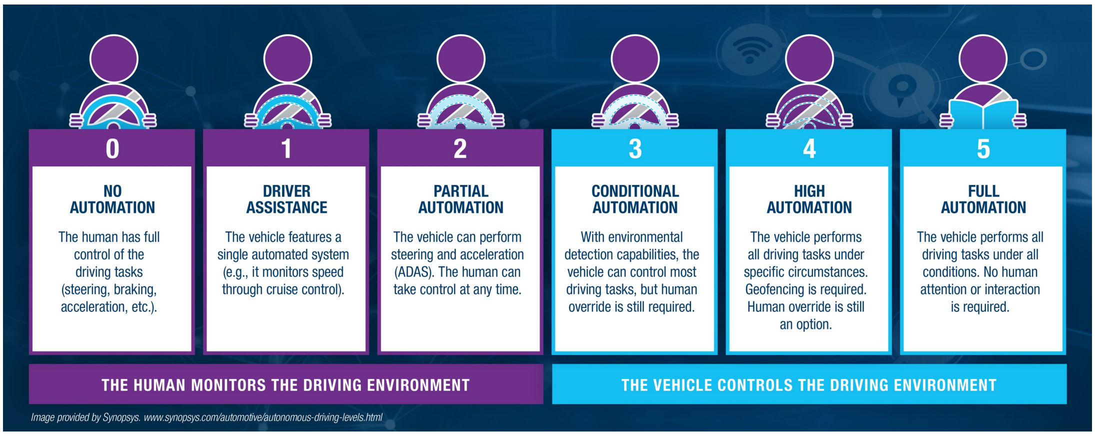

Figure 1 – Levels of Driving Automation in ADAS (EFTHYVOULIDIS, 2021)

In response to these challenges, this work explores the development of a road events simulator concept tailored for testing and validating ADAS systems. The simulator framework aims to create a versatile tool capable of replicating a wide range of road scenarios in a controlled virtual environment. By enabling real-time execution of these scenarios, such a simulator allows developers to systematically assess ADAS features' performance under diverse conditions without the constraints and risks associated with real-world testing. The overarching objective is to provide a comprehensive testing solution that supports the advancement of safer, more reliable ADAS technologies.

The scope of this work extends beyond discussing the simulator's conceptual framework. It emphasizes key principles for efficiency, flexibility, and the generation of outputs that are easy for developers to analyse and suitable for integration with other systems. The adaptability of such a simulator is crucial for accommodating the wide range of behaviours and responses that different ADAS features must handle. The work also discusses establishing a robust testing and validation framework to ensure that outputs are accurate, consistent, and aligned with intended testing parameters.

In summary, this work addresses the critical need for reliable and efficient tools in the testing and validation of ADAS systems. By examining the role of simulators in providing controlled environments for testing, it contributes to the broader goal of advancing ADAS technology, enhancing vehicle safety, and paving the way for more autonomous driving capabilities.

2 BERTRANDT GROUP

The Bertrandt Group is a global engineering and technology company founded in 1974 in Germany. Initially focused on automotive development, Bertrandt has expanded its expertise across various industries, including aerospace, agriculture, medical technology, mechanical engineering, electronics, and energy. The company operates in multiple countries, maintaining a strong presence in Europe, the United States, China, and Morocco.

Figure 2 – Bertrandt Group logo

Bertrandt is recognized for its innovative solutions tailored to meet the demands of emerging technologies. The company's focus areas include digitalization, connectivity, autonomous systems, and sustainable energy development.

2.1 KEY PROJECTS

Bertrandt has contributed to a wide range of innovative projects across various industries. The company's capabilities include developing future mobility solutions, implementing advanced driver assistance systems (ADAS), and integrating sustainable technologies.

Bertrandt has supported several significant automotive and engineering projects, providing expertise in areas such as:

- Vehicle architecture and design
- Electrical and electronic systems
- Interior and exterior equipment
- Fuel and cooling systems
- Quality assurance and validation

These contributions highlight Bertrandt's role in advancing technological solutions for various industries.

2.2 BERTRANDT FRANCE

Bertrandt France operates across multiple locations, including sites in Valbonne, Montbéliard, Toulouse, and Vélizy-Villacoublay. The teams focus on areas such as awareness, fusion, quality, validation, and big data. The Valbonne site, established in early 2022, serves as a hub for project management and human resources, supporting Bertrandt's mission to deliver innovative solutions.

