UNIVERSITÀ DEGLI STUDI DI UDINE
HIC SUNT FUTURA

DEPARTMENT OF MATHEMATICS, COMPUTER SCIENCE AND PHYSICS

BACHELOR THESIS IN
INTERNET OF THINGS, BIG DATA & MACHINE LEARNING

Decreasing the reality gap of a vehicle simulation digital sibling using the addition of a road slope study

CANDIDATE
Fabris Gianluca

SUPERVISOR
Prof. Riccio Vincenzo

CO-SUPERVISOR
Dr. Jonni Sclauzero

Academic Year 2023-2024

Institute Contacts
Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine — Italia
+39 0432 558400
https://www.dmif.uniud.it/

To my Mom,
who with love, patience and infinite dedication has always supported me in every step of my journey. Thank you for teaching me strength, perseverance and the importance of never giving up. This achievement is as much yours as it is mine. With all my love and gratitude.

Abstract

Simulation platforms are essential for developing driver training, advanced driver-assistance systems (ADAS) and autonomous driving agents. They can accurately represent real scenarios without huge costs in case of failed tests. A custom soft-body physics engine, detailed vehicle modeling and high adaptability features ensure realistic vehicle behavior and extensive customization options, making it ideal for ADAS development and immersive driver training.

In traffic simulation, a rigidly defined road model is essential for accurate and reliable results. It allows for detailed representation of road features like lane width and intersections, enabling realistic scenario simulations and precise traffic flow predictions. This rigor facilitates the integration of real-world data, improves calibration and validation, and supports informed decision-making and optimization of road network design and management.

In one simulator, the actual implementation of OSM import output results in undrivable roads, while in both simulators it results in flat roads although the roads are located in a hilly or mountainous area. The implementation of the elevation is needed to make the test more realistic and better undermine the autonomous driving agent. With the help of some metrics we established that there were some changes in the outcome of the test and we gained more information.

Contents

List of Tables

List of Figures

1

Introduction

The growing complexity of driver training systems, Advanced Driver Assistance Systems (ADAS), and autonomous driving technologies has emphasized the need for highly detailed and realistic simulation platforms. These platforms not only provide a safe and controlled environment for the testing and development of autonomous agents but also allow for cost-effective experimentation without real-world risks. Simulations can model intricate driving scenarios, including road topology, traffic patterns, and vehicle dynamics.

This thesis focuses on developing and refining simulation tools to better model road and traffic environments, using platforms such as BeamNG.tech and Hexagon's Virtual Test Drive (VTD). A key aspect of this work is the customization of OpenStreetMap (OSM) imports, which improves the accuracy of road geometries and elevation profiles, crucial for realistic simulation environments. The implementation of these features enables more reliable testing of ADAS and autonomous vehicles in various driving conditions, including those with challenging terrain.

This research not only aims to improve the realism of these simulations but also explores the differences in performance and outcomes between standard and customized imports. Through this work, the goal is to enhance the overall utility of simulation platforms for both academic research and industry applications in autonomous vehicle development.

1.1 Outline

- In chapter 2 you will find the background and the state of the art.
- In chapter 3 you will find the implementation of the custom OSM importer for BeamNG.
- In chapter 4 you will find the implementation of the custom OSM importer for Hexagon.
- In chapter 5 you will find the analysis.
- In chapter 6 you will find the conclusions and future developments.

2

Background and state of the art

This chapter contains the necessary information to understand the background necessary to understand the thesis and the state of the art.

2.1 BeamNG.tech's simulator and BeamNGpy

ADAS researchers and developers use BeamNG.tech as a platform for model-in-the-loop testing and development. BeamNG.tech is a driving simulator with very detailed physics, especially in the geometry of the vehicle, the materials of the components and the forces in action. It provides driving simulation software and virtual tests for the development and testing of autonomous vehicles, ADAS and vehicle dynamics. At the basis of vehicle physics, there is a soft-body engine physics characterized by nodes (mass points) and springs (beams), an example you can see in figure 2.1.

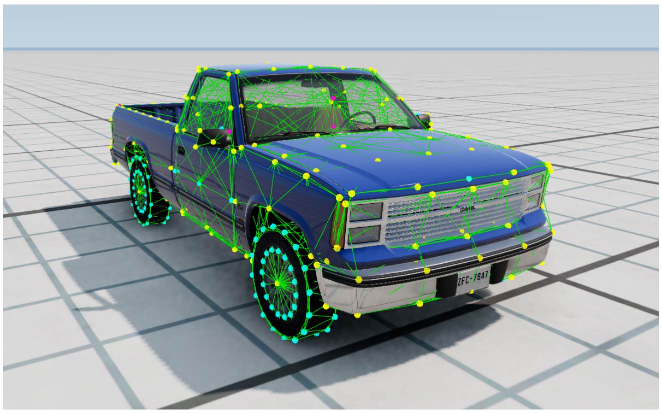


Figure 2.1: Example of nodes and beams

This physics engine allows you to obtain results that are as faithful to reality as possible, both in terms of vehicle behavior (driving dynamics, weight distribution, force propagation, kinematic properties, grip with the road surface, and more) and collision dynamics. BeamNG.tech provides a variety of sensors (camera, Lidar, IMU, ultrasonic, and more) with support for ground truth (instance annotations and bounding boxes for camera data) to be used for ADAS and autonomous driving agents. For more details see BeamNG.tech and [4], [13]. The ego car can be driven by the internal autonomous driving agent that knows everything about the environment (e.g. path of the road), or by an external autonomous driving agent that relies only on sensors (used as inputs of a neural network that decides how to act on controls, like steering, throttle and brake; an example is Dave2).

BeamNGpy is a Python API for the BeamNG.tech simulator, whose main features are remote control of vehicles, AI-controlled vehicles, dynamic sensor models, access to road network and scenario objects. An internal representation of BeamNG.tech and BeamNGpy integration can be found in figure 2.2. For more details see https://github.com/BeamNG/BeamNGpy.

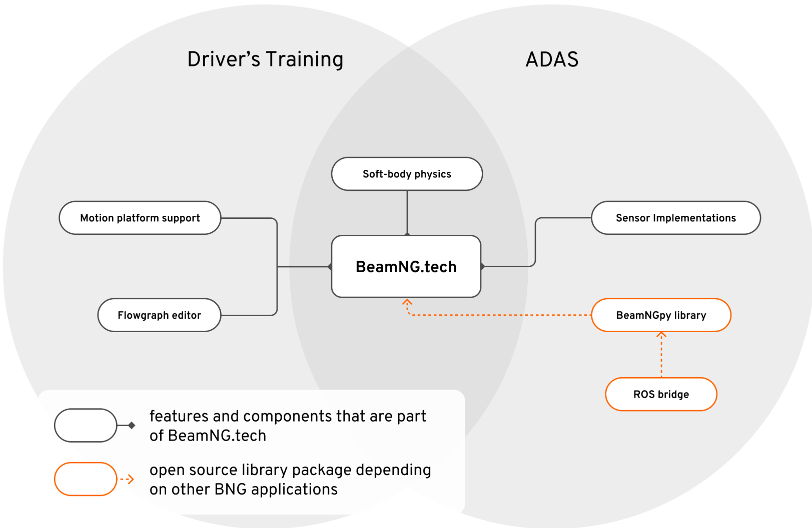


Figure 2.2: BeamNG.tech and BeamNGpy diagram

2.2 SBFT and CPS

The workshop on Search-Based and Fuzz Testing (SBFT) merges Search-Based Software Engineering (SBSE) with Fuzz Testing (FT), reflecting their shared goals and techniques. SBFT explores diverse testing objectives such as coverage, fault detection and non-functional properties. Its main aim is to convene researchers and practitioners from Search-Based Software Testing (SBST), Fuzzing Testing, and Software Engineering to advance automated testing. The event includes research presentations, keynote speeches, tool competitions and panel discussions to foster innovation and collaboration in SBFT research. For more details see https://sbft24.github.io/. Within the SBFT there is a Cyber-physical systems (CPS) testing competition, that is, a competition on self-driving car simulation environments. The CPS competition is focused on testing the lane keeping ADAS of autonomous driving agents (the BeamNG.tech internal one and DAVE-2 ([?]) a custom made one). Competitors should propose a test generator that produces virtual roads to test the system. For more details see https://github.com/sbft-cps-tool-competition/cps-tool-competition. A diagram of the pipeline of SBFT CPS testing competition is shown in the following figures 2.3 and 2.4.

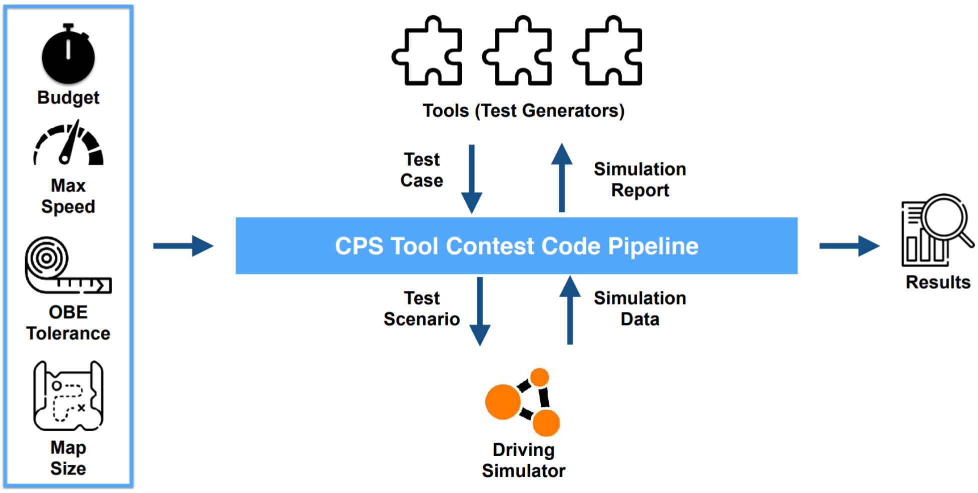


Figure 2.3: SBFT CPS testing competition infrastructure

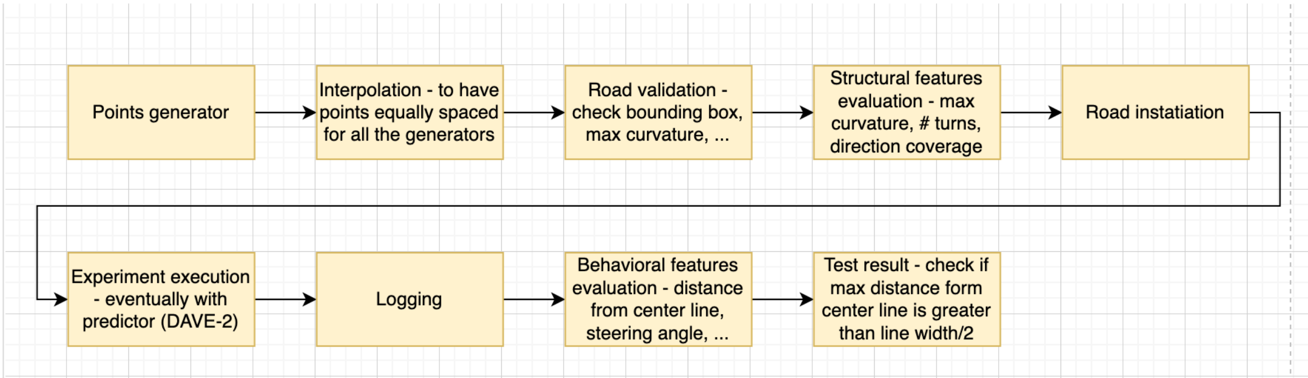


Figure 2.4: SBFT CPS testing competition pipeline

2.3 OSM

OpenStreetMap (OSM) is a collaborative project that creates a free editable map of the world. It is mainly used for navigation, urban planning, disaster response and academic research. OSM data is stored in an XML format, whose main elements are:

- `<osm>`: it is the root element
- `<header>`: contains information such as the version, data format and any metadata related to the dataset
- `<node>`: represents individual points on the map (e.g. points of interest, street corners or other geographical features) and the main attributes are unique ID, latitude, and longitude
- `<way>`: represents linear features on the map (e.g. roads, rivers or boundaries), is defined as a series of connected nodes and the main attributes are tags describing the feature
- `<relation>`: represents complex geometric or logical relationships between nodes, ways, and other relations; it contains a list of member elements

- `<tag>`: is a key-value pair used to describe nodes, ways, and relations, providing additional information about the features represented

More specifically about ways, the main features are:

- boundary, landuse
- building, shops and variants
- geological, natural, waterway
- aeroway
- highway
  - motorway
  - trunk
  - primary
  - secondary
  - tertiary
  - residential
  - footway
  - cycleway
  - raceway
- railway

for more details see https://wiki.openstreetmap.org/wiki/Map_features.

2.4 Uniform Catmull-Rom spline

To compute a smooth Catmull-Rom spline, you start with a set of control points. These points define the vertices of the spline. Each curve segment between two consecutive control points is influenced by four control points because the Catmull-Rom spline uses a basis function that depends on these four points to determine the position of the curve. The uniform Catmull-Rom spline is continuously differentiable up to the first degree (C^1) and has geometric continuity up to the first degree (G^1). This means that the curve is smooth and has no edges and that the tangents of the curve segments are continuous.

The smooth Catmull-Rom spline basis function is defined in such a way that each control point affects only local curve segments. This is achieved through a linear combination of control points. The basis function for a uniform Catmull-Rom spline is typically a cubic polynomial function that assigns appropriate weights to the four control points that define the curve segment. It is described as follows

$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} 1/2 \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

to obtain the first and second derivatives, simply derive the first part

$$P'(t) = \begin{bmatrix} 0 & 1 & 2t & 3t^2 \end{bmatrix} 1/2 \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$P''(t) = \begin{bmatrix} 0 & 0 & 2 & 6t \end{bmatrix} 1/2 \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

The curvature at a point on the curve indicates how much the direction of the tangent varies as the position along the curve varies. The Osculating Circle is a geometric concept associated with the curvature of a spline. At any point on the curve, the Osculating Circle is the circle that best approximates the curve at that point. To obtain the radius of the circle, follow the following formula

$$r = \frac{||P'||^3}{\begin{vmatrix} P'_x & P''_x \\ P'_y & P''_y \end{vmatrix}}$$

2.5 Hexagon's simulator, ROD and VTD

Hexagon is a multinational company that provides advanced information technologies for various sectors, with a particular focus on metrology and manufacturing, geospatial, security and surveillance, construction, precision agriculture, augmented and virtual reality. Hexagon's "Autonomous Solutions" project represents one of the company's most innovative initiatives, aimed at developing and implementing autonomous and automated technologies in various industrial sectors. This project leverages a combination of advanced sensors, artificial intelligence, machine learning and software to create solutions that improve efficiency, safety and productivity. The main applications and objectives of the project are autonomous vehicles (see figure 2.5), autonomous navigation systems, industrial automation, autonomous monitoring and control, data analysis and decision making.

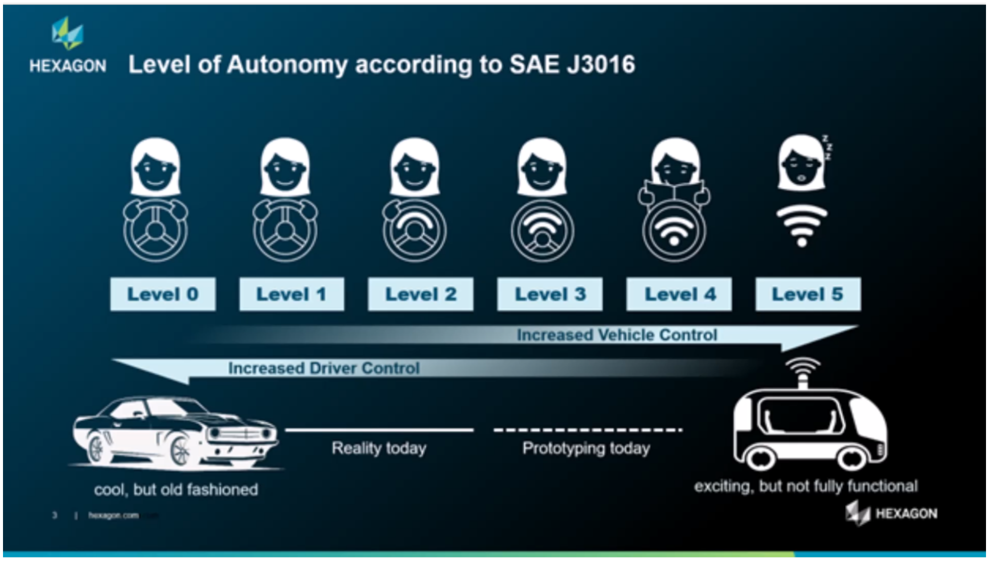


Figure 2.5: ADAS levels

Virtual Test Drive (VTD) is a software developed by Hexagon and it is used to describe the scenario both statically and dynamically and to run the simulation on the scenario and collect the data; it is composed of many modules, both internal and external, regarding the vehicles dynamics model, the driver model (same concept as autonomous driving agent), graphics and many more (see figure 2.6).

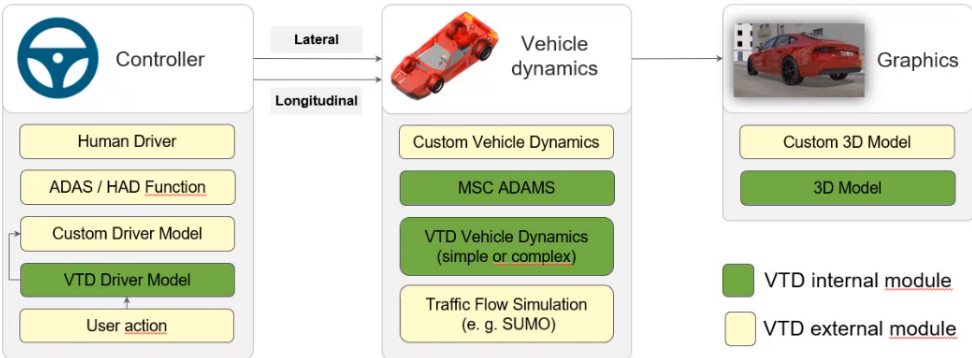


Figure 2.6: VTD plugins

Road Designer (ROD) is a module of VTD and it is used to describe all the static aspects of the scenario (roads, terrain, building, signals, ...) using the OpenDRIVE standard; it can be used to create the roads manually or to import the roads from an OSM. Scenario Editor is a module of VTD and it is used to describe all the dynamic aspects of the scenario (stoplights, traffic, pedestrians, sensors, weather, vehicle dynamics, ...) using the OpenScenario standard. In the figure 2.7 you can see the relation between OpenDRIVE and OpenSCENARIO.