I completed my internship at the Bertrandt Sophia-Antipolis site in Valbonne, contributing to projects related to automotive systems and validation processes.

3 DEVELOPMENT

3.1 CONTEXT

This section presents and explains the key systems, components, and tools necessary for understanding the developed project.

- **Advanced Driver Assistance System**

  ADAS encompasses a suite of systems designed to aid drivers by enhancing their control over the vehicle and reducing the risk of accidents. These systems include features like lane departure warnings, automatic emergency braking, and adaptive cruise control, all of which work together to make driving safer and more efficient.

  Furthermore, ADAS systems offer a wide array of advantages that extend beyond the scope of safety improvements. These systems enhance the overall driving experience through the automation of routine tasks, such as adaptive cruise control, while also providing advanced analytical insights into driving behaviour. Moreover, ADAS contributes significantly to vehicle maintenance optimization by enabling predictive maintenance strategies, thereby minimizing downtime and associated costs. By analysing and interpreting driving patterns, these systems can anticipate potential issues, allowing for proactive interventions that ensure optimal vehicle performance and reliability.

- **Electronic Horizon Provider (EHP)**

  The EHP is typically an embedded system within the vehicle that supplies essential map and navigation data to various other vehicle components. By providing real-time data about the vehicle's surroundings, the EHP enables other systems to make informed decisions regarding navigation and safety.

- **Speed Adaptation System (SAS)**

  The SAS leverages the *horizon* data to adjust the vehicle's speed appropriately. By anticipating changes in the road and traffic conditions, the SAS ensures the vehicle maintains a safe and efficient speed, enhancing overall driving comfort and safety.

- **Adaptive Cruise Control (Acc)**

  Adaptive Cruise Control (Acc) is an advanced variant of traditional Cruise Control (CC). Unlike CC, which maintains a constant speed set by the driver, Acc dynamically adjusts the vehicle's speed using SAS, based on the surrounding traffic and road conditions. By continuously monitoring the distance to the vehicle ahead and other contextual factors, Acc ensures a smoother and safer driving experience.

- **ADASIS Version 2 (ADASISv2)**

  ADASISv2 (ADASIS, 2010) is the second version of ADAS Interface Specification (ADASIS), a protocol designed to interface between ADAS and EHP systems. Navigation data is often inaccessible to components outside the EHP, and since each EHP can differ significantly, the data representation and organization may vary greatly. This necessitated a standardized protocol to ensure consistent communication across different systems.

- **CARLA**

  CARLA (DOSOVITSKIY et al., 2017) is an open-source urban driving simulator developed using the Unreal Engine in `C++`. It is widely used in the autonomous driving domain due to its highly realistic 3D environments. CARLA provides API in both `C++` and `Python`, allowing for seamless connection and control of the simulator.

- **OpenDRIVE**

  OpenDRIVE (OPENDRIVE. . . , 2023) is an open-source format for describing road networks. It is used by CARLA to define the structure of the environment, including roads, lanes, intersections, and other relevant features necessary for realistic simulations of driving scenarios. The associated file extension is `xodr`.

3.2 MOTIVATION

Testing ADAS in the real world is not only time-consuming but also potentially hazardous. It involves driving while paying attention to potentially incorrect advice or even allowing a potentially unsafe autonomous system to take control. Additionally, testing specific scenarios outside a simulation necessitates finding those scenarios in the real world, which can be impractical and geographically restrictive. In contrast, a simulation environment allows for the easy creation of diverse scenarios, facilitating thorough testing without the associated risks and logistical challenges.

A simulation that supports the testing of arbitrary scenarios enables much safer and more efficient testing. Such simulations provide a controlled environment where rare and hazardous events can be reproduced consistently, ensuring the robustness and reliability of ADAS systems. This controlled setting allows for detailed analysis, as simulations can be paused, replayed, and scrutinized in ways that real-world testing cannot match. Furthermore, running simulations in accelerated time dramatically speeds up the testing process, allowing developers to simulate hours or days of driving in a much shorter period. This not only reduces costs by eliminating the need for physical test vehicles, safety drivers, and extensive logistical arrangements but also accelerates the development cycle by enabling rapid prototyping and immediate feedback. Overall, simulation-based testing, especially with the capability to run in accelerated time, is essential for developing and validating ADAS systems to the highest safety standards.