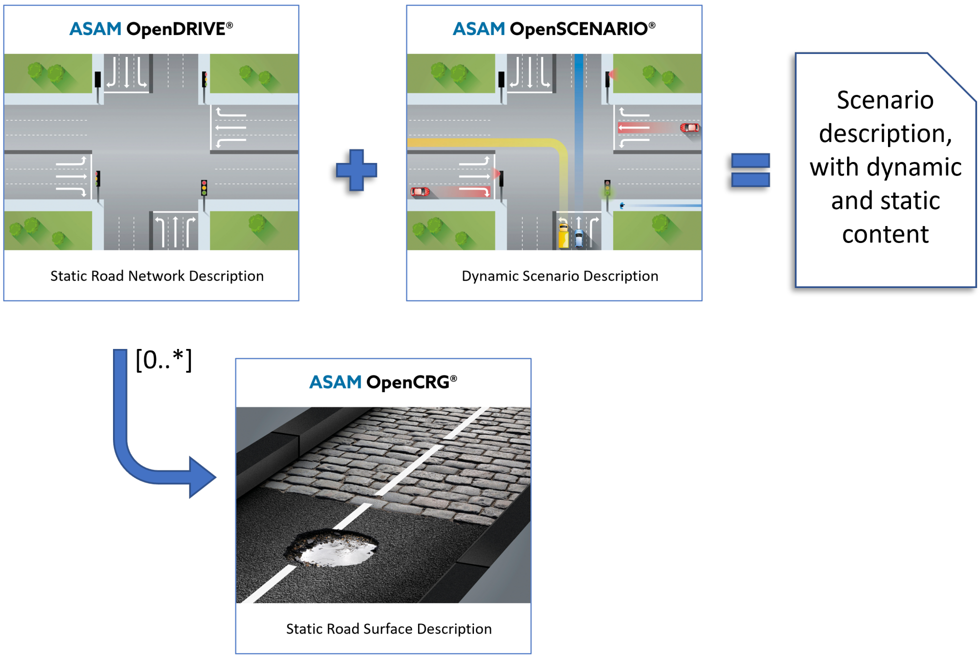


Figure 2.7: OpenDRIVE and OpenSCENARIO

VTD has two types of vehicle dynamics models, the first one has only one rigid body and the second one has the four wheels (flexible bodies) and the chassis (rigid body); more complex simulation can be done using a vehicle dynamics model inside ADAMS (software developed by Hexagon). In VTD, the ego car can be driven by an internal driver model which knows everything about the environment (e.g. path of the road), or by an external driver model that relies only on sensors (e.g. camera, infrared camera, LIDAR, radar and more) that can be used as inputs of a neural network that decides how to act on controls (e.g. like steering, throttle and brake) on the vehicle driving model. This type of driver model is achievable thanks to an external module that interfaces with VTD (it needs to be implemented separately and its functions can be the most diverse, such as line keeping, collision avoidance, and much more).

2.6 OpenDRIVE

OpenDRIVE is an open standard for describing digital roads and road networks, used primarily in driving simulations, autonomous driving systems and other traffic engineering applications. The main characteristics are detailed road description, modeling of road elements, topology and connectivity. Such a detailed description of the road is needed to have a good scenario and simulation. OpenDRIVE data is stored in an XML format, whose main elements are:

- Road coordinate system: s coordinate follows the road reference line, t coordinate is perpendicular to s, h coordinate describes elevation; the road can also have superelevation (see figure 2.8)

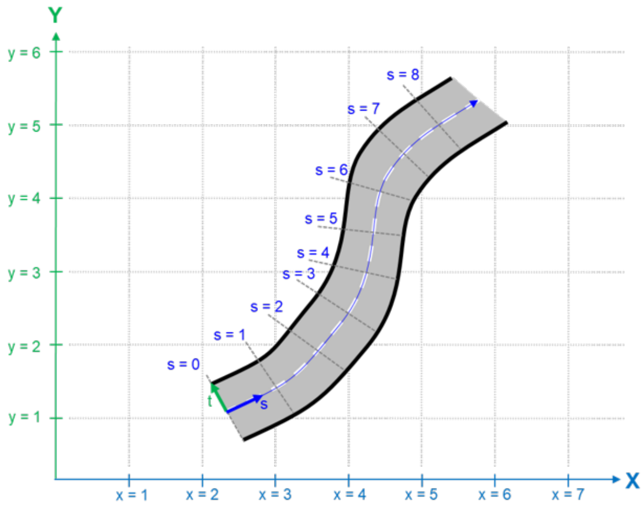


Figure 2.8: OpenDRIVE road coordinate system

- Geometries: reference line of the road can be described by straight lines, spirals or clothoids with a linearly changing curvature, arcs with a constant curvature, parametric cubic polynomials (see figure 2.9)

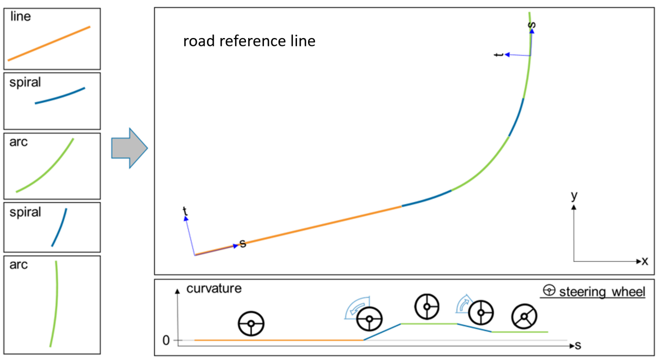


Figure 2.9: OpenDRIVE geometries

- Roads: roads must be linked via the reference line smoothly and can have many properties (type, speed, elevation, superelevation, cross section, ...); elevation, superelevation and cross section can be described by a cubic polynomial (see figure 2.10)

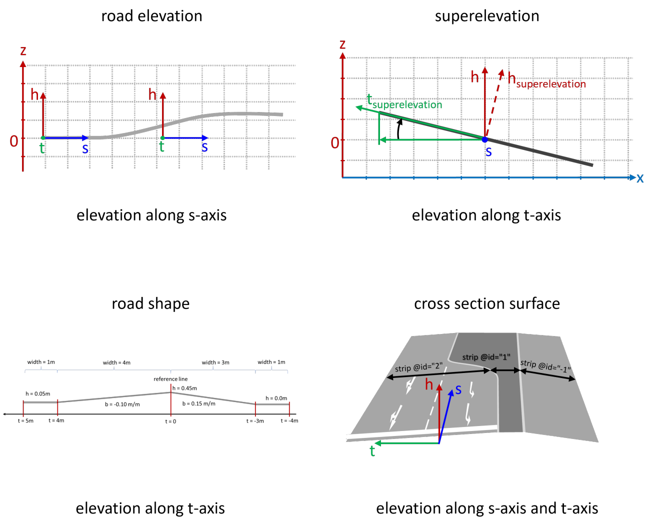


Figure 2.10: OpenDRIVE roads

- Lanes: roads have many lanes which are described by an id (based on the reference line); lanes can have many properties (type, width, speed, access, marking, section, offset, ...); width and offset can be described by a cubic polynomials (see figures 2.11, 2.12, 2.13, 2.14 and 2.15)

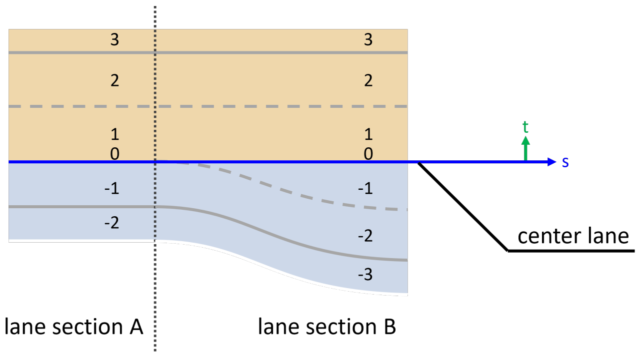


Figure 2.11: OpenDRIVE lanes

[H]

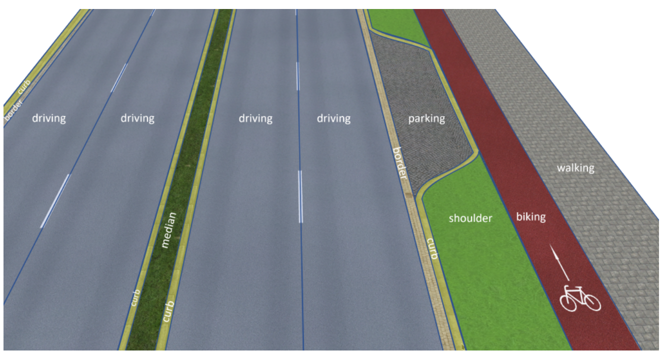


Figure 2.12: OpenDRIVE lanes - urban

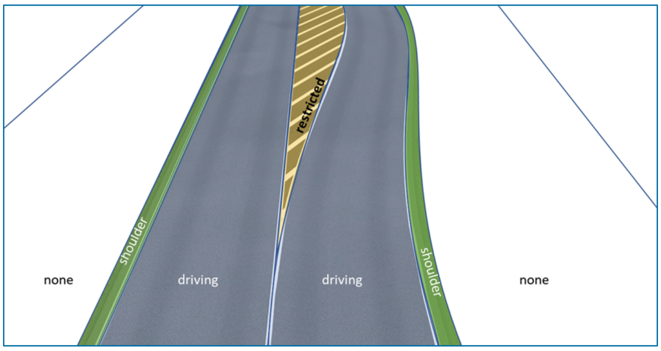


Figure 2.13: OpenDRIVE lanes - rural

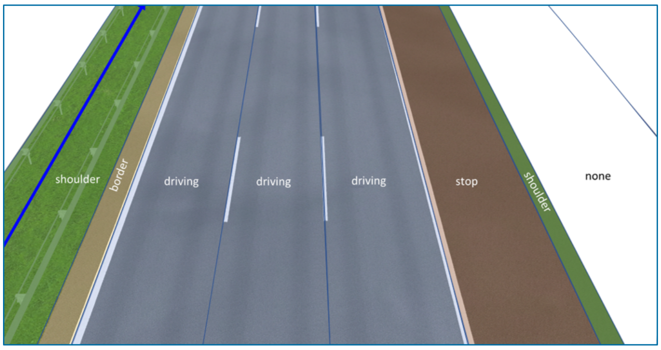


Figure 2.14: OpenDRIVE lanes - motorway

Figure 2.15: OpenDRIVE lanes - motorway entry and exit

- Junctions: road intersections can be made via junctions; they describe how the roads are connected, priority (which car need to give the way). There are three main types: direct, virtual (used only for driveways) and crossing (e.g. used to cross railways); junctions can have many proprieties (connecting roads, reference line, boundary, elevation (only latest version), and more) (see figures 2.16, 2.17, 2.18, 2.19 and 2.20)

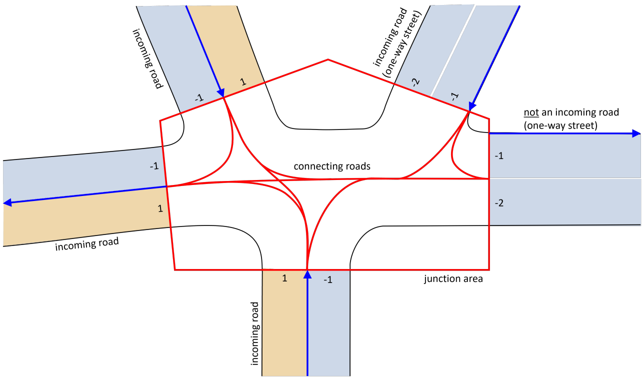


Figure 2.16: OpenDRIVE junctions

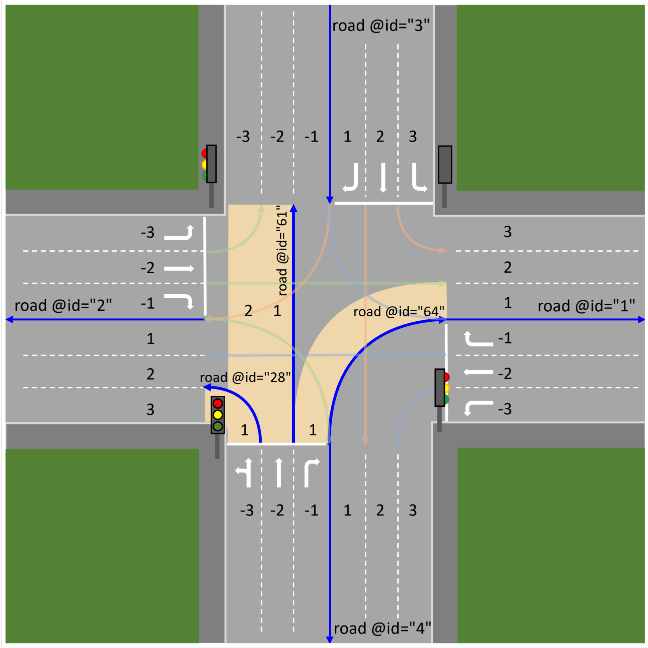


Figure 2.17: OpenDRIVE junctions - connections

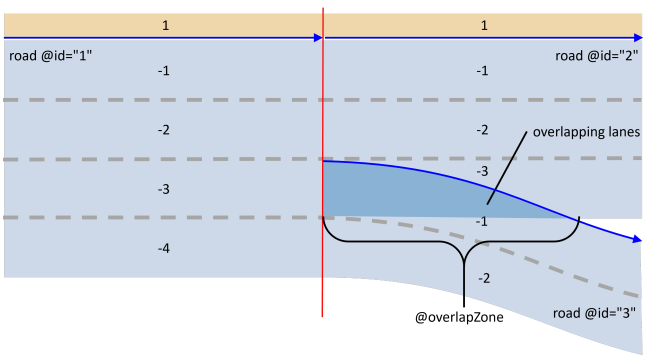


Figure 2.18: OpenDRIVE junctions - direct

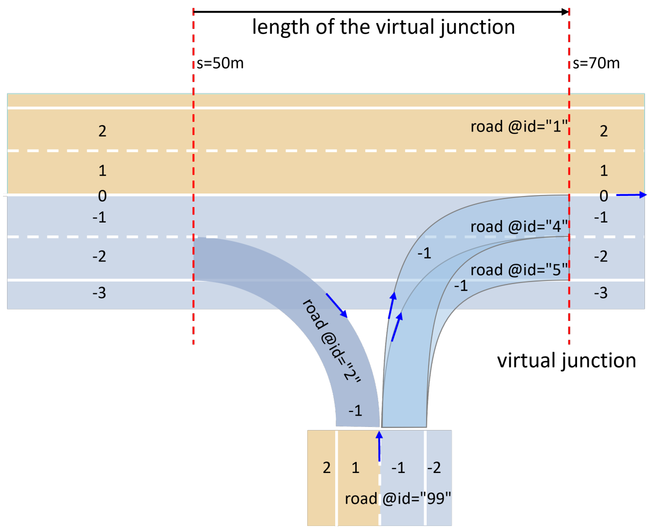


Figure 2.19: OpenDRIVE junctions - virtual

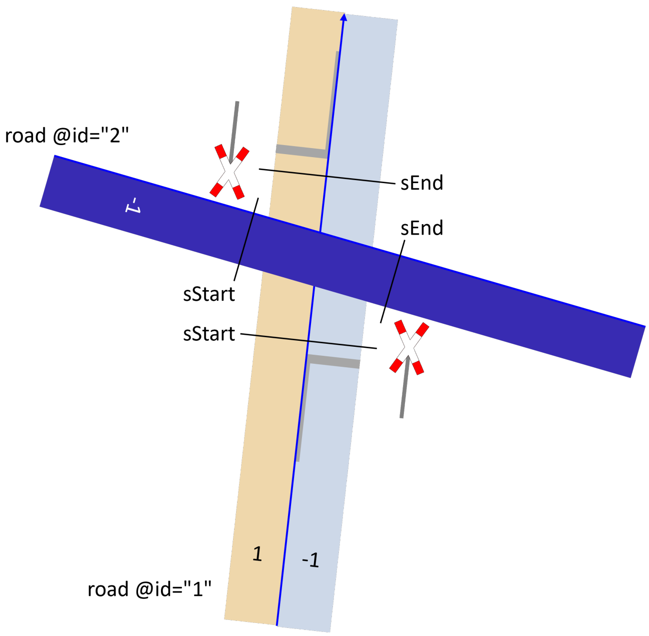


Figure 2.20: OpenDRIVE junctions - crossing

- Objects: are other elements (e.g. traffic island, traffic light pole, tree, building, tunnel, bridge); objects can have many properties (outline, skeleton, material, access, markings, bounding, and more) (see figure 2.21)

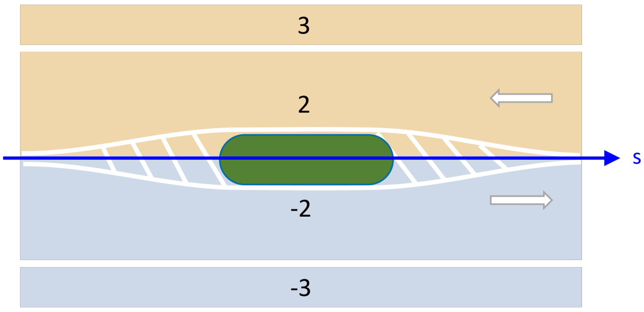


Figure 2.21: OpenDRIVE objects

- Signals: describe horizontal and vertical signals for regulating road traffic (e.g. traffic signs, traffic lights, road marking, and more) (see figure 2.22)

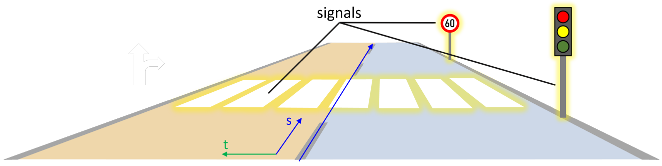


Figure 2.22: OpenDRIVE signals

and many more. For more details see ASAM OpenDRIVE documentation and [2].

2.7 Digital twin and digital sibling

A digital twin is a highly detailed, real-time virtual representation of a physical asset, system, or process, allowing for enhanced monitoring, simulation, and analysis. This technology integrates data from various sources to mirror the physical counterpart's characteristics and behaviors, and facilitate predictive maintenance, optimization, and decision-making. Digital twins have found extensive applications across industries such as manufacturing, healthcare, and smart cities.

The notion of digital siblings extends the digital twin concept by incorporating multiple interconnected digital replicas that represent different aspects or instances of a physical entity. This approach enhances collaborative analysis and comprehensive insights, particularly in complex systems with numerous interacting components. For instance, in the field of autonomous vehicle (AV) testing, digital siblings employ multiple simulators to improve the accuracy and reliability of the digital twin's predictions. This multi-simulator approach, as demonstrated in testing the Nvidia DAVE-2 deep neural network for lane-keeping, effectively bridges the simulation fidelity gap by cross-verifying results across different platforms and predicting real-world AV failures with higher fidelity than single-simulator methods ([5]). The continuous advancement in IoT, AI, and data analytics drives the evolution and sophistication of these digital models, underscoring their critical role in the modern technological landscape.

As stated in [5], the development of autonomous vehicles (AVs) has received great attention in the last decade. As of 2020, more than $150 billions have been invested in AVs, a sum that is expected to double in the near future (Boutan https://medium.com/swlh/autonomous-driving-market-overview-b8c71d81c072). AVs typically integrate multiple advanced driver-assistance systems (e.g., for adaptive cruise control, parking assistance, and lane-keeping) into a unified control unit, using a perception-plan-execution strategy (Yurtsever et al. [18]). Advanced driver-assistance systems based on Deep Neural Networks (DNNs) are trained on labeled input-output samples of real-world driving data provided by the vehicle sensory to learn human-like driving actions (Grigorescu et al. [10]). Before deployment on public roads, AVs are thoroughly tested in the field, on private test tracks (BGR Media https://techcrunch.com/2018/10/10/waymos-self-driving-cars-hit-10-million-miles/; Borg et al. [6]; Cerf [9]; Stocco et al. [15]). While essential for fully assessing the dependability of AVs on the road, field testing has known limitations in terms of cost, safety and adequacy (Stocco et al. [15]). To overcome these limitations, driving simulators are used to generate several real-life edge case scenarios that are unlikely to be experienced during field testing, or that are dangerous to reproduce for human operators (Borg et al. [7]; Koopman and Wagner [12]). Simulation-based testing represents a consolidated testing practice, being more affordable than field testing, yet capable of exposing many bugs before deployment (BGR Media https://techcrunch.com/2018/10/10/waymos-self-driving-cars-hit-10-million-miles/; Borg et al. [8]; Cerf [9]; Stocco et al. [15]). Related work of [5], digital twins are used by researchers to reproduce real-world conditions within a simulation environment for testing purposes (Barosan et al. [3]; Yun and Park [17]; Kapteyn et al. [11]; San [14]; Almeaibed et al. [1]; Veledar et al. [16]).