3.3 OBJECTIVE

This project's main objective is to simulate an EHP using CARLA. CARLA uses the OpenDRIVE format to store map data, and its API provide high-level access to it. Since an EHP's output should be encoded using ADASISv2, a tool that performs the conversion from OpenDRIVE to ADASISv2 was necessitated.

CARLA's high-level map data comprises `Waypoints`, which are directed points (often referred to as rays) associated with road-specific information. `Waypoints` enable querying adjacent lanes and retrieving other `Waypoints` at specified distances ahead or behind. This querying can yield multiple results when there are multiple potential paths. `Waypoints` are procedurally generated along roads at arbitrary positions, meaning there are no preset positions on the road as the name may imply.

The ADASISv2 protocol is composed of seven message types, that are to be procedurally generated by the simulation with data extracted from CARLA. The seven message types that compose the ADASISv2 protocol are:

- '`SYSTEM SPECIFIC`', carrying system-specific messages;
- '`POSITION`', describing the vehicle's position in relation to the *horizon*;
- '`SEGMENT`', specifying attributes of the road ahead;
- '`STUB`', describing the structure, or branching, of the path;
- '`PROFILE SHORT`', carrying data about road shape and objects;
- '`PROFILE LONG`', providing detailed data about road shape and objects on the road;
- '`META`', describing basic system characteristics, such as speed units and driving country.

3.4 STUB

A Stub message carries information regarding how the road is structured. That is, how long each road is, where intersections are, and what roads they connect, together with some structural information.

3.4.1 Content

Every ADASISv2 message must be encoded onto 64 bits to be sent over Controller Area Network (CAN) (ROAD VEHICLES — CONTROLLER AREA NETWORK

	7	6	5	4	3	2	1	0
0	Message Type			Offset				
1	Offset							
2	Cyclic Counter		Path Index					
3	Sub-Path Index						Retrans	Update
4	Relative Probability					Functional Road Class		
5	Part of Calc Route		Cmplx Intersection		Form of Way			
6	Turn Angle							
7	LastStub	Right of Way		Num Lanes Opp		Num Lanes Driving		

Figure 3 – CAN layout of STUB message

(CAN)..., 2024), thus there's a lot of thought put into how many bits are needed for each field. Amongst its various fields shown in figure 3, the most notable:

- **Offset**, providing the Stub's distance to the *horizon*'s origin;
- **Path Index**, describing which Stub this Stub is a child of;
- **Sub Path Index**, specifying the new Stub's path identifier;
- **Turn Angle**, describing the Stub's angle relative to its parent;
- **Number of Lanes**, describing how many lanes there are in either direction.

3.4.2 A Waypoint Visualizer

Initially, the logic behind `Waypoints` was not clearly understood, necessitating a method to visualize them within the simulation. To address this, an OpenGL (SHREINER et al., 2013) overlay was developed. This overlay utilizes the standard image output from the simulation as a background and covers it with differently coloured arrows, each representing a `Waypoint`, as illustrated in figure 4.

Considerable effort was dedicated to optimizing this overlay. Initially, it could render no more than eight arrows without significant performance degradation. However, by learning to utilize buffers, we enabled the overlay to handle hundreds of arrows without any loss in performance, ensuring that arrow rendering was no longer a bottleneck. Another optimization involved efficiently copying CARLA's background texture to OpenGL. Initially, the texture was converted on the CPU to a more manageable format. By interpreting it directly, we achieved a substantial performance improvement.

3.4.3 Generation

Figure 5 illustrates the basic structure of stubs. Each path segment terminates at an intersection, marking the end of one stub and the beginning of its child stubs. Intersections can be categorized into two types: breaking and normal. A breaking intersection, typically seen in T-junctions, occurs when the originating path is the central one,

Figure 4 – `Waypoint` visualization

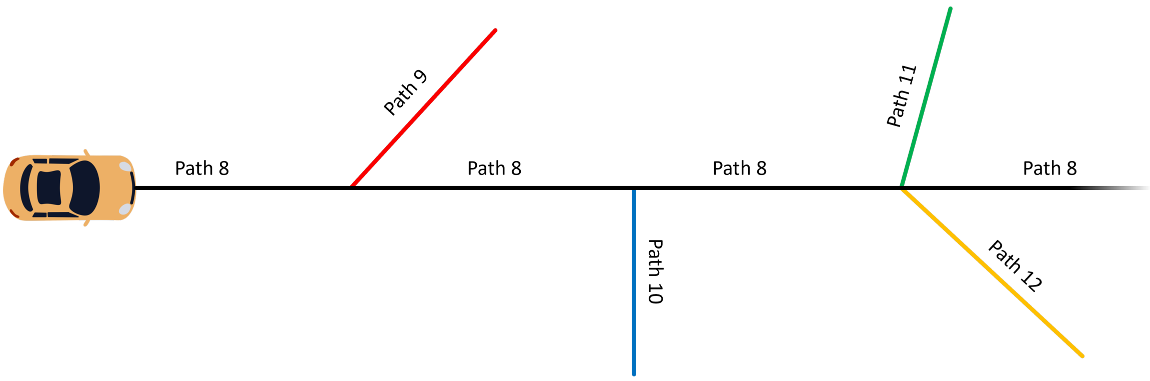

Figure 5 – Stub diagram

and no clear continuation exists. In contrast, a normal intersection has an obvious next path, referred to as the Path Continuation. Accurately identifying these continuations is a key aspect of this project. Another critical concept is the Main Path, which refers to the road the vehicle is currently on, along with all its Path Continuations.

Both concepts play a crucial role in the generation of Stub messages, as there are specific constraints that must be met for the generated path. These constraints ensure that the ADAS has sufficient visibility of the road ahead, enabling it to make informed decisions.

3.5 POSITION

The generation of position messages followed the creation of stubs, as specifying the vehicle's position relies on an understanding of the road structure.

	7	6	5	4	3	2	1	0
0	Message Type			Offset				
1	Offset							
2	Cyclic Counter		Path Index					
3	Index		Probability					Age
4	Age							
5		Confidence			Current Lane			Speed
6	Speed							
7	Relative Heading							

Figure 6 – CAN layout of POSITION message

3.5.1 Content

The purpose of a position message is to describe the vehicle's location relative to the road. Figure 6 shows the complete CAN layout, and amongst these fields, the key ones are:

- **Offset**, providing the vehicle's distance to the *horizon*'s origin;
- **Path Index**, indicating the Stub the vehicle is on;
- **Current Lane**, indicating the lane in which the vehicle is on;
- **Speed**, specifying the vehicle's speed;
- **Relative Heading**, describing the vehicle's angle in relation to the road.

3.5.2 Generation

A major benefit of simulation is the availability of precise information about most objects, including the vehicle. The **Path Index** is easily determined since the vehicle's position is already used for generating the Stubs. Similarly, the vehicle's speed is readily available.

Determining lane information is more complex. To find adjacent lanes, the functions `get_left_lane` and `get_right_lane` can be used. Initially, it might seem practical to repeatedly call `get_right_lane` until no further lanes are found, and similarly for `get_left_lane`. However, this approach fails because the definitions of left and right are relative to the `Waypoint`'s heading. If a lane faces the opposite direction, the algorithm could enter an infinite loop, as depicted in Figure 7.

For **Relative Heading**, the simplest and most effective method was to calculate the difference between the vehicle's yaw and the road's yaw. The only downside to this approach is the loss of angles outside the horizontal plane, but in practice, our interest lies primarily in these horizontal angles.