2.8 Problem definition

In one simulator, the actual implementation of OSM import output results in undrivable roads, while in both simulators it results in flat roads although the roads are located in a hilly or mountainous area. The implementation of the elevation is needed to make the test more realistic and better undermine the autonomous driving agent.

3

Import OSM custom BeamNG

3.1 Raw OSM importer

Inside BeamNGpy it is possible to import OSM files and OpenDRIVE files. This implementation of the OSM importer has some issues:

- it imports everything as a road (even buildings, waterways, railways, ... - more than 1300 ways are imported) and it does not apply any material to the `MeshRoad`
- it does not create any `Road` (decal road) that is used to create the paths for the internal autonomous driving agent to follow
- it imports the map flipped vertically and the proportions horizontal/vertical are wrong

In figure 3.1, you can see an example of import OSM.

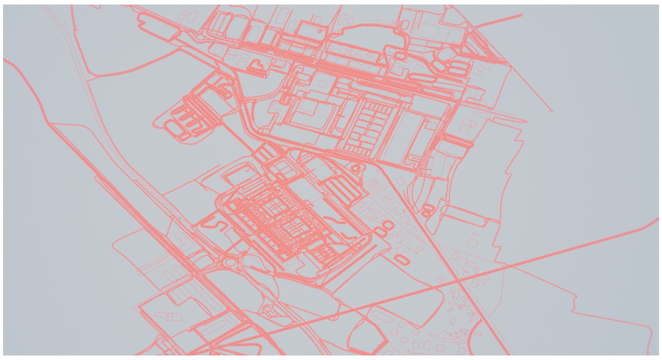


Figure 3.1: Example of import OSM with standard importer - uniud

3.2 Custom OSM importer

To overcome these limitations, for the SBFT-CPS competition, I was responsible for implementing a custom OSM importer that could also use the elevation. My implementation is divided into two parts: download and import; it offers the following benefits:

- it imports only the filtered roads (about 125 ways with standard filter, mileage might vary with custom filters) and it applies with material to the `MeshRoad`

- it crates the `MeshRoad` so that the internal autonomous driving agent can drive it
- it implements the elevation and it is possible to choose how to use the elevation (`both` recommended)
- it creates a terrain (with elevation) that follows the road so that the ego car can follow the road even at high altitude without trying to follow it on the ground (without elevation)

In figures 3.2 and 3.3, you can see an example of import OSM both with and without elevation.

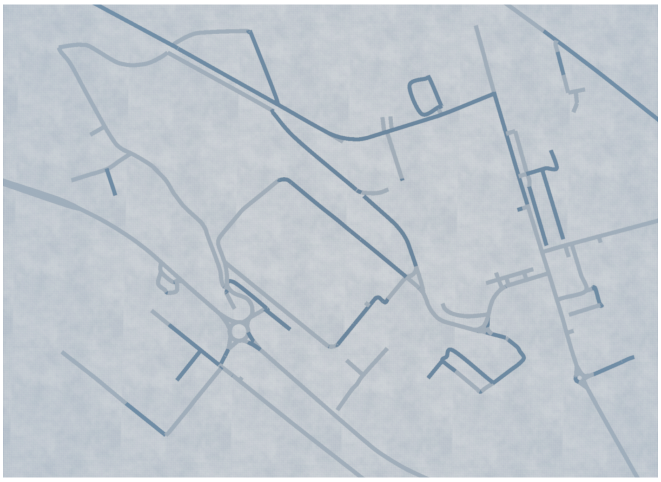


Figure 3.2: Example of import OSM with custom importer - uniud

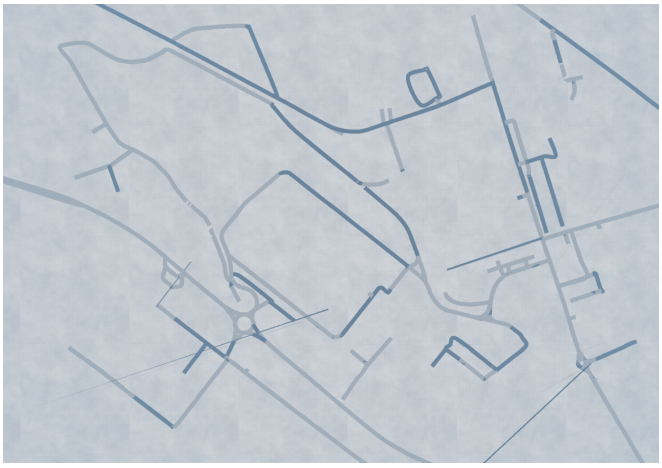


Figure 3.3: Example of import OSM with custom importer with elevation - uniud

The inputs for the download phase are: an area defined by a minimum and maximum latitude and longitude, alternatively can be used a name of a city (one of these is mandatory, if both inputs are given, the city will be used) and a custom filter (optional). It is also possible to decide which elevation API will be used and whether to use the elevation (`none`, `road`, `terrain`, `both`). A JSON file describing the OSM with elevation is given as an output. If the city is not `None`, the https://nominatim.openstreetmap.org/search API is queried and will provide the city id as output. After creating the appropriate query, given bounding box or city id and filter, for the https://overpass-api.de/api/interpreter API is queried and will provide the OSM data already filtered as output. To obtain the elevation data, all the coordinates of all the roads are grouped into an array, then many queries will be created for the elevation API according to the `requestLimit` parameter. Due to the `requestLimit` and the `rateLimit` this process might take some time, but this is the fastest way to do it, e.g. grouping coordinates by road would take even longer because some queries are not full. Then some transformations to the data are done

- equirectangular projection

$$x = Er\left(x - \frac{lon_{min} + lon_{max}}{2}\right)\cos\left(\frac{lat_{min} + lat_{max}}{2}\right)$$

$$y = Er\left(y - \frac{lat_{min} + lat_{max}}{2}\right)$$

  where `Er` is 6372.797 (Earth radius)

- from coordinates to meters

$$x = x(111412.84\cos(lat) - 93.5\cos(2lat) + 0.118\cos(4lat))$$

$$y = y(111132.92 - 559.82\cos(2y) + 1.175\cos(4y) - 0.0023\cos(6y))$$

- translation to the origin

$$x = x - lon_{min_{norm}}$$

$$y = y - lat_{min_{norm}}$$

$$z = z - elevation_{min_{norm}}$$

To create the terrain, a grid is created, from the maximum and minimum points of the road plus 100 meters of margin in all directions with points spaced every `scale` meters.

$$w = \left\lfloor \frac{\lfloor lon_{max_{norm}} - lon_{min_{norm}}) + 1 + 2 * 100\rfloor}{scale + 1} \right\rfloor$$

$$h = \left\lfloor \frac{\lfloor lat_{max_{norm}} - lat_{min_{norm}}) + 1 + 2 * 100\rfloor}{scale + 1} \right\rfloor$$

To get the elevation data the values are mapped back to the coordinates value

$$x = \frac{x - \frac{100}{scale}}{\left(w - 1 - \left(\frac{100}{scale}\right)\right) - \frac{100}{scale}}(lon_{max} - lon_{min}) + lon_{min}$$

$$y = \frac{y - \frac{100}{scale}}{\left(h - 1 - \left(\frac{100}{scale}\right)\right) - \frac{100}{scale}}(lat_{max} - lat_{min}) + lat_{min}$$

To obtain the elevation data the process is the same as for the road. Then some transformations to the data are done

- the roads are aligned with the terrain

$$x = x + 100$$

$$y = y + 100$$

- translation to the origin

  - if it is used with the elevation set as `both`, the roads are translated into the terrain origin (mapped)

$$z = z - \left(\frac{te_{min} - re_{min}}{re_{max} - re_{min}} (re_{max_{norm}} - re_{min_{norm}}) + re_{min_{norm}} \right)$$

  - if it is used with elevation set as `terrain` or `both`, the terrain is translated into the origin

$$z = z - te_{min}$$

  - if it is used with elevation set as `road` or `none`, the roads are translated into the terrain default height

$$z = z + (te_{max} - te_{min})$$

where te stands for terrain elevation and re stands for road elevation.

The import part uses the JSON created previously to create the roads and terrain in the scenario using BeamNGpy classes `Road` and `Terrain_Importer`.

You can see a diagram of the custom OSM importer in the following figures 3.4 and 3.5. The code can be retrieved on GitHub github.com/gianlucafabris/Import-OSM-custom.

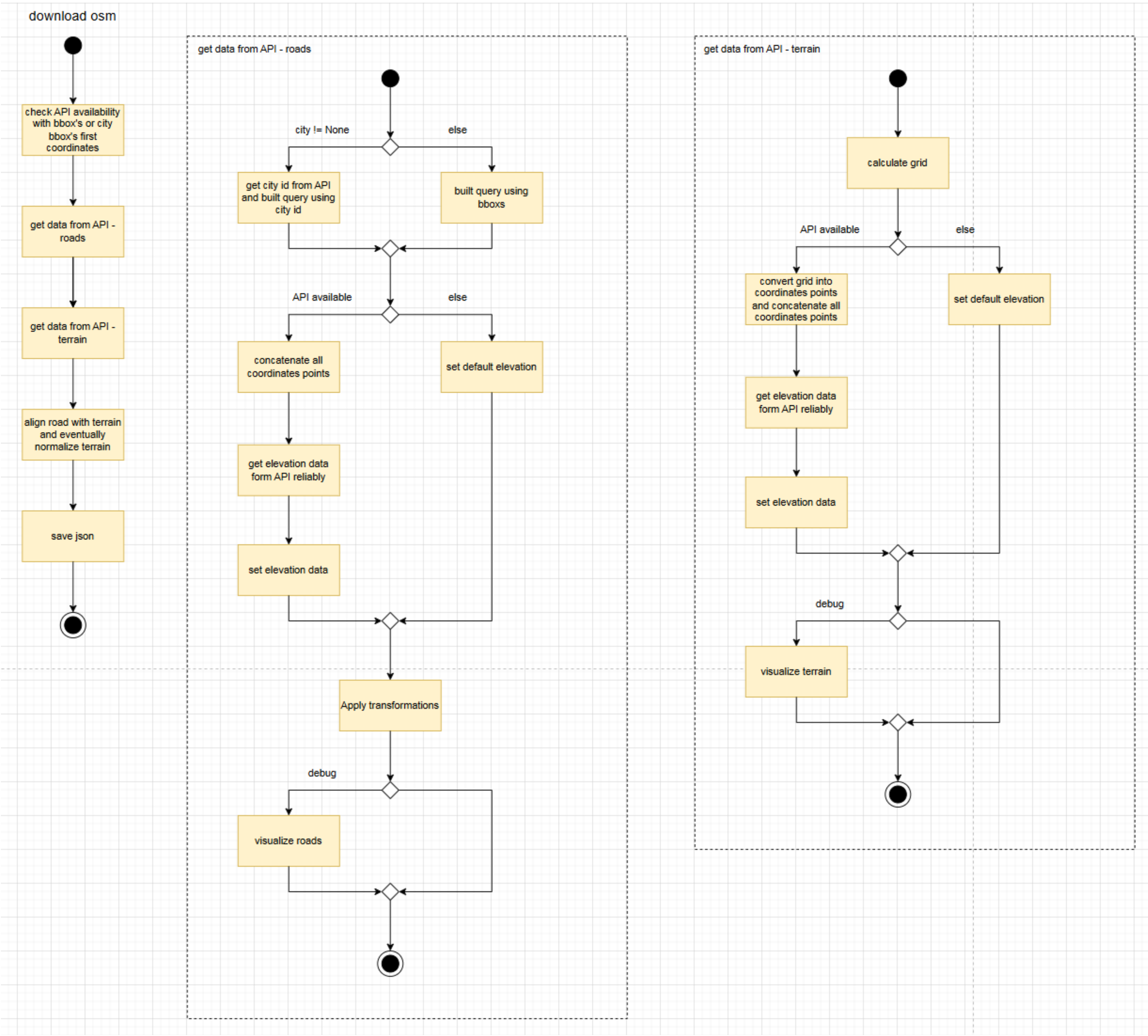


Figure 3.4: Diagram of the custom OSM importer - download

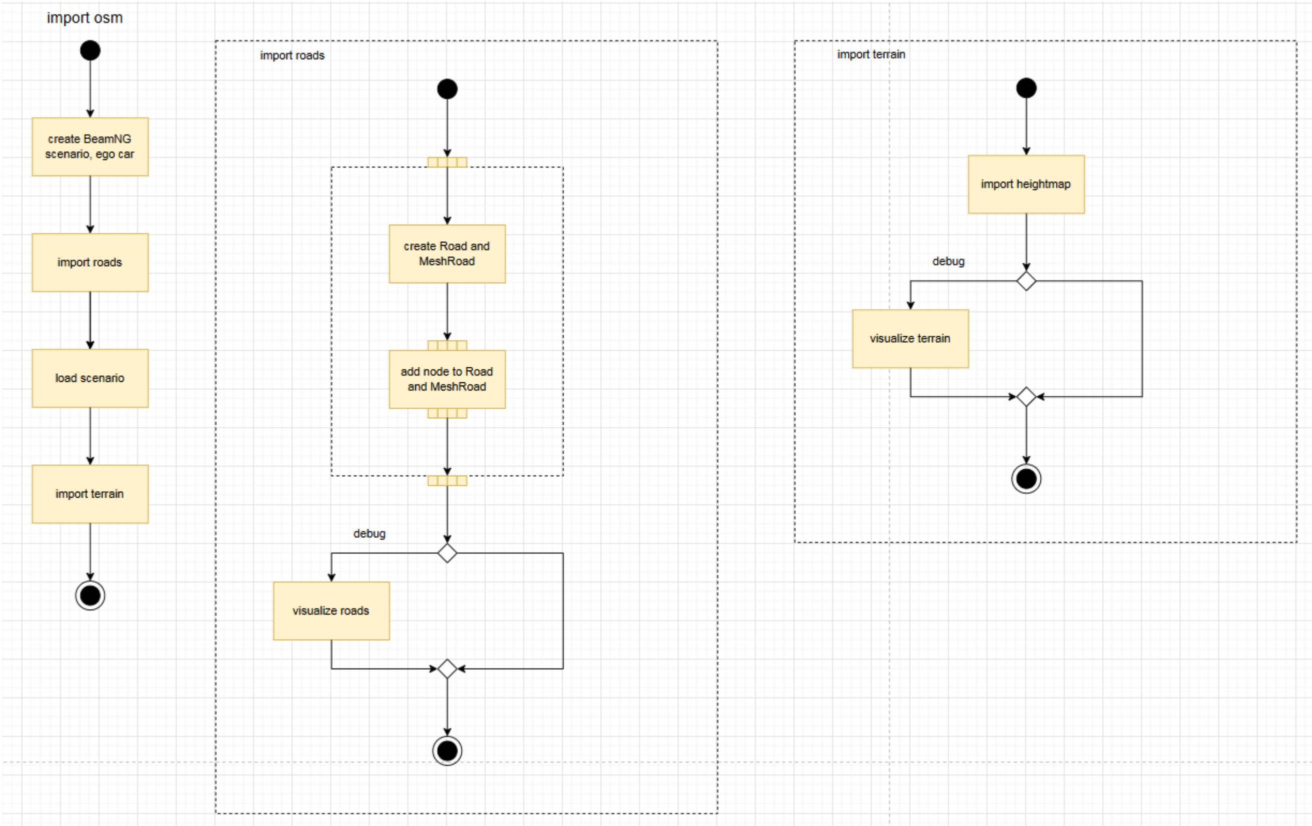


Figure 3.5: Diagram of the custom OSM importer - import

4

Import OSM custom Hexagon

4.1 Raw OSM importer

Inside ROD, it is possible to import OSM files and OpenDRIVE files. This implementation of the OSM importer is already good, with the following features:

- it imports only the roads (with some filters applied), about 55 roads are imported as roads
- it creates the roads following the rigid OpenDRIVE standard, but it makes all the roads of the same type
- thanks to the OpenDRIVE standard, in VTD, the internal autonomous driving agent can drive the roads

In figures 4.1 and 4.2, you can see an example of import OSM.

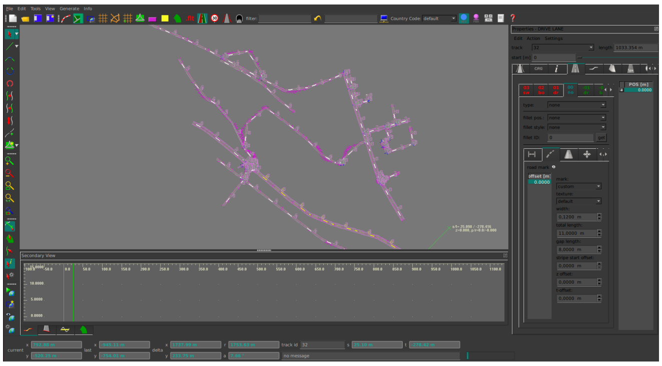


Figure 4.1: Example of import OSM with standard importer - uniud

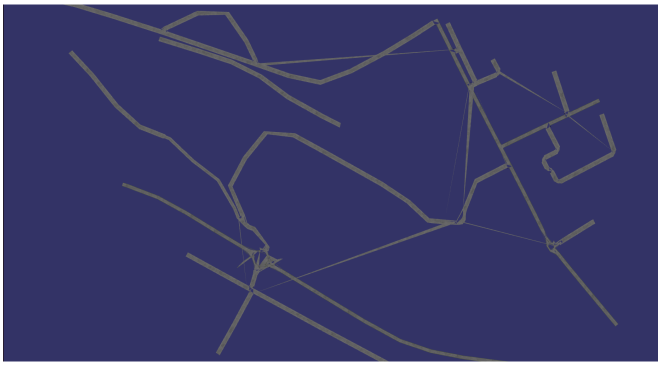


Figure 4.2: Example of import OSM with standard importer - uniud

4.2 Custom OSM importer

MapImporter has overcome the limitations of the native implementation, improving its modularity and extensibility. It is a standalone software that uses an OSM and a file XML that describes how to filter it and how to convert it into OpenDRIVE format as input and provides an OpenDRIVE file that can be imported in ROD as output. This implementation expands the native ROD implementation with the following features:

- it provides the possibility to have a custom filter and to define with which civil engineering rules the roads must be converted (e.g. German or Italian rules), around 50 ways are imported as roads
- it merges two roads that are divided (in the original OSM there is a junction with other ways that are filtered out) into a continuous road
- it converts the roads according to their type (e.g. with bicycle lane, with parking, ...) and some other elements are converted (e.g. pedestrian crossings, junctions' arms, ...)

In figures 4.3 and 4.4, you can see an example of import OSM without elevation.

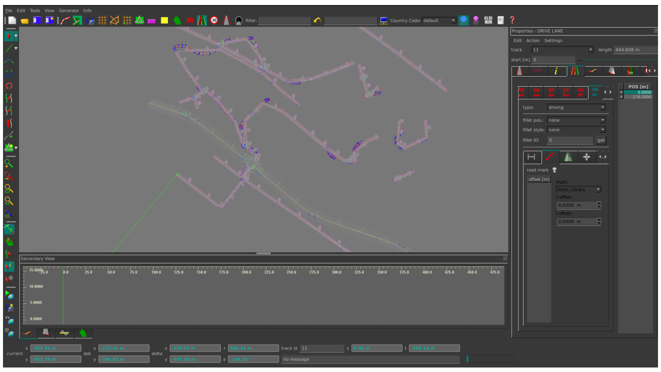


Figure 4.3: Example of import OSM with custom importer (MapImporter) - uniud

Figure 4.4: Example of import OSM with custom importer (MapImporter) - uniud

Here the term 'ways' is used to refer to the elements inside the OSM and the term 'roads' is used to refer to the ways after they are converted and post processed. Thanks to filters it is possible to limit the type of roads imported. Since the importers of the two simulators have different filters, the results are different; the values of the number of roads imported are thus not comparable and the scope of the thesis is not to compare the simulator, but it is to create new test cases for the simulator and analyze them. A diagram of the pipeline of MapImporter is shown in the figure 4.5.