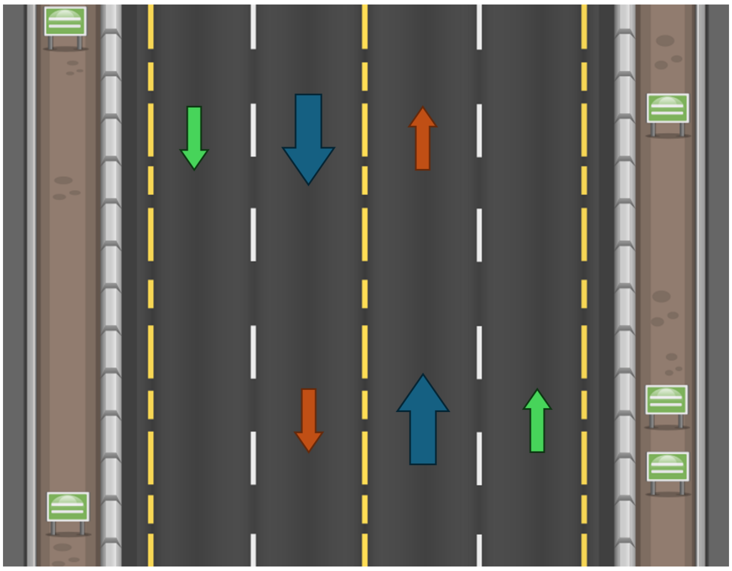

Figure 7 – Waypoint's left (in red) and right (in green).

3.6 SEGMENT

Segment messages carry information about segments of the road and are essential for a complete understanding of the road.

3.6.1 Content

Segments are very tied to the generation of Stubs. Not only do they reference a Stub, it wouldn't be incorrect to state that they are parts of the Stub. Their complete CAN layout can be seen in figure 8, and their key fields are:

- **Offset**, providing the segment's distance to the horizon's origin;
- **Path Index**, indicating the parent Stub;
- **Speed Limit**, specifying the speed limit and speed unit;
- **Number of Lanes**, describing how many lanes there are in either direction;
- **Tunnel and Bridge**, indicating whether this segment is a tunnel, a bridge, both, or neither.

3.6.2 Generation

Unlike Stubs, Segments were implemented incrementally. Initially, each Stub is associated with a single Segment containing temporary data. A Segment's **Offset** value depends on its origin. If it originates from a Stub, it shares the same **Offset** as the Stub. Otherwise, it must fall between the previous Segment's **Offset** and the closest Stub's

	7	6	5	4	3	2	1	0
0	Message Type			Offset				
1	Offset							
2	Cyclic Counter		Path Index					
3	Tunnel		Bridge		Built-up Area		Retrans	Update
4	Relative Probability					Functional Road Class		
5	Part of Calc Route		Cmplx Intersection		Form of Way			
6	Effective Speed Limit					Effective Speed Limit Type		
7		Divided Road		Num Lanes Opp		Num Lanes Driving		

Figure 8 – CAN layout of SEGMENT message

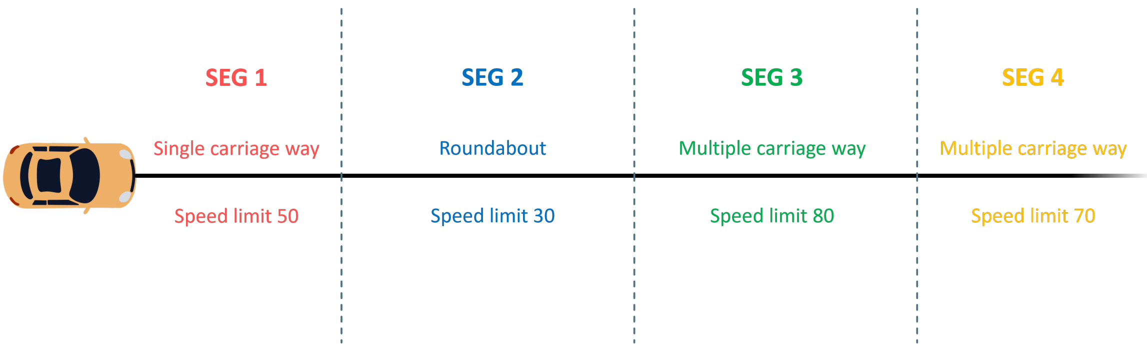

Figure 9 – Segment diagram

Offset. Every Stub must have at least one Segment, with no upper limit. Indicating the parent Stub is straightforward; the Segment's **Path index** is set match the Stub's **Sub Path Index**.

Including the **Number of Lanes** within the Segment might seem redundant when it's also encoded in the Stub. However, this redundancy serves a purpose. It may seem redundant to have the **Number of Lanes** encoded in the Segment again, but it's actually the opposite. Leaving this value in the Segment allows for indicating a change in the number of lanes in the middle of a Stub, while the Stub only encodes this value at the beginning of its road. After all, it is true that there's a bit of redundancy, and we cannot explain its purpose.