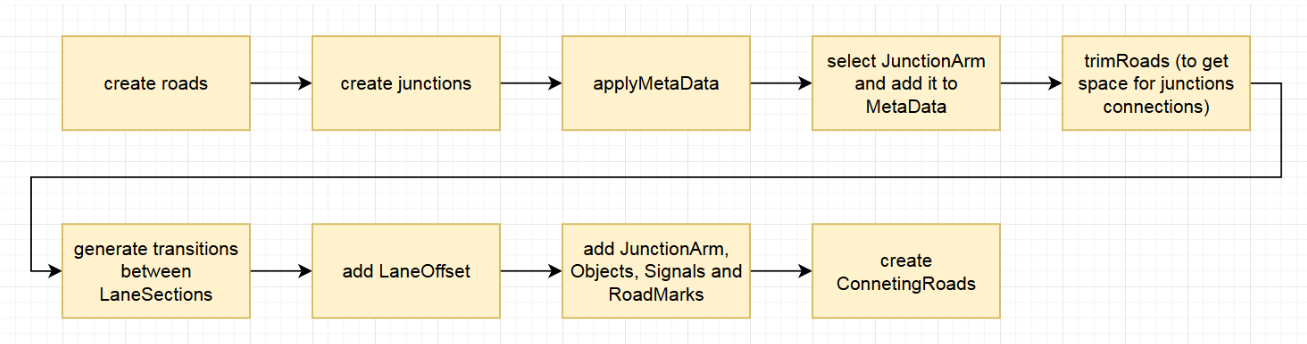


Figure 4.5: MapImporter pipeline

4.3 Elevation implementation

At this point, all the imported roads are flat although the OSM map represents roads in a hilly or mountainous area. Due to this, the elevation information is missing in the OSM file. During my internship, I was responsible for implementing the elevation inside MapImporter, which means adding the elevation information inside the Open DRIVE roads. By following the hilly or mountainous profile of the area, the road were recreated in a more faithful way and thus made the resulting simulation more realistic.

This implementation expands MapImporter with a post-processing step. Given the OSM roads, the OpenDRIVE roads and a lookup table, the script searches for all the coordinates in the OSM roads; it calls the elevation API similarly to the BeamNG implementation; then, using the lookup table, the elevation data is added to the OpenDRIVE roads. In figures 4.6 and 4.7, you can see an example of import OSM with elevation.

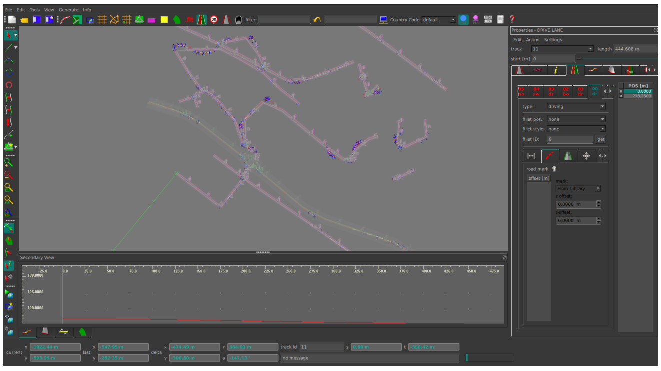


Figure 4.6: Example of import OSM with custom importer (MapImporter) with elevation - uniud

Figure 4.7: Example of import OSM with custom importer (MapImporter) with elevation - uniud

OpenDRIVE roads need an elevation profile; to obtain the smoothest possible profile, it was used a cubic function that passes in two given points and at given points the first derivative is zero, to achieve G^1 geometric continuity. Given two points $(s_i,\, h_i)$ and $(s_f,\, h_f)$, the function is described as follows:

$$f(x) = a + bx + c + x^2 + dx^3$$

$$a = \frac{3h_f s_f s_i^2 - h_f s_i^3 - 3h_i s_f^2 s_i + h_i s_f^3}{3s_f s_i^2 - s_i^3 - 3s_f^2 s_i + s_f^3}$$

$$b = -\frac{6s_f s_i (h_f - h_i)}{3s_f s_i^2 - s_i^3 - 3s_f^2 s_i + s_f^3}$$

$$c = \frac{3(h_f - h_i)(s_f + s_i)}{3s_f s_i^2 - s_i^3 - 3s_f^2 s_i + s_f^3}$$

$$d = \frac{2(h_i - h_f)}{3s_f s_i^2 - s_i^3 - 3s_f^2 s_i + s_f^3}$$

In figure 4.8, you can see the cubic function.

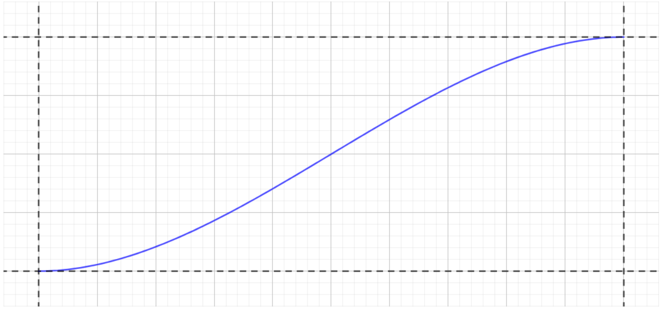


Figure 4.8: Elevation cubic function

It was decided to get the elevation only for the first and last point of each OSM way, in order to make the program faster and the road less bumpy; but it might remove some details (e.g. if the road goes up and down, only the global difference will be used; see figure below). The cubic function works with any two points not only the first and last. In figure 4.8, you can see an example of cubic function applied on all points.

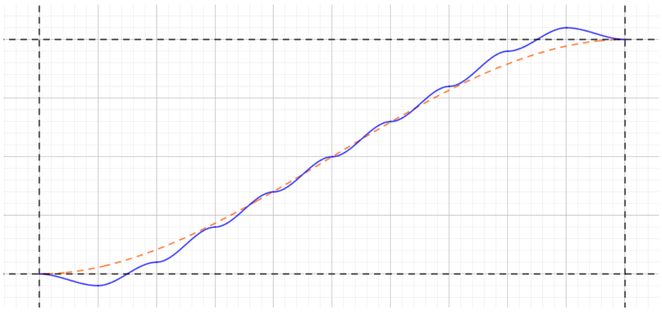


Figure 4.9: Elevation cubic function - on all points

Previously the elevation was implemented by hand and it used a civil engineering rule that defines the minimum curvature radius for that given type of road. This results in an elevation profile as follows:

- straight flat section, modelled as $a + 0x + 0x^2 + 0x^3$
- curved section, modelled as $a + bx + cx^2 + 0x^3$ such that respects the minimum radius constraint
- straight inclined section, modelled as $a + bx + 0x^2 + 0x^3$
- curved section, modelled as $a + bx + cx^2 + 0x^3$ such that respects the minimum radius constraint
- straight flat section, modelled as $a + 0x + 0x^2 + 0x^3$

In figure 4.10, you can see the civil engineering function.

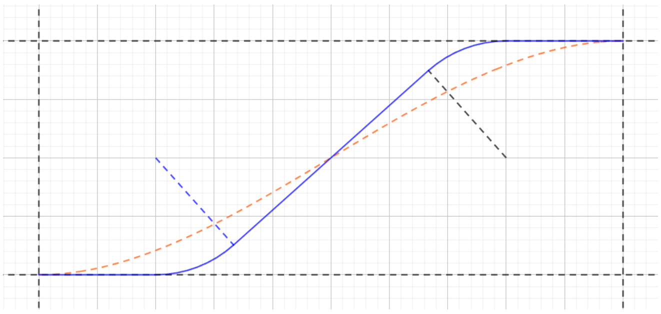


Figure 4.10: Elevation civil engineering function

The implementation with the cubic function is smoother and has a greater curvature radius; in edge cases where a smaller curvature radius than the civil engineering rule is needed, the civil engineering implementation gives an error, while the cubic function implementation tries its best to give the smoothest possible profile (even if it is less than the rule).

During the last days of the internship, I calculated the equations for the elevation profile with the civil engineering rule. This implementation is a simplified version of the above without the first and last straight segments. The following steps were taken:

- two circles with given radius r were created, so that the top or bottom part touches the points $(s_i,\, h_i)$ and $(s_f,\, h_f)$
- a line that is tangent to the circles was created (see https://en.wikipedia.org/wiki/Tangent_lines_to_circles)
- the points where the line and the circles touch were calculated
- the parameters of the line were calculated
- two quadratic functions were created to best fit the circles; these have the same first derivative at points $(s_i,\, h_i)$, first meeting point with the line $(x_1, g(x_1))$ and second meeting point with the line $(x_2, g(x_2))$, $(s_f,\, h_f)$
- the parameters of the quadratic functions were calculated

Given two points $(s_i,\, h_i)$, $(s_f,\, h_f)$ and r, the function is described as follows:

- to make it more concise, the following variables will be used

$$eq_1 = (-2h_i^2 s_f + 4h_i h_f s_f - 8h_i s_f r - 2h_f^2 s_f + 8h_f s_f r - 2s_f^3 + 4s_f^2 s_i - 2s_f s_i^2 - 8r^2 s_i)^2$$

$$eq_{2a} = -4(h_i^2 - 2h_i h_f + 4h_i r + h_f^2 - 4h_f r + s_f^2 - 2s_f s_i + 4r^2 + s_i^2)$$

$$eq_{2b} = h_i^2 s_f^2 - 4h_i^2 r^2 - 2h_i h_f s_f^2 + 8h_i h_f r^2 + 4h_i s_f^2 r - 16h_i r^3 + h_f^2 s_f^2 - 4h_f^2 r^2$$

$$eq_{2c} = -4h_f s_f^2 r + 16h_f r^3 + s_f^4 - 2s_f^3 s_i - 4s_f^2 r^2 + s_f^2 s_i^2 + 8s_f r^2 s_i$$

$$eq_3 = 2h_i^2 s_f - 4h_i h_f s_f + 8h_i s_f r + 2h_f^2 s_f - 8h_f s_f r + 2s_f^3 - 4s_f^2 s_i + 2s_f s_i^2 + 8r^2 s_i$$

$$eq_4 = 2(h_i^2 - 2h_i h_f + 4h_i r + h_f^2 - 4h_f r + s_f^2 - 2s_f s_i + 4r^2 + s_i^2)$$

$$x_t = \frac{\sqrt{eq_1 + eq_{2a}(eq_{2b} + eq_{2c})} + eq_3}{eq_4}$$

$$y_t = \sqrt{(x_t - s_f + 2r)(-x_t + s_f + 2r)} + h_f - r$$

- first "circle"

$$f(x) = a + bx + cx^2 + dx^3$$

$$a = \frac{(a_g + b_g x_1)s_i^2 + h_i x_1^2 - 2h_i x_1 s_i}{(x_1 - s_i)^2}$$

$$b = \frac{-2(a_g + b_g x_1)s_i + 2h_i s_i}{(x_1 - s_i)^2}$$

$$c = \frac{(a_g + b_g x_1) - h_i}{(x_1 - s_i)^2}$$

$$d = 0$$

- first contact point

$$x_1 = \frac{r\left(-\sqrt{\frac{(-y_t+h_i+r)^2}{(s_i-x_t)^2} + 1}\right) - \frac{s_i(s_i-x_t)}{-y_t+h_i+r} - \frac{s_i(-y_t+h_i+r)}{s_i-x_t}}{-\frac{-y_t+h_i+r}{s_i-x_t} - \frac{s_i-x_t}{-y_t+h_i+r}}$$

- line

$$g(x) = a + bx + cx^2 + dx^3$$

$$a = h_i + r - \frac{h_i + r - y_t}{s_i - x_t}s_i - r\sqrt{1 + ((\frac{h_i + r - y_t}{s_i - x_t}))^2}$$

$$b = \frac{h_i + r - y_t}{s_i - x_t}$$

$$c = 0$$

$$d = 0$$

- second contact point

$$x_2 = \frac{-\frac{x_t(s_i-x_t)}{-y_t+h_i+r} - r\sqrt{\frac{(-y_t+h_i+r)^2}{(s_i-x_t)^2} + 1} - \frac{s_i(-y_t+h_i+r)}{s_i-x_t} - y_t + h_i + r}{-\frac{-y_t+h_i+r}{s_i-x_t} - \frac{s_i-x_t}{-y_t+h_i+r}}$$

- second "circle"

$$h(x) = a + bx + cx^2 + dx^3$$

$$a = \frac{(a_g + b_g x_2)s_f^2 + x_2^2 h_f - 2x_2 h_f s_f}{(x_2 - s_f)^2}$$

$$b = \frac{-2(a_g + b_g x_2)s_f + 2h_f s_f}{(x_2 - s_f)^2}$$

$$c = \frac{(a_g + b_g x_2) - h_f}{(x_2 - s_f)^2}$$

$$d = 0$$

these equations refer to the case $h_i < h_f$, the case $h_i > h_f$ is similar with some signs changed. The result can be seen in the figure 4.11.

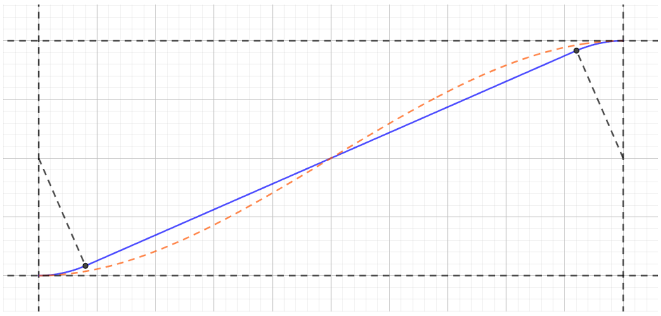


Figure 4.11: Elevation civil engineering function - simplified

Transforming this into the correct version is trivial: adjust s_i and s_f and add at the start and end the segments $h_i + 0x + 0x^2 + 0x^3$ and $h_f + 0x + 0x^2 + 0x^3$.

In the OpenDRIVE there are many more roads than the one imported from the OSM. This is due to junctions and these roads are junction connections. MapImporter and ROD use an older version of the OpenDRIVE standard so roads inside the junction need to be treated as follows; junction connections do not have corresponding OSM roads, but they connect two OpenDRIVE roads that already have the elevation profile, so the elevation profile in these roads can be added using the elevation of the predecessor and successor to have a smooth transition. The newer version of the OpenDRIVE standard supports junction elevation as a grid. In this case for the points on the grid the elevation data can be collected similarly to BeamNG terrain implementation. When all the elevation profiles are calculated they are saved into the OpenDRIVE file with the right XML tags.

Some notable facts are:

- the implementation of elevation does not need any further implementations in ROD, VTD and ADAMS because the simulation (inside VTD or ADAMS) knows everything about the scenario environment

- all the previous knowledge gained thanks to BeamNG about OSM, spline functions and the elevation code and new knowledge gained at Hexagon about OpenDRIVE and all the Hexagon's software were used to make this implementation

5
Analysis

5.1 Research questions

The main research question is whether the test outcome changes between the native importer and the custom one and between with and without elevation. To answer this research question, the following sub-research questions were formulated:

- RQ1: Does Raw, Custom and Custom elevation mean and variance of distance from the reference line change?
  - it is important because it proves that there is difference between Raw and Custom and between Custom and Custom elevation
  - metrics: distance from the reference line
- RQ2: Does Raw, Custom and Custom elevation mean and variance of steering input change?
  - it is important because it proves that there is difference between Raw and Custom and between Custom and Custom elevation
  - metrics: steering input
- RQ3: Does Raw, Custom and Custom elevation mean and variance of driven distance 2D and 3D change?
  - it is important because it proves that there is difference between Custom and Custom elevation
  - metrics: distance 2D and 3D
- RQ4: Does Raw, Custom and Custom elevation mean and variance of speed 2D and 3D change?
  - it is important because it proves that there is difference between Custom and Custom elevation
  - metrics: speed 2D and 3D
- RQ5: Does Raw, Custom and Custom elevation mean and variance of pitch and acceleration on z axis change?
  - it is important because it proves that there is difference between Custom and Custom elevation

- metrics: pitch and acceleration on z axis

To answer these research questions statistical tests on mean and variance were carried out. From now on the native importer could be referred as Raw, the custom importer as Custom, the custom importer with elevation as Custom elevation, so these terms have the same meaning.

5.2 Road test

After checking for defects in the Road component of the BeamNGpy API for the BeamNG.tech simulator and opening the issue on the GitHub repository I proceeded to better verify what the problem was, creating road segments with an increasing number of points and checking if there were any artifacts, you can see an example in figure 5.1.

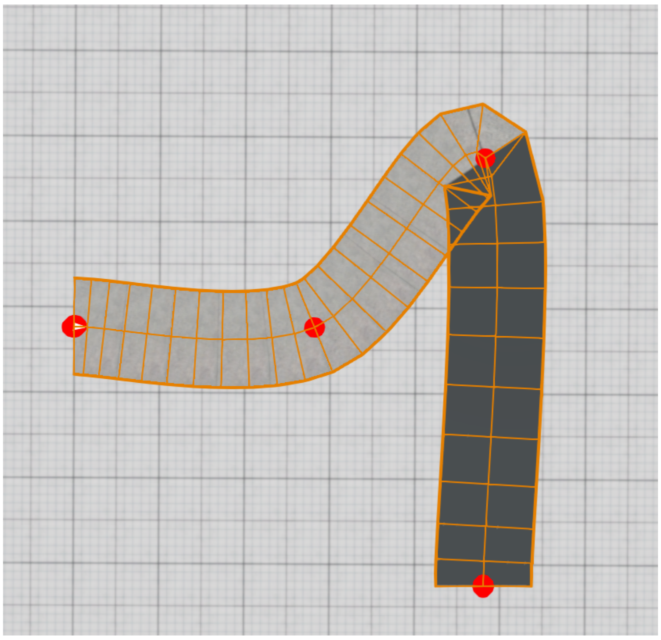


Figure 5.1: Example of artifact in the Road component

Test code available on GitHub github.com/gianlucafabris/BeamNGpy - Road test and the analysis is available in the appendices.

5.3 Statistical tests

The F test, T test, and Welch test are fundamental tools in statistical analysis, each serving distinct purposes in hypothesis testing.

The F test is primarily used to compare the variances of two populations and is integral to the Analysis of Variance when assessing the equality of multiple group means. It assumes normally distributed populations and independent samples.

The T test, available in independent and paired forms, compares the means of two populations. The independent T test is used for different groups, while the paired T test is used for the same group at different times. It assumes normality, independence, and homogeneity of variances. The Welch test, a robust alternative to the T test, is designed for comparing means when the assumption of equal variances is violated, accommodating different variances and sample sizes.

Confidence intervals provide a range of values that likely contain the population parameter with a specified confidence level, such as 95%. These intervals are calculated from sample data and indicate the reliability of an estimate. The p value, representing the probability of observing the test results under the null hypothesis, guides the decision to reject or fail to reject the null hypothesis. A p value less than or equal to 0.05 typically indicates strong evidence against the null hypothesis, warranting its rejection.

Together, these statistical methods and concepts are crucial for conducting rigorous and reliable inferential statistics.

5.4 BeamNG - test setup

After converting the OSM into drivable roads in the simulator (via the built-in converter or my converter), the scenario was created in BeamNGpy with all standard settings. A route that the ego car must follow was added via the world editor in BeamNG.tech. Then a simulation was carried out and the data were collected. Some considerations during the simulations:

- in the native importer, imported roads are not drivable
- in the custom importer with elevations, there were noticeable differences between the custom importer with and without elevation and towards the end of the simulation run of customer elevation the right tyre exploded due to the road artifact

5.5 BeamNG - analysis

After recreating the missing data and normalizing the data, the statistical tests were run, with the following results:

- here is an overview of the simulations with the positions x, y and z axes (doing a statistical test on position on z axis is trivial) (see figure 5.2 and table 5.1)

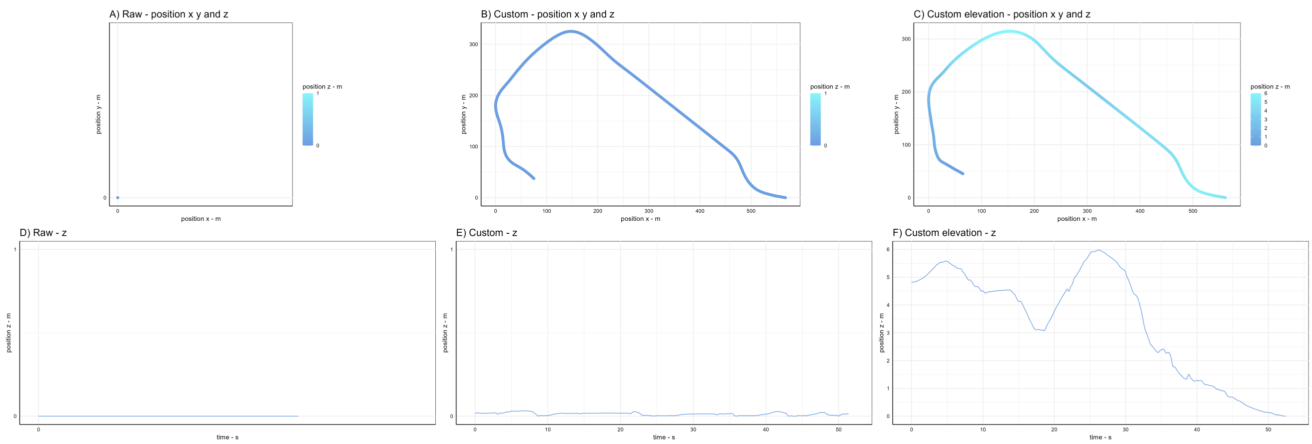


Figure 5.2: Simulation overview

	Raw	Custom	Custom elevation
time	3.0695	52.3635	53.4505
distance 2D	0.0008	931.0170	909.4712
distance 3D	0.0009	931.1020	909.7872
elevation	0.0002	0.5559	13.8175

Table 5.1: Simulation overview

- RQ1: Distance from reference line
  It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (left asymmetry and outliers), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the road generator influences the average distance form reference (also the variance) (see figure 5.3and tables 5.2 and 5.3).

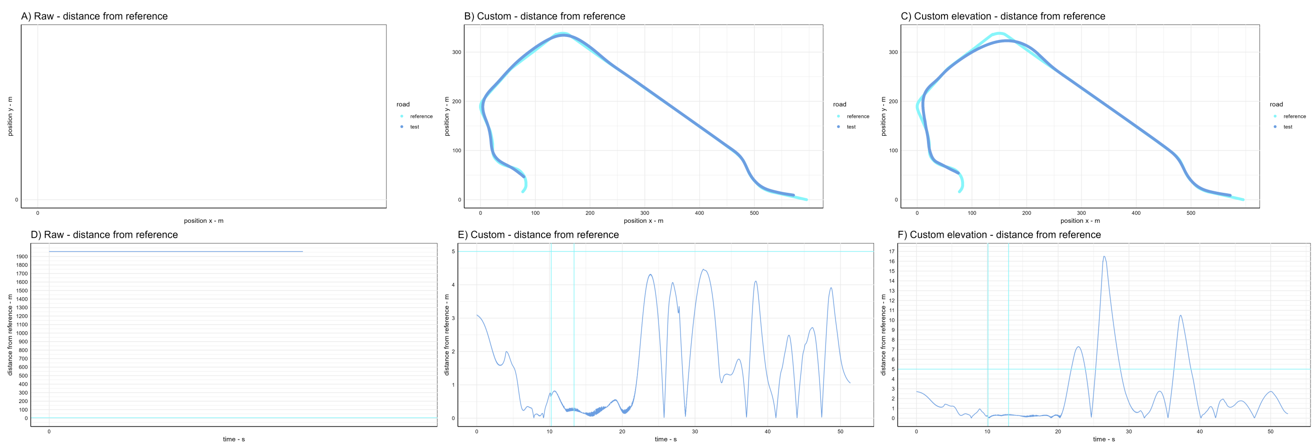


Figure 5.3: Distance from the reference line

	Raw	Custom	Custom elevation
min	1958	0.0070	0.0129
1st qu	1958	0.3917	0.3428
median	1958	1.4223	1.2213
mean	1958	1.6474	2.4659
3rd qu	1958	2.5922	2.6439
max	1958	4.4675	16.5105
sd	0.019e-04	1.3033	3.2994
skewness	-0.2934	0.5700	2.2125
kurtosis	2.3838	2.1364	7.7978

Table 5.2: Distance from the reference line

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	2.019e-12	0.1561
F test - p-value	$<$ 2.200e-16	$<$ 2.200e-16
F test - 99% confidence int.	[1.529e-12, 2.763e-12]	[0.1456, 0.1672]
T/Welch test - t_{oss}	113115.0000	-16.9980
T/Welch test - p-value	$<$ 2.200e-16	$<$ 2.200e-16
T/Welch test - 99% confidence int.	[1956.6190, 1956.7080]	[-0.9425, -0.6944]

Table 5.3: Distance from the reference line - test

RQ1: mean (Welch test) and variance (F test) of distance from the reference line: both with the statistical tests and visually, there is evidence that the road generator (both between native and custom and between with and without elevation) influences the mean and variance of distance from the reference line

- RQ2: Steering
  It is observable that native simulation run failed and the test between native and custom without elevation will not be run, while for custom with and without elevation it is observable that the populations are not normal (right asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis

that the variances are equal. From the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the road generator influences the average steering (also the variance) (see figure 5.4 and tables 5.4 and 5.5).

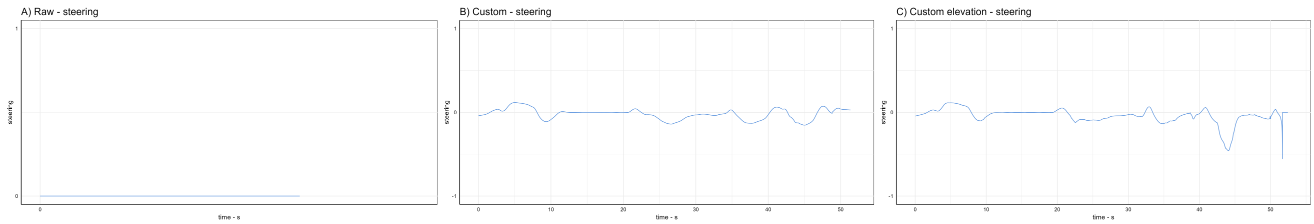


Figure 5.4: Steering

	Raw	Custom	Custom elevation
min	0	-0.1533	-0.5542
1st qu	0	-0.0623	-0.0740
median	0	-0.0049	-0.0301
mean	0	-0.0211	-0.0375
3rd qu	0	0.0200	-0.0007
max	0	0.1168	0.1135
sd	0	0.0637	0.0832
skewness	NaN	-0.1875	-2.013
kurtosis	NaN	2.5126	10.7870

Table 5.4: Steering

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	NA	0.5867
F test - p-value	NA	< 2.200e-16
F test - 99% confidence int.	NA	[0.5474, 0.6288]
T/Welch test - t_{oss}	NA	11.6270
T/Welch test - p-value	NA	< 2.200e-16
T/Welch test - 99% confidence int.	NA	[0.0128, 0.0201]

Table 5.5: Steering - test

RQ2: mean (Welch test) and variance (F test) of steering input: the difference between native and custom importer is trivial; both with the statistical tests and visually, there is evidence that the road generator (between with and without elevation) influences the mean and variance of steering input

- RQ3: Distance 2D
  It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (right asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the custom road generator influences the average distance 2D (also the variance), but the elevation doesn't influence the distance 2D see figure 5.5 and tables 5.6 and 5.7).

  RQ3: Distance 3D
  It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (right asymmetry),

thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the custom road generator influences the average distance 3D (also the variance), but the elevation doesn't influence the distance 3D (see figure 5.5 and tables 5.8 and 5.9).

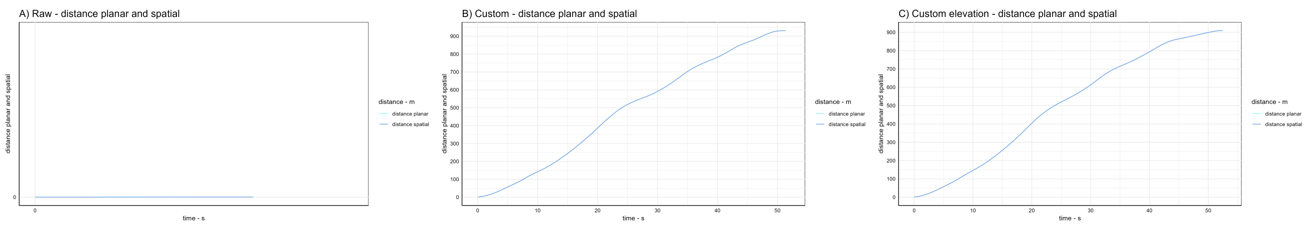


Figure 5.5: Distance 2D and 3D

	Raw	Custom	Custom elevation
min	0.0003	1.0560	1.1290
1st qu	0.0004	205.6160	219.2720
median	0.0004	536.7030	537.9060
mean	0.0005	495.0930	499.0780
3rd qu	0.0007	766.5900	765.1940
max	0.0008	931.0170	909.4710
sd	0.0002	300.0704	296.9502
skewness	-0.1161	-0.1640	-0.2310
kurtosis	1.7505	1.6784	1.6812

Table 5.6: Distance 2D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.296e-12	1.0211
F test - p-value	$<$ 2.200e-16	0.4372
F test - 99% confidence int.	[0.224e-12 0.405e-12]	[0.9527 1.0944]
T/Welch test - t_{oss}	-124.3100	-0.7019
T/Welch test - p-value	$<$ 2.200e-16	0.4828
T/Welch test - 99% confidence int.	[-505.3547, -484.8309]	[-18.6120, 10.6421]

Table 5.7: Distance 2D - test

	Raw	Custom	Custom elevation
min	0.0003	1.1400	1.2780
1st qu	0.0004	205.7000	219.4380
median	0.0006	536.7900	538.1150
mean	0.0006	495.1800	499.3030
3rd qu	0.0007	766.6700	765.4880
max	0.0009	931.1000	909.7870
sd	0.0002	300.0705	297.0076
skewness	-0.1167	-0.1640	-0.2308
kurtosis	1.7581	1.6784	1.6812

Table 5.8: Distance 3D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.32313e-12	1.0207
F test - p-value	< 2.200e-16	0.4457
F test - 99% confidence int.	[0.2447e-12, 0.4422e-12]	[0.9524, 1.0939]
T/Welch test - t_{oss}	-124.3400	-0.7264
T/Welch test - p-value	< 2.200e-16	0.4676
T/Welch test - 99% confidence int.	[-505.4395, -484.9157]	[-18.7531, 10.5037]

Table 5.9: Distance 3D - test

RQ3: mean (Welch test between native and custom and T test between with and without elevation) and variance (F test) of driven distance 2D and 3D: both with the statistical tests and visually, there is evidence that the road generator (between native and custom) influences the mean and variance of driven distance (both 2D and 3D), while there is not evidence that the road generator (between with and without elevation) influences the mean and variance of driven distance (both 2D and 3D). The fact that it does not influence the distance 2D is the expected result, while the fact that it does not influence the distance 3D is not the expected result. This is due to the road slope: the test location is about 5 meters of elevation across about 800 meters (2D) result in 800.01 meters (3D) - average slope 0.36°, even Pikes Peak (a steep hill climb race) has an elevation change of about 2,4km across about 20km (2D) results in 20.14km (3D) - average slope 6.84°, these differences are minimal and therefore cannot be significant

- RQ4: Speed 2D

It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the road generator influences the average speed 2D (also the variance) (see figure 5.6 and tables 5.10 and 5.11).

RQ4: Speed 3D

It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (asymmetry and outliers), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the road generator influences the average speed 3D (also the variance) (see figure 5.6 and tables 5.12 and 5.13).

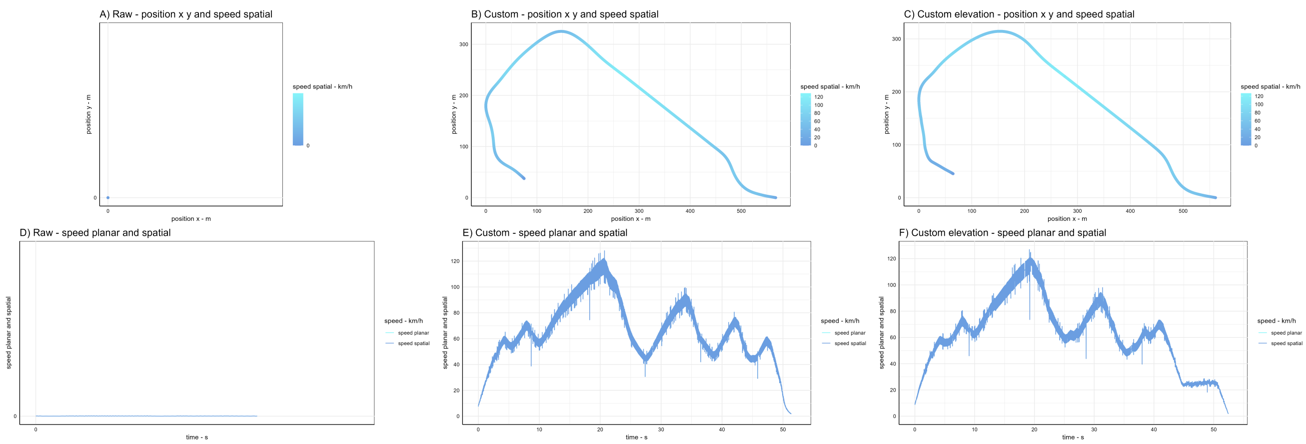


Figure 5.6: Speed 2D and 3D

	Raw	Custom	Custom elevation
min	0.118e-04	0.5007	0.5103
1st qu	1.128e-04	14.1104	13.8545
median	2.397e-04	17.0276	17.5952
mean	2.604e-04	18.0285	17.5020
3rd qu	4.052e-04	22.4251	22.2138
max	6.297e-04	35.5701	35.2568
sd	0.0002	6.7181	7.2405
skewness	0.4242	0.0164	-0.0557
kurtosis	2.1002	3.1030	2.5422

Table 5.10: Speed 2D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	603.320e-12	0.8609
F test - p-value	$<$ 2.200e-16	2.552e-08
F test - 99% confidence int.	[456.881e-12 825.724e-12]	[0.8032, 0.9226]
T/Welch test - t_{oss}	-202.1900	3.9603
T/Welch test - p-value	$<$ 2.200e-16	0.753e-04
T/Welch test - 99% confidence int.	[-18.2580, -17.7985]	[0.1840, 0.8690]

Table 5.11: Speed 2D - test

	Raw	Custom	Custom elevation
min	0.167e-04	0.5007	0.5162
1st qu	1.259e-04	14.1104	13.8549
median	2.483e-04	17.0276	17.5984
mean	2.729e-04	18.0285	17.5053
3rd qu	4.144e-04	22.4251	22.2142
max	6.488e-04	35.5701	35.2604
sd	0.0002	6.7181	7.2415
skewness	0.4532	0.0164	-0.0560
kurtosis	2.1402	3.1030	2.5418

Table 5.12: Speed 3D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	578.540e-12	0.8607
F test - p-value	$<$ 2.200e-16	2.411e-08
F test - 99% confidence int.	[438.117e-12, 791.812e-12]	[0.8030, 0.9224]
T/Welch test - t_{oss}	-202.1900	3.9352
T/Welch test - p-value	$<$ 2.200e-16	0.836e-04
T/Welch test - 99% confidence int.	[-18.2579, -17.7985]	[0.1807, 0.8657]

Table 5.13: Speed 3D - test

RQ4: mean (Welch test) and variance (F test) of speed 2D and 3D: both with the statistical tests and visually, there is evidence that the road generator (both between native and custom and between with and without elevation) influences the mean and variance of speed (both 2D and 3D). Here we can apply the same consideration as above about the difference between 2D and 3D, but the statistical test rejected the hypothesis that the means are equal (proving that, although is not visible, there is difference in the speed 3D between with and without elevation)

- RQ5: Pitch
  It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (asymmetry and outliers), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the road generator influences the average pitch (also the variance) (see figure 5.7 and tables 5.14 and 5.15).

RQ5: Acceleration z

It is observable that native simulation run failed and the population is not normal, while for custom with and without elevation it is observable that the populations are not normal (asymmetry and outliers), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the road generator doesn't influence the average z acceleration, but it influences the variance (see figure 5.8 and tables 5.16 and 5.17).