Obtaining **Speed Limits** proved more challenging than anticipated. While CARLA internally possesses speed limits for all roads, these are inaccessible though its API. Consequently, we faced a choice: modify the API or manually parse CARLA's map file to extract the information. To maintain compatibility and ease of installation we opted for the latter approach.

As previously mentioned, CARLA uses the OpenDRIVE (OPENDRIVE. . . , 2023), an open format specification that standardizes the logical road description, used for storing map data. After identifying and reading CARLA's OpenDRIVE file, we managed to extract precise information we needed for filling this message's fields.

3.6.3 Tunnels and Bridges

The detection of **Tunnels** and **Bridges** was the most challenging, especially because neither CARLA nor OpenDRIVE have any information in this regard. The solution proposed was to take advantage of CARLA's *Semantic Segmentation* cameras, and a property called *face culling*, represented on figure 10, which allowed us to effectively (at least within our map pool) detect the presence of tunnels and bridges.

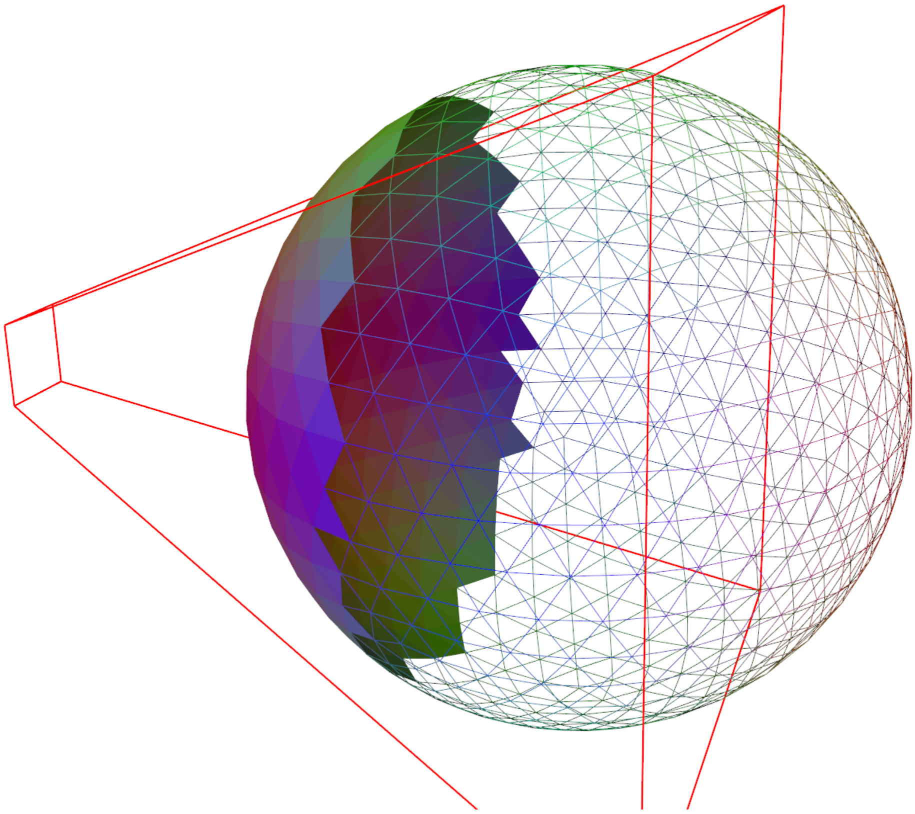

Figure 10 – Back-face culling in sphere model

3.7 META

Meta messages carry meta-information about other messages being sent. They are essential to have a ensure correct interpretation of values.