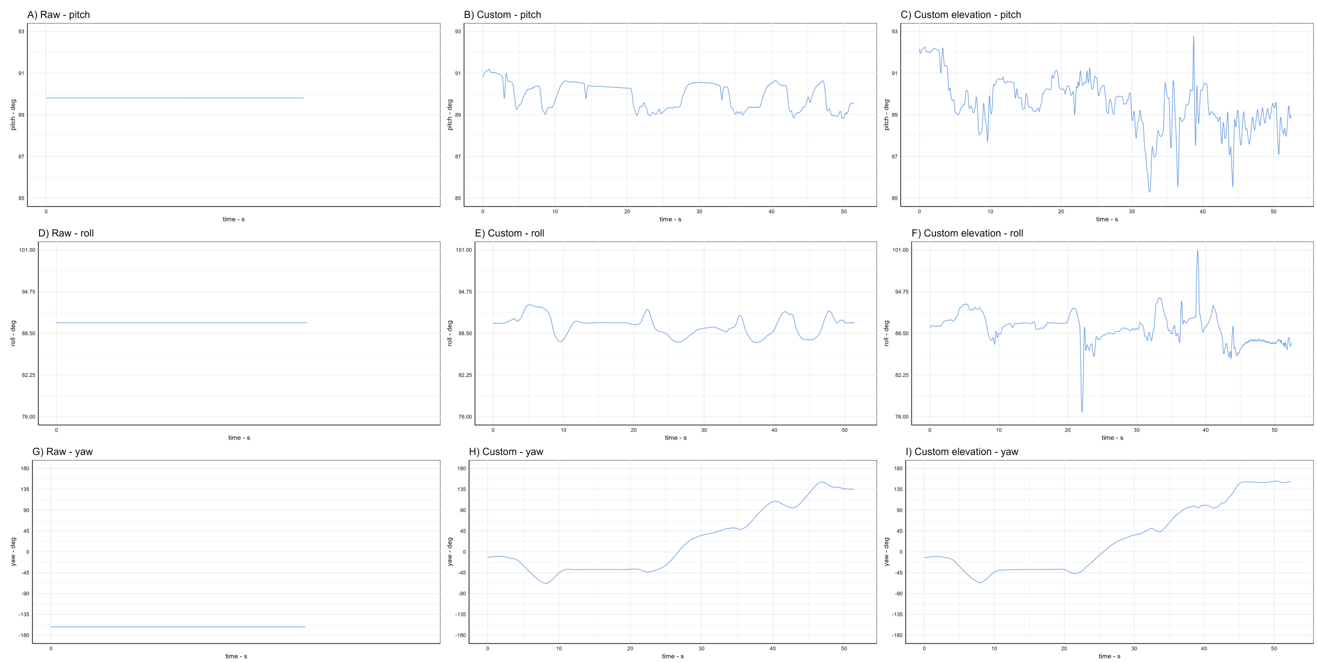


Figure 5.7: Gyroscope

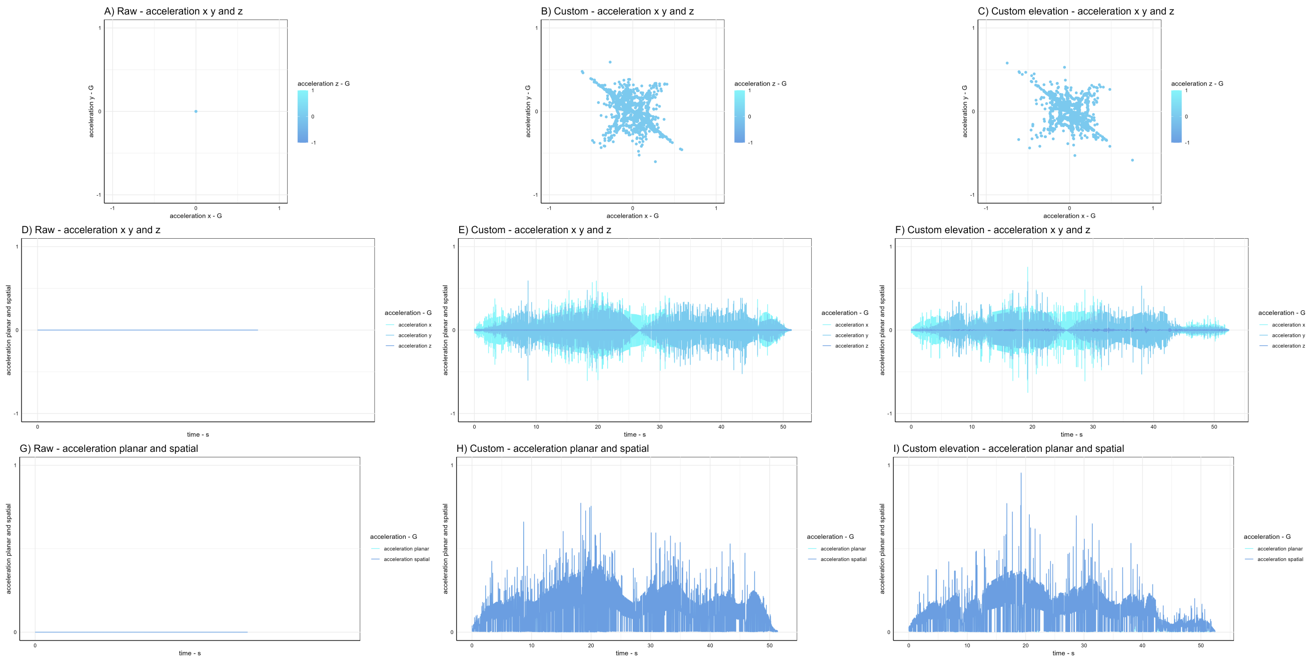


Figure 5.8: Accelerometer

	Raw	Custom	Custom elevation
min	-0.0033	-0.0207	-0.0822
1st qu	-0.0033	-0.0116	-0.0207
median	-0.0033	0.0029	-0.0089
mean	-0.0033	-0.0013	-0.0084
3rd qu	-0.0033	0.0080	0.0053
max	-0.0033	0.0208	0.0483
sd	0.036e-04	0.0110	0.0221
skewness	-0.1418	-0.1167	-0.2895
kurtosis	2.5013	1.6195	3.7163

Table 5.14: Pitch

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	10.424e-08	0.2486
F test - p-value	< 2.200e-16	< 2.200e-16
F test - 99% confidence int.	[7.894e-08 14.267e-08]	[0.2320, 0.2664]
T/Welch test - t_{oss}	-14.1820	21.3690
T/Welch test - p-value	< 2.200e-16	< 2.200e-16
T/Welch test - 99% confidence int.	[-0.0024, -0.0017]	[0.0063, 0.0080]

Table 5.15: Pitch - test

	Raw	Custom	Custom elevation
min	-1.882e-04	-318.200e-04	-0.3422
1st qu	-0.342e-04	-12.210e-04	-0.0157
median	-0.030e-04	-0.110e-04	-0.0002
mean	0.001e-04	-0.087e-04	-0.0001
3rd qu	0.358e-04	12.250e-04	0.0152
max	1.954e-04	349.800e-04	0.3765
sd	0.684e-04	0.0040	0.0387
skewness	0.0350	-0.1290	0.3737
kurtosis	3.5405	19.9225	13.1447

Table 5.16: Acceleration z

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.0003	0.0105
F test - p-value	< 2.200e-16	< 2.200e-16
F test - 99% confidence int.	[0.0002, 0.0004]	[0.0098, 0.0112]
T/Welch test - t_{oss}	0.1668	0.1990
T/Welch test - p-value	0.8675	0.8423
T/Welch test - 99% confidence int.	[-0.0001, 0.0001]	[-0.0013, 0.0015]

Table 5.17: Acceleration z - test

RQ5: mean (Welch test) and variance (F test) of pitch and acceleration on z axis: both with the statistical tests and visually, there is evidence that the road generator (both between native and custom and between with and without elevation) influences the mean and variance of pitch and acceleration on z axis, proving that there is a difference with and without elevation

Note that these test were run on an older version of the custom osm importer, which imported the road using both `Road` and `MeshRoad`. Thanks to the issue on the GitHub repository was found that the `MeshRoad` caused the tyre to explode and was also found that caused an issue in the navgraph of road drivability (affects the internal agent, does not affect DAVE-2). So in the newer version of the custom osm importer the road are imported using only `Road`, the result is a little bit less accurate but it is drivable. Due to time constraints the test on custom elevation was not re-run.

The analysis details and the code can be retrieved on GitHub github.com/gianlucafabris/Import OSM custom - analysis BeamNG and Hexagon.

5.6 Hexagon - test setup

After converting the OSM into OpenDRIVE (via the built-in converter or the MapImporter converter), a database was created in ROD, then, the database and the OpenDRIVE files were copied into VTD and the scenario was created in VTD with all standard settings and a route that the ego car must follow was added. Then a simulation was run and the data were collected. Some considerations during the simulations:

- everything went smoothly during the simulations. The ego car seemed slightly too adherent to the road. This might be due to the simpler dynamics model used (a more complex model like ADAMS was not used) or might be thanks to the more rigidly defined road

- in the custom importer with elevations, there were noticeable differences with the custom importer without elevation

5.7 Hexagon - analysis

After recreating the missing data and normalizing the data, the statistical tests were run, with the following results:

- here is an overview of the simulations with the positions x, y and z axes (doing a statistical test on position on z axis is trivial) (see figure 5.9 and table 5.18)

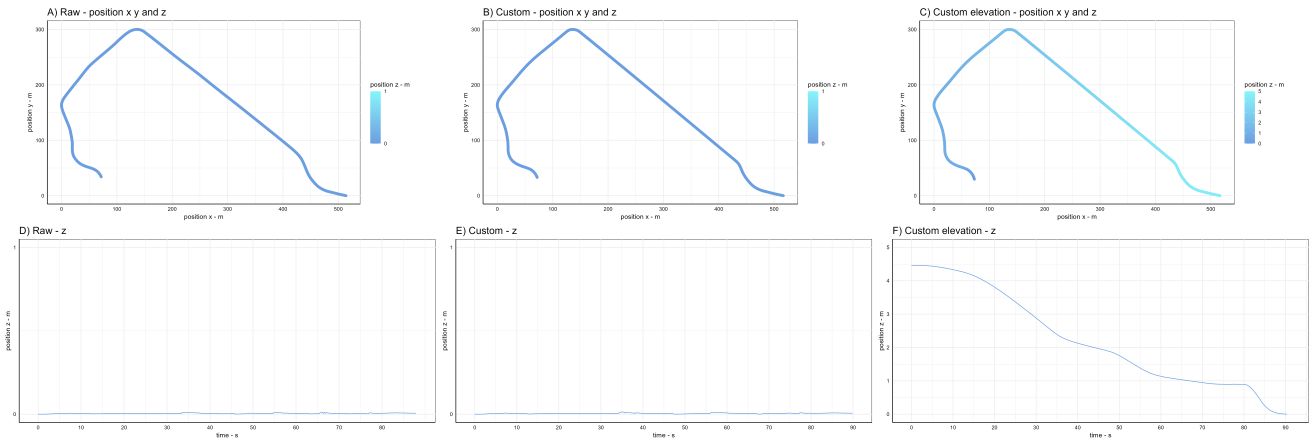


Figure 5.9: Simulation overview

	Raw	Custom	Custom elevation
time	88.9000	90.8500	91.3500
distance 2D	850.0714	850.4499	854.4989
distance 3D	850.0716	850.4499	854.5186
elevation	0.1393	0.1169	4.4806

Table 5.18: Simulation overview

- RQ1: Distance from reference line
  It is observable that for native, custom with and without elevation simulations runs the populations are not normal (right asymmetry and outliers), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99%

confidence the two means can be considered different and also the high p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the custom road generator influences the variance distance form reference, but the elevation doesn't influence the variance; the road generator doesn't influence the mean distance form reference (see figure 5.10 and tables 5.19 and 5.20).

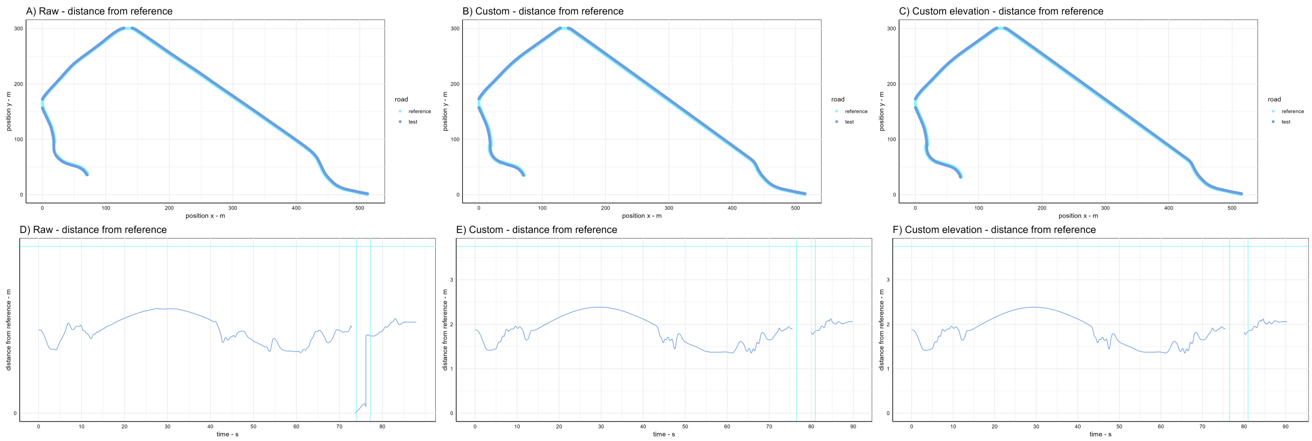


Figure 5.10: Distance from the reference line

	Raw	Custom	Custom elevation
min	-0.0507	-1.9220	-1.9220
1st qu	1.5384	1.5420	1.5450
median	1.7248	1.8610	1.8610
mean	1.6960	1.6850	1.6880
3rd qu	1.9479	2.0890	2.0870
max	2.1935	2.3840	2.3840
sd	0.4110	0.8472	0.8455
skewness	-2.0822	-3.2541	-3.2639
kurtosis	9.3866	13.7360	13.8098

Table 5.19: Distance from the reference line

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.2353	1.0041
F test - p-value	< 2.200e-16	0.8817
F test - 99% confidence int.	[0.2193, 0.2525]	[0.9361, 1.0769]
T/Welch test - t_{oss}	0.83102	-0.1533
T/Welch test - p-value	0.4060	0.8782
T/Welch test - 99% confidence int.	[-0.0224, 0.0438]	[-0.0444, 0.0394]

Table 5.20: Distance from the reference line - test

RQ1: mean (Welch test between native and custom and T test between with and without elevation) and variance (F test) of distance from the reference line: both with the statistical tests and visually, there is evidence that the road generator (both between native and custom and between with and without elevation) does not influence the mean, while variance of distance from the reference line is only influenced between native and custom, while between with and without elevation the variance is not influenced, this is not the expected result. This might be due to the simpler dynamics model or thanks to the more rigidly defined road. We are confident that using the more complex (ADAMS) test will be more significant and give more positive results

- RQ2: Steering
  It is observable that for native, custom with and without elevation simulations runs the populations are not normal (right asymmetry and outliers), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to reject with reservation the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the custom road generator influences the mean steering, but the elevation doesn't influence the

steering (see figure 5.11 and tables 5.21 and 5.22).

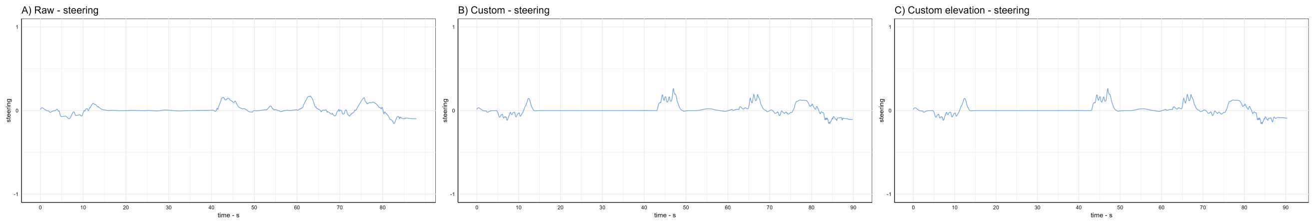


Figure 5.11: Steering

	Raw	Custom	Custom elevation
min	-0.1548	-0.1556	-0.1596
1st qu	-0.0064	-0.0049	-0.0052
median	0.0012	0.0001	0.0001
mean	0.0065	0.0087	0.0084
3rd qu	0.0191	0.0161	0.0157
max	0.1741	0.2645	0.2645
sd	0.0562	0.0609	0.0609
skewness	0.4817	0.8059	0.8212
kurtosis	4.0718	4.7765	4.7820

Table 5.21: Steering

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.8514	1.0021
F test - p-value	0.441e-08	0.9398
F test - 99% confidence int.	[0.7934, 0.9137]	[0.9342, 1.0748]
T/Welch test - t_{oss}	-1.9889	0.2844
T/Welch test - p-value	0.0467	0.7761
T/Welch test - 99% confidence int.	[-0.0052, 0.0007]	[-0.0027, 0.0034]

Table 5.22: Steering - test

RQ2: mean (Welch test) and variance (F test) of steering input: both with the statistical tests and visually, there is evidence that the road generator (between native and custom) influences the mean and variance of steering input, while between with and without elevation both mean and variance are not influenced. This is not the expected result and might be due to the simpler dynamics model or thanks to the more rigidly defined road. We are confident that using the more complex (ADAMS) test will be more significant and give more positive results

- RQ3: Ditance 2D
  It is observable that for native, custom with and without elevation simulations runs the populations are not normal (right asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the T test between native and custom without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the road generator doesn't influence the average distance 2D (also the variance) (see figure 5.12 and tables 5.23 and 5.24).

  RQ3: Ditance 3D
  It is observable that for native, custom with and without elevation simulations runs the populations are not normal (right asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the high p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the T test between native and custom without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered equal and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the road generator doesn't influence the average distance 3D (also the variance) (see figure 5.12 and tables 5.25 and 5.26).

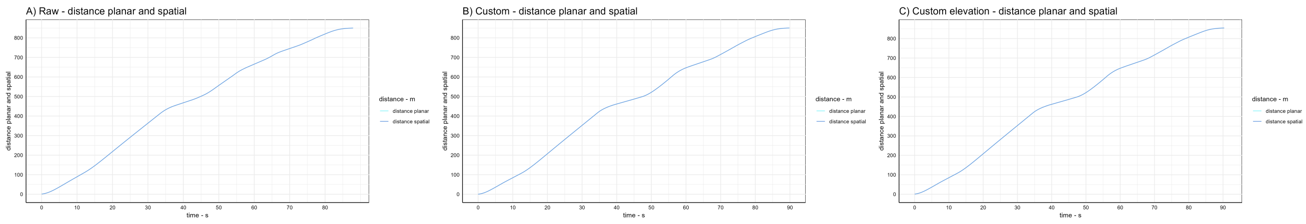


Figure 5.12: Distance 2D and 3D

	Raw	Custom	Custom elevation
min	0.8994	0.0746	0.7462
1st qu	247.2329	243.3074	245.8072
median	492.9694	486.4288	488.3912
mean	475.3516	469.0384	471.9688
3rd qu	717.0656	692.3073	695.6641
max	850.0714	850.4499	854.4989
sd	264.2784	262.2226	263.2352
skewness	-0.2700	-0.2465	-0.2477
kurtosis	1.8059	1.8305	1.8317

Table 5.23: Distance 2D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	1.0157	0.9923
F test - p-value	0.5685	0.7769
F test - 99% confidence int.	[0.9466, 1.0900]	[0.9252, 1.0644]
T/Welch test - t_{oss}	1.2384	-0.5799
T/Welch test - p-value	0.2156	0.5620
T/Welch test - 99% confidence int.	[-6.8204, 19.4468]	[-15.9483, 10.0876]

Table 5.24: Distance 2D - test

	Raw	Custom	Custom elevation
min	0.8994	0.0746	0.7462
1st qu	247.2329	243.3074	245.8091
median	492.9694	486.4289	488.3987
mean	475.3517	469.0385	471.9756
3rd qu	717.0657	692.3074	695.6740
max	850.0716	850.4499	854.5186
sd	264.2785	262.2226	263.2399
skewness	-0.2700	-0.2465	-0.2477
kurtosis	1.8059	1.8305	1.8317

Table 5.25: Distance 3D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	1.0157	0.9923
F test - p-value	0.5684	0.7759
F test - 99% confidence int.	[0.9466, 1.0900]	[0.9251, 1.0643]
T/Welch test - t_{oss}	1.2384	-0.5813
T/Welch test - p-value	0.2156	0.5611
T/Welch test - 99% confidence int.	[-6.8204, 19.4468]	[-15.9552, 10.0810]

Table 5.26: Distance 3D - test

RQ3: mean (T test) and variance (F test) of driven distance 2D and 3D: both with the statistical tests and visually, there is evidence that the road generator (both between native and custom and between with and without elevation) does not influence the mean and variance of driven distance (both 2D and 3D). This is not the expected result and it is due to the road slope: the test location is about 5 meters of elevation across about 800 meters (2D) result in 800.01 meters (3D) - average slope 0.36°, even the Pikes Peak (a steep hill climb race) has an elevation change of about 2,4km across about 20km (2D) results in 20.14km (3D) - average slope 6.84°, these differences are minimal and therefore cannot be significant

- RQ4: Speed 2D

  It is observable that for native, custom with and without elevation simulations runs the populations are not normal (left asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the custom road generator influences the average speed 2D (also the variance), but the elevation doesn't influence the speed 2D (see figure 5.13 and tables 5.27 and 5.28).

RQ4: Speed 3D

It is observable that for native, custom with and without elevation simulations runs the populations are not normal (left asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the T test between custom with and without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to accept the hypothesis that the means are equal. So the custom road generator influences the average speed 3D (also the variance), but the elevation doesn't influence the speed 3D (see figure 5.13 and tables 5.29 and 5.26).