3.7.1 Content

Meta messages main objective is to describe information about the map and hardware, such as what country the vehicle is in and the speed unit. The complete CAN layout can be seen in figure 11, and their key features are:

- **Country Code**, providing the ISO 3166-1 (ISO CENTRAL SECRETARY, 2006) code for the current country;

	7	6	5	4	3	2	1	0
0	Message Type			Map Provider			Country Code	
1	Country Code							
2	Cyclic Counter		Prot Version Major	Prot Version Minor Sub				Hw Ver
3	Hardware Version							
4	Drv Side	Region Code						
5	Region Code							
6	Map Version Qtr		Map Version Year					
7				Spd Unit	Prot Version Minor			

Figure 11 – CAN layout of META message

- **Region Code**, indicating the ISO 3166-2 (ISO CENTRAL SECRETARY, 2007) code the region inside the current country;
- **Speed Unit**, specifying the speed unit in use;
- **Protocol Version**, describing the version of ADASIS being used to encode the data;
- **Driving Side**, indicating the side of the road one should drive on.

3.7.2 Generation

There's really not much to say here. The **Country Code** is obtained by using Python's `pycountry` library, where one can input a country or region name and easily get the corresponding ISO code. The Speed Unit is configured via a flag at program start, and if we find that we're sending the wrong unit (if a value has the wrong unit it is always converted before being sent) we send a new Meta message changing it. The protocol version will always be the same, in our case we're working with version 2.0.4. The driving side is hardcoded to Right-Hand Traffic (RHT) since for now we don't plan on

3.8 PROFILE SHORT

The Profile Short was left for last, since it describes several different characteristics of the road, such as its slope and curvature.

3.8.1 Content

Profile Short messages contain multiple values that are interpreted according to the specified **Profile Type**. The key possible values for **Profile Type** include:

- **Curvature** specifies the road's curvature at a particular point.
- **Slope** specifies the road's gradient at a certain location.

<table>
<tr><th></th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="3">Message Type</td><td colspan="5">Offset</td></tr>
<tr><td>1</td><td colspan="8">Offset</td></tr>
<tr><td>2</td><td colspan="2">Cyclic Counter</td><td colspan="6">Path Index</td></tr>
<tr><td>3</td><td colspan="5">Profile Type</td><td>CtrlPoint</td><td>Retrans</td><td>Update</td></tr>
<tr><td>4</td><td colspan="2">Accuracy</td><td colspan="6">Distance 1</td></tr>
<tr><td>5</td><td colspan="4">Distance 1</td><td colspan="4">Value 0</td></tr>
<tr><td>6</td><td colspan="6">Value 0</td><td colspan="2">Value 1</td></tr>
<tr><td>7</td><td colspan="8">Value 1</td></tr>
</table>

Figure 12 – CAN layout of PROFILE-SHORT message

- **Road Condition** describes the type of material used for paving the road and evaluates its current condition.
- **Variable Speed Sign Position** indicates the position of a variable speed sign along the road.

The fields **Value 0** and **Value 1** store a value of type **Profile Type**, and respectively correspond to positions **Offset** and **Offset** + **Distance 1**. The full CAN layout is represented by figure 12, and its key fields are:

- **Offset**, representing the position at which the profile starts;
- **Profile Type**, indicating the type of value being described;
- **Value 0**, representing the profile's starting value;
- **Value 1**, indicating the profile's end value;
- **Distance 1**, describing the distance between the profile's start and end;
- **Control Point**, allowing for higher order interpolation.

4 CONCLUSION

This project successfully developed a road events simulator tailored for testing ADAS. The primary aim was to create a versatile and efficient tool capable of simulating a wide range of road scenarios in real-time. The simulator leverages optimized methods to ensure smooth execution, generating two key output formats—verbose and hexadecimal—each serving distinct purposes within the system.

The simulator produces verbose and hexadecimal logs to balance human readability and computational efficiency. The verbose format is designed to include redundant information, allowing easier interpretation and analysis by developers, especially during the debugging process. Conversely, the hexadecimal format is more compact, optimized for processing and storage. This dual-format approach enhances flexibility, enabling deeper insights into both the underlying system and the ADAS performance, while maintaining efficient data management.

To verify the simulator's outputs and guarantee the accuracy of its operations, a comprehensive testing and validation framework was established. The system's core functionalities, particularly those related to message encoding, decoding, and conversion, underwent rigorous unit testing. These tests were crucial in identifying and resolving potential errors at an early stage, ensuring consistent performance. Additionally, type checking was incorporated to enforce data type consistency across the codebase, significantly reducing the likelihood of errors stemming from type mismatches. Output verification was further strengthened through the use of custom-built scripts, namely the Extractor, Validator, and Converter. These tools allowed for efficient data extraction, message verification, and format conversion, ensuring that all outputs were accurate and adhered to the specified requirements.