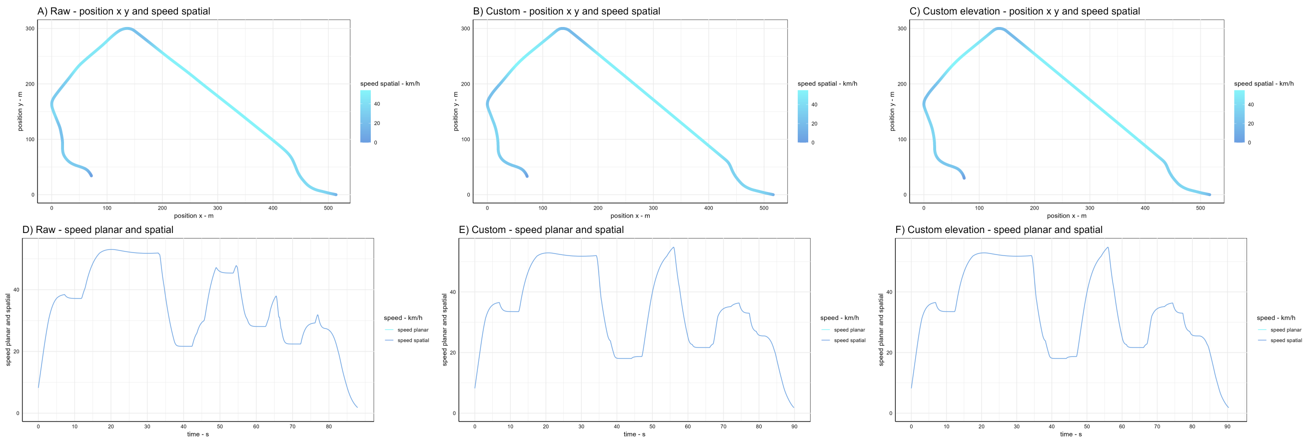


Figure 5.13: Speed 2D and 3D

	Raw	Custom	Custom elevation
min	0.5041	0.5045	0.5027
1st qu	7.1946	6.2695	6.2858
median	9.4545	9.3265	9.3204
mean	9.6591	9.4629	9.4479
3rd qu	12.8106	13.7523	13.7242
max	14.7306	15.2125	15.2125
sd	3.5550	3.7825	3.7795
skewness	-0.2227	-0.0830	-0.0773
kurtosis	2.3694	2.0406	2.0486

Table 5.27: Speed 2D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.8833	1.0016
F test - p-value	599.200e-08	0.9535
F test - 99% confidence int.	[0.8232, 0.9479]	[0.9338, 1.0743]
T/Welch test - t_{oss}	2.7612	0.2061
T/Welch test - p-value	0.0059	0.8368
T/Welch test - 99% confidence int.	[9.6591, 9.4629]	[-0.1724, 0.2023]

Table 5.28: Speed 2D - test

	Raw	Custom	Custom elevation
min	0.5041	0.5045	0.5027
1st qu	7.1946	6.2696	6.2874
median	9.4545	9.3266	9.3204
mean	9.6591	9.4629	9.4481
3rd qu	12.8106	13.7524	13.7244
max	14.7306	15.2125	15.2127
sd	3.5550	3.7825	3.7795
skewness	-0.2227	-0.0830	-0.0773
kurtosis	2.3694	2.0406	2.0487

Table 5.29: Speed 3D

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.8833	1.0016
F test - p-value	599.100e-08	0.9533
F test - 99% confidence int.	[0.8232, 0.9479]	[0.9338, 1.0743]
T/Welch test - t_{oss}	2.7611	0.2031
T/Welch test - p-value	0.0058	0.8391
T/Welch test - 99% confidence int.	[0.0131, 0.3793]	[-0.1726, 0.2021]

Table 5.30: Speed 3D - test

RQ4: mean (Welch test between native and custom and T test between with and without elevation) and variance (F test) of speed 2D and 3D: both with the statistical tests and visually, there is evidence that the road generator (between native and custom) influences the mean and variance of speed (both 2D and 3D), while between with and without elevation both mean and variance are not influenced. This is not the expected result and might be due to the simpler dynamics model or thanks to the more rigidly defined road. We are confident that using the more complex (ADAMS) test will be more significant and give more positive results

- RQ5: Pitch

  It is observable that for native, custom with and without elevation simulations runs the populations are not normal (left asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered different and also the high p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the custom road generator doesn't influence the average pitch, but the elevation influences the average pitch, the road generator influences the variance pitch (see figure 5.14 and tables 5.31 and 5.32).

RQ5: Pitch

It is observable that for native, custom with and without elevation simulations runs the populations are not normal (left asymmetry), thanks to high number of data in the populations (over 5000 entries) it is possible to assume them to be normal. From the f_{oss} of the F test between native and custom without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal, while from the f_{oss} of the F test between custom with and without elevation, with 99% confidence the two variances can be considered different and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the variances are equal. From the t_{oss} of the Welch test between native and custom without elevation, with 99% confidence the two means can be considered equal and also the low p-value reinforces the decision to decisively reject the hypothesis that the means are equal, while from the t_{oss} of the Welch test between custom with and without elevation, with 99% confidence the two means can be considered equal and also the extremely low p-value reinforces the decision to decisively reject the hypothesis that the means are equal. So the road generator influences the average z acceleration (also the variance) (see figure 5.15 and tables 5.33 and 5.34).

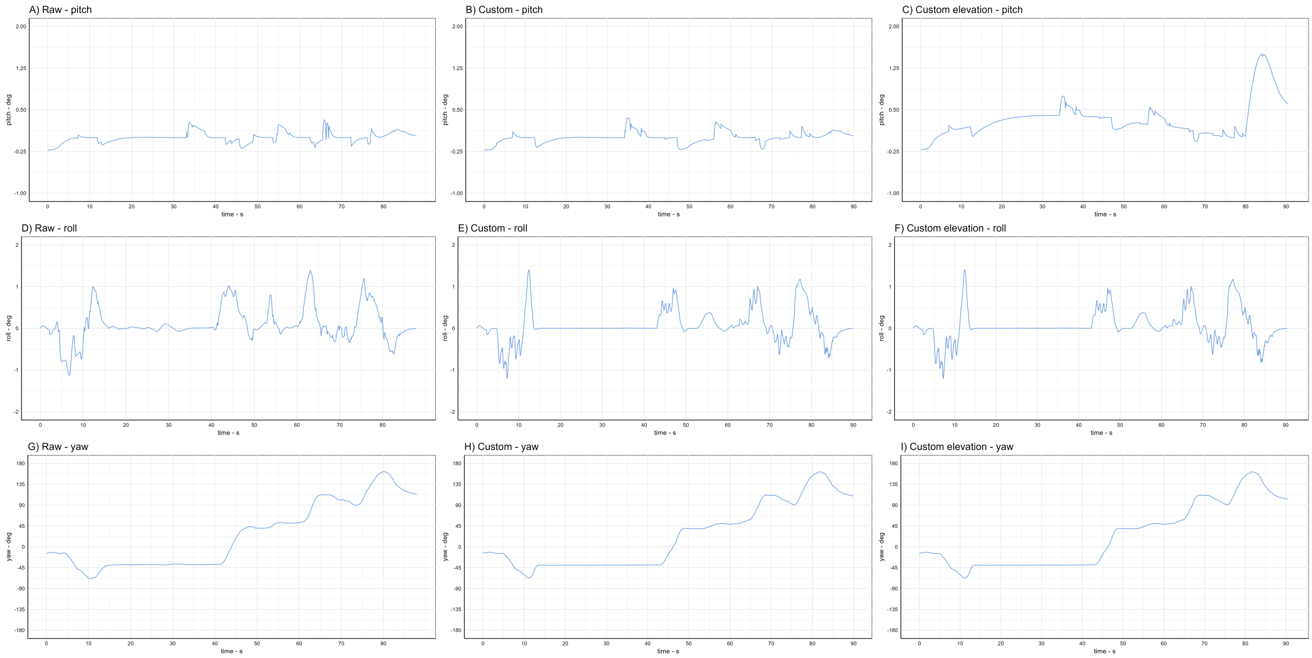


Figure 5.14: Gyroscope

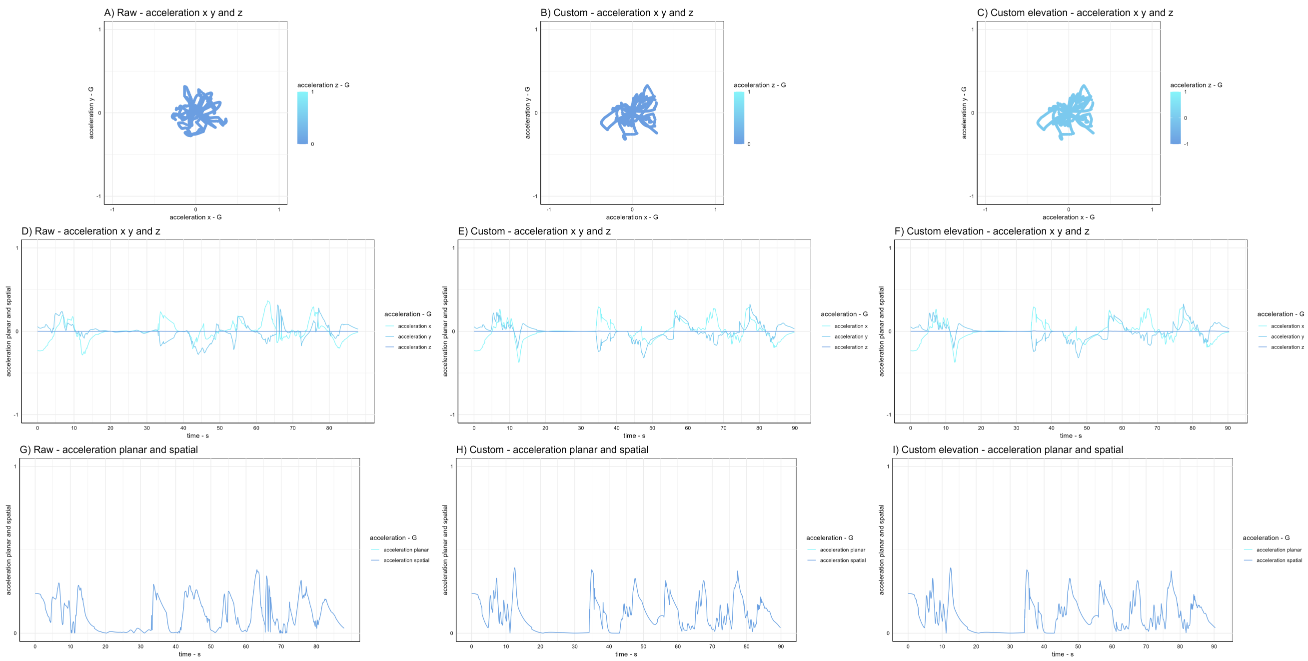


Figure 5.15: Accelerometer

	Raw	Custom	Custom elevation
min	-38.810e-04	-38.810e-04	-0.0038
1st qu	-6.253e-04	-4.631e-04	0.0023
median	0.058e-04	0.004e-04	0.0046
mean	0.332e-04	0.325e-04	0.0057
3rd qu	3.896e-04	4.378e-04	0.0069
max	55.980e-04	62.080e-04	0.0263
sd	0.0016	0.0017	0.0057
skewness	0.3661	0.2912	1.7209
kurtosis	4.2331	4.6652	6.5497

Table 5.31: Pitch

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.8747	0.0870
F test - p-value	103.800e-08	$<$ 2.200e-16
F test - 99% confidence int.	[0.8151, 0.9387]	[0.0811, 0.0933]
T/Welch test - t_{oss}	0.0229	-69.8660
T/Welch test - p-value	0.9817	$<$ 2.200e-16
T/Welch test - 99% confidence int.	[-0.804e-04 0.818e-04]	[-0.0058, -0.0054]

Table 5.32: Pitch - test

	Raw	Custom	Custom elevation
min	0	0	-59.960e-04
1st qu	0.014e-04	0.016e-04	-1.921e-04
median	1.402e-04	1.226e-04	0.225e-04
mean	14.840e-04	16.970e-04	31.550e-04
3rd qu	17.500e-04	18.510e-04	22.940e-04
max	188.500e-04	231.800e-04	488.900e-04
sd	0.0027	0.0033	0.0084
skewness	2.5893	2.9954	2.5756
kurtosis	10.6360	14.7426	10.3245

Table 5.33: Acceleration z

	Raw and Custom	Custom and Custom elev.
F test - f_{oss}	0.6773	0.1493
F test - p-value	$<$ 2.200e-16	$<$ 2.200e-16
F test - 99% confidence int.	[0.6312, 0.7268]	[0.1392, 0.1601]
T/Welch test - t_{oss}	-3.6814	-11.8730
T/Welch test - p-value	0.0002	$<$ 2.200e-16
T/Welch test - 99% confidence int.	[-3.612e-04, -0.638e-04]	[0.0017, 0.0032]

Table 5.34: Acceleration z - test

RQ5: mean (Welch test) and variance (F test) of pitch and acceleration on z axis: both with the statistical tests and visually, there is evidence that the road generator between native and custom does not influence the mean, while it does influence the variance of pitch. In the case of the road generator between with and without elevation there is evidence that it influences both mean and variance of pitch; there is also evidence that the road generator (both between native and custom and between with and without elevation) influences the mean and variance of acceleration on z axis, proving that there is a difference with and without elevation

Although the simulation runs were performed in the most similar situations, as stated before, the importers of the two simulators, the settings of simulation runs and the simulators are different, so the results are different and not comparable and the scope of the thesis is not to compare the simulator, but it is to create new test cases for the simulator and analyze them. The analysis details and the code can be retrieved on GitHub github.com/gianlucafabris/Import OSM custom - analysis BeamNG and Hexagon.

6

Conclusions and future developments

6.1 Conclusions

A custom OSM importer which offers the possibility to filter the roads and add the elevation, has been developed for BeamNG.tech's simulator and SBFT CPS testing competition. The elevation on roads and junctions, as a post processing of a pipeline that already creates a well defined road, has been implemented for Hexagon's simulator.

It has been proven statistically that the elevation changes the outcome of the test (already on a very simple road BeamNG.tech's ego car in the test without elevation stayed on the road, while in the test with elevation, the ego car went off the road). As stated above, this was not observed the same in Hexagon's ego car, probably because of the simpler dynamics model. We are confident that using the more complex (ADAMS) test will be more significant and give more positive results.

6.2 Future developments

For the SBFT CPS testing competition code pipeline, the standalone code needs to be adapted, there are the key points:

- to expand the infrastructure to be able to have road networks (instead of just one road) and to have the elevation information (on terrain)
- to add a selection for the starting and ending point of the simulation, then use an algorithm like Dijkstra or A^* to find the path, and add checkpoints on the path

A notable fact: due to the implementation of elevation, DAVE-2 needs to be retrained (it has been trained on flat roads), once it is retrained and the code is adapted to the SBFT CPS testing competition code pipeline, it will be possible to use the road generator to generate some test for BeamNG.tech's simulator and SBFT CPS testing competition. Here are some possible examples that might undermine the autonomous driving agent:

- Nürburgring
- Isle of Man TT

- Pikes Peak
- various rally stages
- various mountain passes

since every road in the world can be used, the possibility are endless.

For Hexagon's simulator, in some cases, for example, a main road that is on a slope with many junctions with secondary roads, the main road is split into pieces, and each piece is then shortened to make space for the junctions. The elevation sampling is based on the original length, which means that the shortened road will have more slope and the junction will be almost flat. To improve this situation the starting and ending points can be interpolated to have less slope in the road between junctions and have more slope inside the junctions. It is also possible to use more points to calculate the elevation function more in detail (see figure 4.9). It is possible to use the civil engineering elevation function to make the code better suit to the MapImporter code pipeline, although the cubic function already satisfies the civil engineering rule, and it is also possible to use Italian civil engineering rules. Thanks to the implementation of elevation inside MapImporter, now, it is now possible to implement bridges and tunnels. Some other implementations that can be added to MapImporter are motorway slip lanes and roundabouts.

I

Appendices

A

BeamNG.tech analysis

A.1 Road test

From an initial analysis using the world editor present in the simulator, it was possible to verify the following results:

- the road appears to be modeled by a spline Catmull-Rom uniform, as it follows its characteristics
- the following equivalence classes were verified:
  - equivalence by symmetry: vertically symmetric streets are in the same equivalence class
  - recursive equivalence: it applies only if the first two segments are aligned, then the validity of the road is as in the previous step; this always derives from the definition of the Catmull-Rom spline
- beyond 5 points the road behaves like a 5 point road (for uniform Catmull-Rom spline properties)
- furthermore, other equivalence classes are created to reduce the search space:
  - segment length: 1 to 250 with 10 exponential increments
  - angle between two segments:
    * 0° to 180° in 10° linear increments for the first segment (symmetry)
    * 0° to 360° in 45° linear increments for subsequent segments (4 and 5 point roads only)
  - for roads with 4 and 5 points the search space is further reduced due to the complexity and exponential growth of the test cases. For example, the test of roads made up of 5 points performed with the search space of 3 point roads would have produced more than 2.2 billion test cases; instead with the reduced search space about 2600 test cases were produced.

Subsequently, the data produced by the tests were analysed. In particular, the coding of the road in segment lengths and angles was converted into a radius of curvature to create a summary metric and since. From a visual examination, the malfunction seemed to be caused by too narrow radiuses of curvature (less than half of the width of the road) which resulted in the edges of the road (offset of the median spline) crossing. This was thus assumed to be the cause of malfunction. Directional coverage analyses were also performed (using some test analysis code from sbft-cps-tool-competition/cps-tool-competition), to evaluate the quality of the search space, and the correlation analyses between the characteristics of the road and the presence of defects, in order both to understand the possible presence of a particular characteristic that causes the malfunction and to confirm the thesis described above.