The implementation of a Continuous Integration and Continuous Delivery (CI/CD) pipeline marked a key advancement in the automation of testing and validation processes. By automating unit tests, type checking, and dead code analysis, the pipeline facilitated consistent code quality across iterations. Moreover, the remote execution feature of the pipeline ensured that code could be tested independently of specific machine configurations, thus verifying its portability. Each simulation run generated fresh outputs that were subsequently validated, reinforcing the system's reliability and robustness over time.

Beyond its technical implementation, this simulator offers significant contributions to the development of ADAS systems. By providing reliable, repeatable, and adaptable road event simulations, the system enables developers to test and validate ADAS features under controlled conditions. The flexibility of the simulator, combined with its efficient output handling, makes it a valuable tool for simulating a variety of road scenarios, ultimately contributing to safer and more advanced driving technologies.

Looking ahead, several opportunities for expanding the simulator's capabilities have been identified. A primary area for improvement involves the incorporation of the missing message types, which include *Profile Long* and some *Profile Short* subtypes. Integrating these would enhance the simulator's accuracy and provide more comprehensive event data for testing purposes.

Additionally, a potential area for future development involves the creation of a scenario reconstruction tool based on hexadecimal output files. Such a tool would be invaluable for debugging ADAS systems, offering a timeline slider to visualize data at specific points during the simulation. This would allow developers to see a clear representation of the road network and event data transmitted up to any given point in time. By providing a visual, intuitive interface, the tool would simplify the debugging process, enabling developers to quickly identify issues in the ADAS' interpretation of outputs and improve overall system performance.

In conclusion, the road events simulator developed in this project stands as a comprehensive tool for ADAS testing and validation. Its efficient message generation, robust validation procedures, and adaptable design make it an essential asset in the advancement of ADAS technologies. Future developments, including the incorporation of additional message types and the potential scenario reconstruction tool, would further enhance its capabilities, ensuring it remains a vital resource for ADAS system developers.

REFERENCES

ADASIS. **ADASIS v2 Specification**. [S.l.], 2010. Available from: https://adasis.org/specification/. Visited on: 29 Sept. 2023.

ASSOCIATION FOR STANDARDIZATION OF AUTOMATION and MEASURING SYSTEMS. **OpenDRIVE**. [S.l.], Nov. 2023. Available from: https://www.asam.net/standards/detail/opendrive/.

DOSOVITSKIY, Alexey; ROS, German; CODEVILLA, Felipe; LOPEZ, Antonio; KOLTUN, Vladlen. CARLA: An Open Urban Driving Simulator. In: LEVINE, Sergey; VANHOUCKE, Vincent; GOLDBERG, Ken (Eds.). **Proceedings of the 1st Annual Conference on Robot Learning**. [S.l.]: PMLR, Nov. 2017. (Proceedings of Machine Learning Research), p. 1–16. Available from: https://proceedings.mlr.press/v78/dosovitskiy17a.html.

EFTHYVOULIDIS, Angelo. **Advanced Driver Assistance Systems (ADAS): The Bridge to Autonomous Vehicles**. [S.l.: s.n.], 2021. https://www.jabil.com/blog/advanced-driver-assistance-systems-adas-the-bridge-to-autonomous-vehicles.html.

ROAD VEHICLES — CONTROLLER AREA NETWORK (CAN). en. [S.l.], May 2024. Available from: https://www.iso.org/standard/86384.html.

ISO CENTRAL SECRETARY. **Codes for the representation of names of countries and their subdivisions - Part 1: Country code**. [S.l.], 2006. Available from: https://www.iso.org/standard/39719.html.

ISO CENTRAL SECRETARY. **Codes for the representation of names of countries and their subdivisions - Part 2: Country subdivision code**. [S.l.], 2007. Available from: https://www.iso.org/standard/39718.html.

SHREINER, Dave; SELLERS, Graham; KESSENICH, John; LICEA-KANE, Bill. **OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3**. 8th. Boston, MA: Addison-Wesley, 2013. ISBN 978-0321773036.