This can be seen from the test results:

- 2-points roads: they are all correct (see figures A.1, A.2 and A.3)

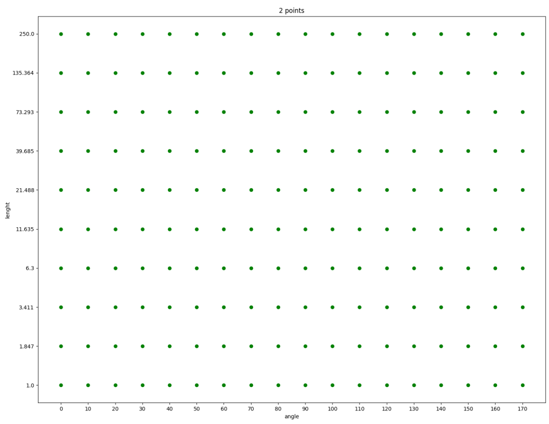


Figure A.1: 2-points roads test results

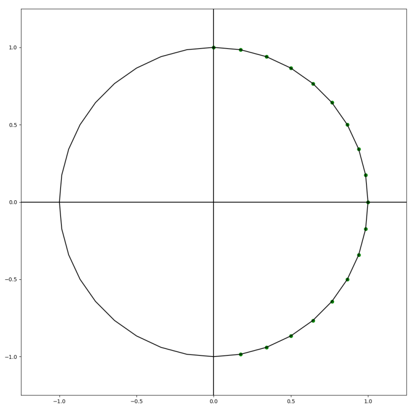
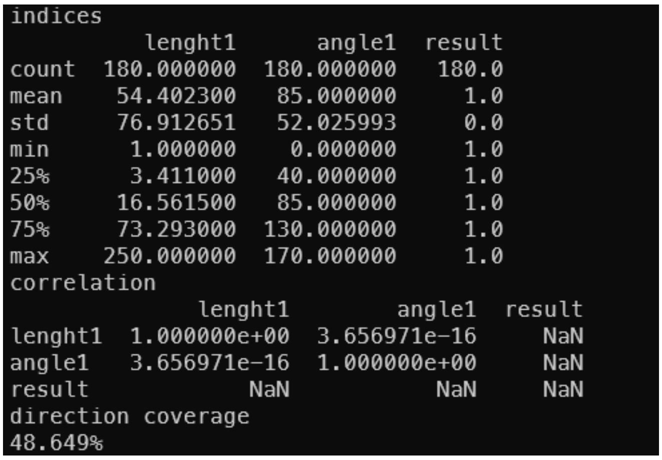


Figure A.2: 2-points roads direction coverage



```
indices
          lenght1      angle1  result
count  180.000000  180.000000   180.0
mean    54.402300   85.000000     1.0
std     76.912651   52.025993     0.0
min      1.000000    0.000000     1.0
25%      3.411000   40.000000     1.0
50%     16.561500   85.000000     1.0
75%     73.293000  130.000000     1.0
max    250.000000  170.000000     1.0
correlation
              lenght1        angle1  result
lenght1  1.000000e+00  3.656971e-16     NaN
angle1   3.656971e-16  1.000000e+00     NaN
result            NaN           NaN     NaN
direction coverage
48.649%
```

Figure A.3: 2-points roads analysis

- 3-points roads: we can visually notice the presence of a border between the road with and without a defect, reinforcing the above thesis, but from the analysis of curvatures there seems to be no correlation (see figures A.4, A.5, A.6, A.7, A.8, A.9, A.10 and A.11)

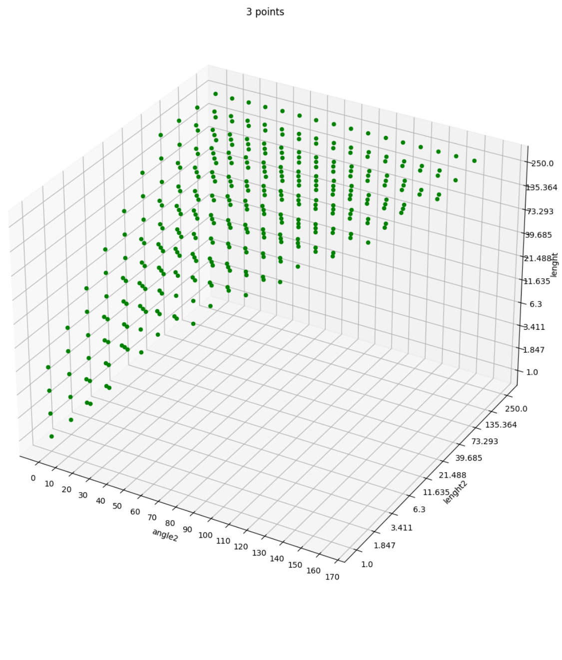


Figure A.4: 3-points roads test results

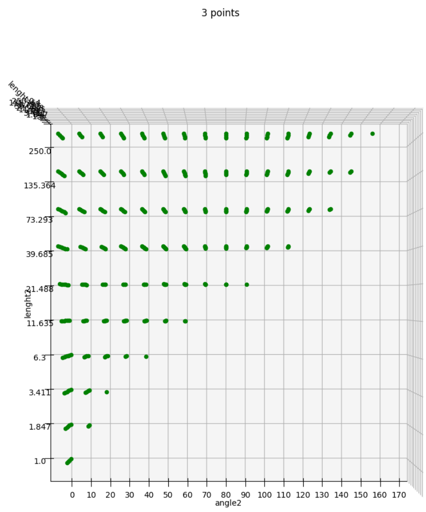


Figure A.5: 3-points roads test results - second angle and second segment length

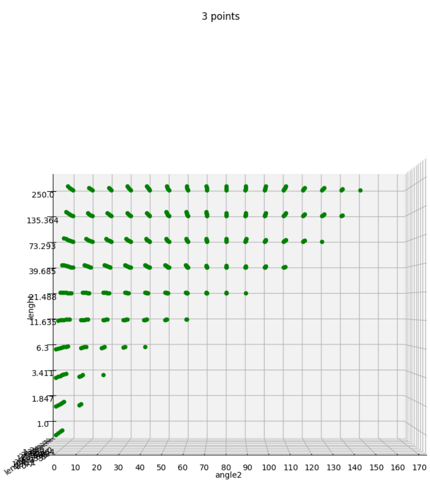


Figure A.6: 3-points roads test results - first angle and first segment length

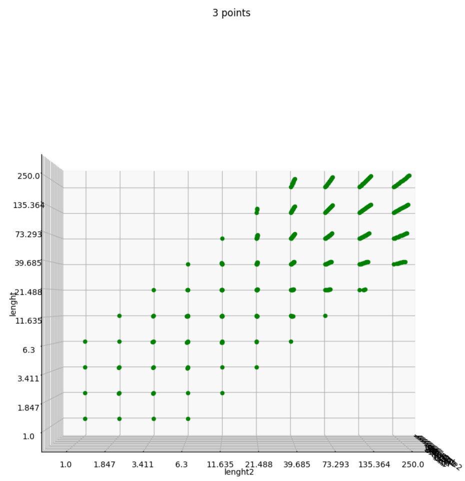


Figure A.7: 3-points roads test results - second segment length and first segment length

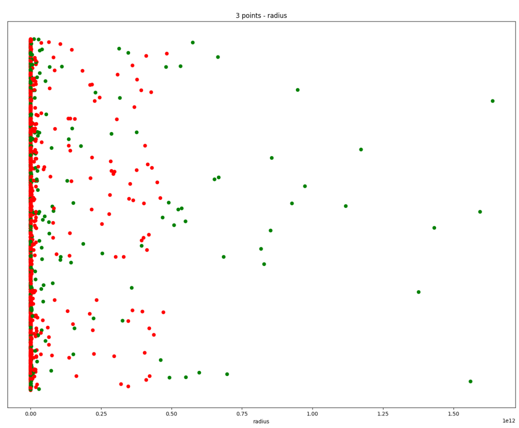


Figure A.8: 3-points roads curvature radius

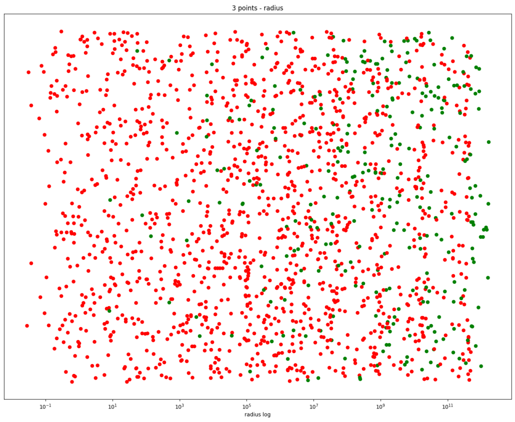


Figure A.9: 3-points roads curvature radius - logarithmic scale

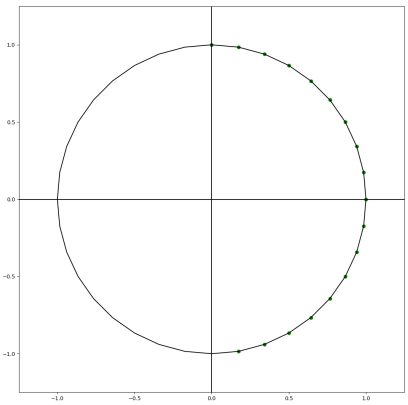


Figure A.10: 3-points roads direction coverage

```
indices
           lenght1  angle1      lenght2       angle2       result       radius1
count  1800.000000  1800.0  1800.000000  1800.000000  1800.000000  1.800000e+03
mean     54.402300     0.0    54.402300    85.000000     0.194444  3.171387e+10
std      76.720021     0.0    76.720021    51.895692     0.395882  1.340187e+11
min       1.000000     0.0     1.000000     0.000000     0.000000  0.000000e+00
25%       3.411000     0.0     3.411000    40.000000     0.000000  3.210455e+02
50%      16.561500     0.0    16.561500    85.000000     0.000000  8.134707e+05
75%      73.293000     0.0    73.293000   130.000000     0.000000  5.172592e+08
max     250.000000     0.0   250.000000   170.000000     1.000000  1.638211e+12
correlation
              lenght1  angle1       lenght2        angle2    result   radius1
lenght1  1.000000e+00     NaN -9.757306e-17 -6.927928e-16  0.207332  0.577993
angle1            NaN     NaN           NaN           NaN       NaN       NaN
lenght2 -9.757306e-17     NaN  1.000000e+00 -5.028335e-17  0.205768  0.141595
angle2  -6.927928e-16     NaN -5.028335e-17  1.000000e+00 -0.349570 -0.019486
result   2.073320e-01     NaN  2.057675e-01 -3.495698e-01  1.000000  0.232936
radius1  5.779926e-01     NaN  1.415954e-01 -1.948616e-02  0.232936  1.000000
direction coverage
48.649%
```

Figure A.11: 3-points roads analysis

- 4-points roads: from the analysis of the curvatures there does not seem to be a correlation (see figures A.12, A.13, A.14, A.15, A.16, A.17, A.18 and A.19)

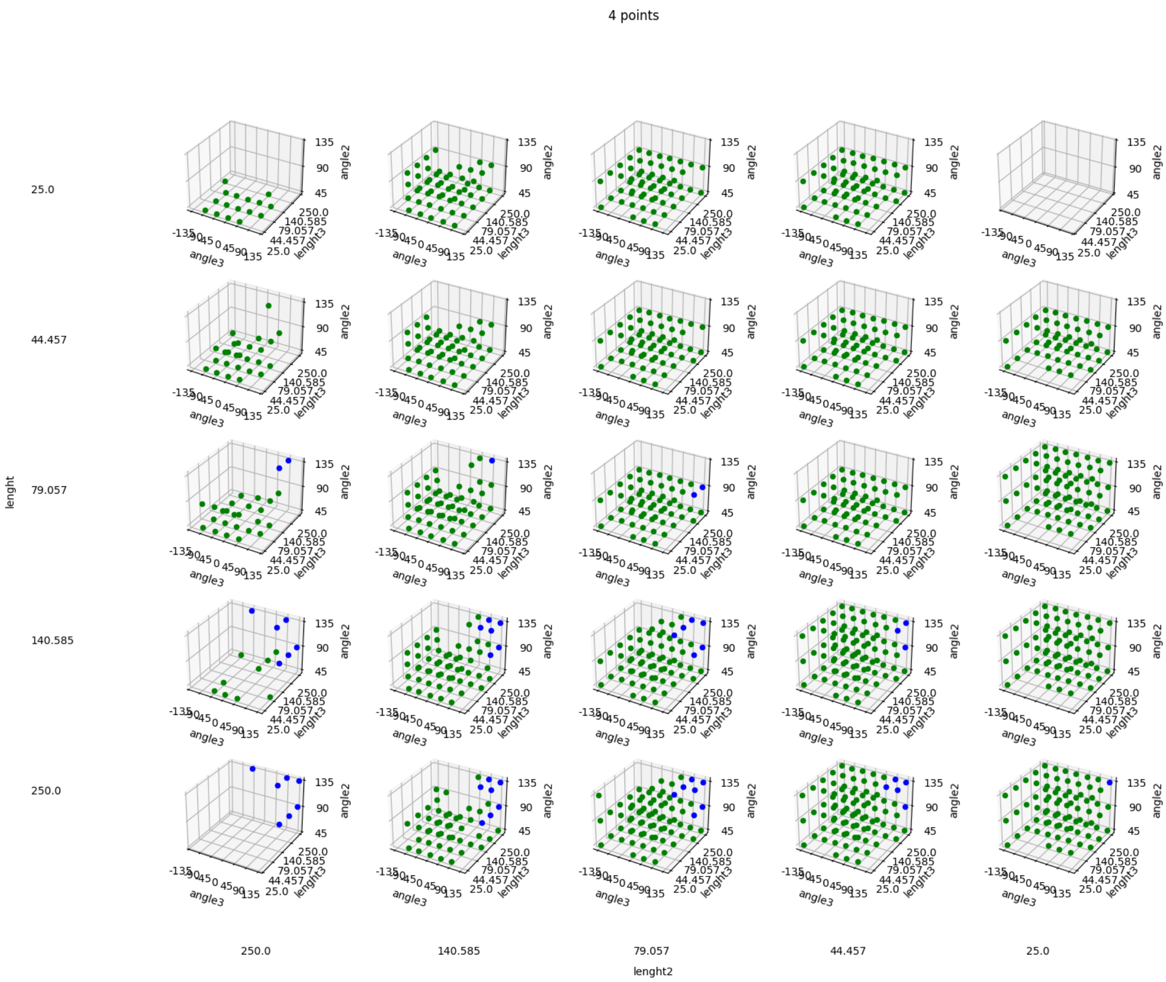


Figure A.12: 4-points roads test results

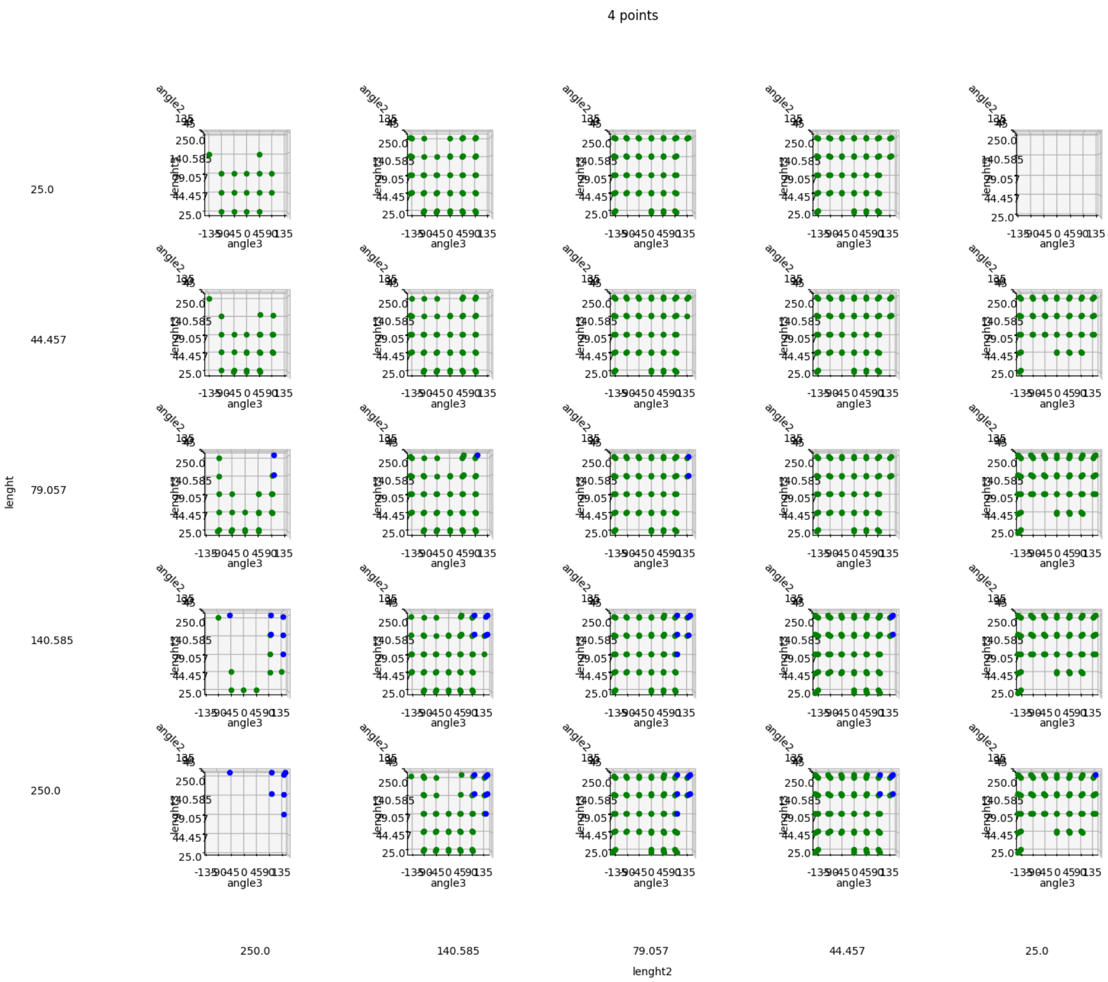


Figure A.13: 4-points roads test results - third angle and third segment length

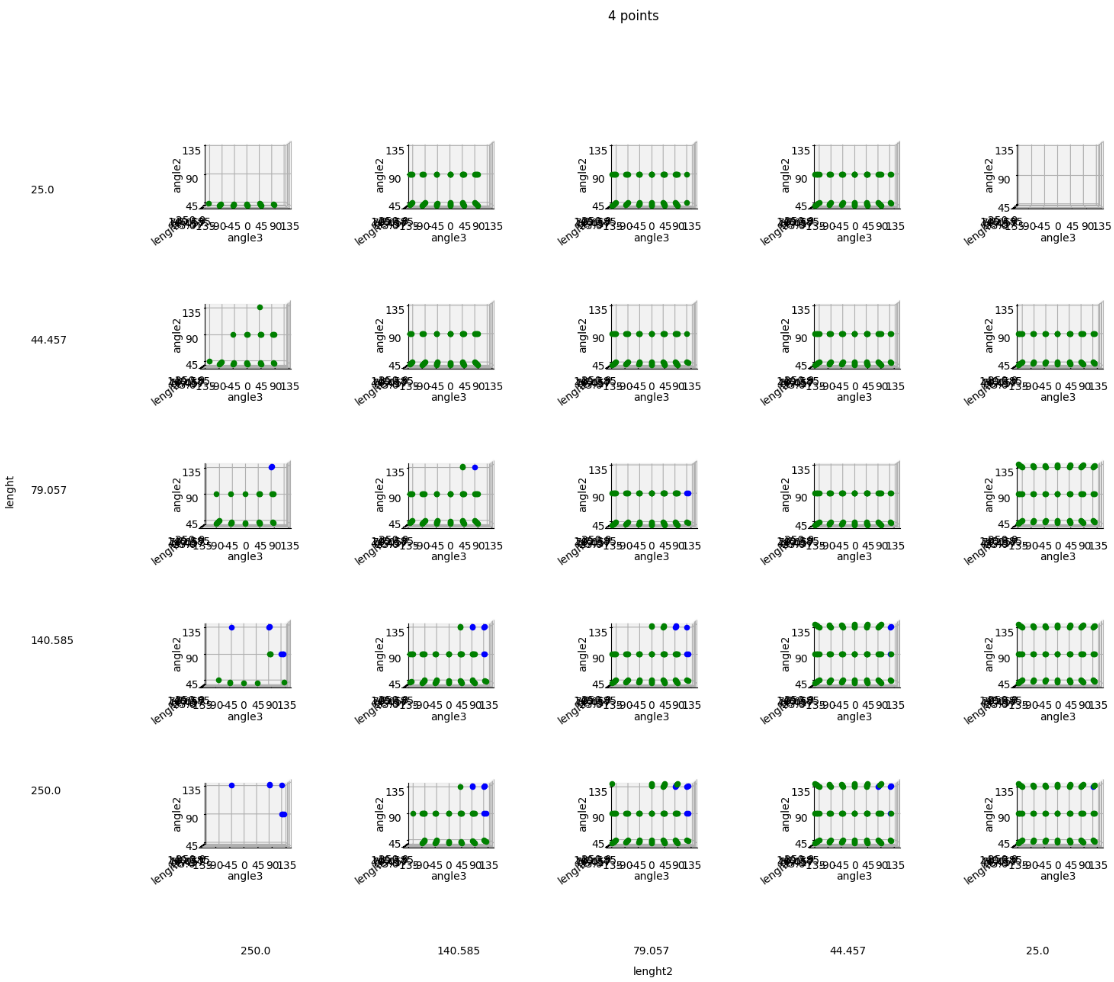


Figure A.14: 4-points roads test results - third angle and second angle

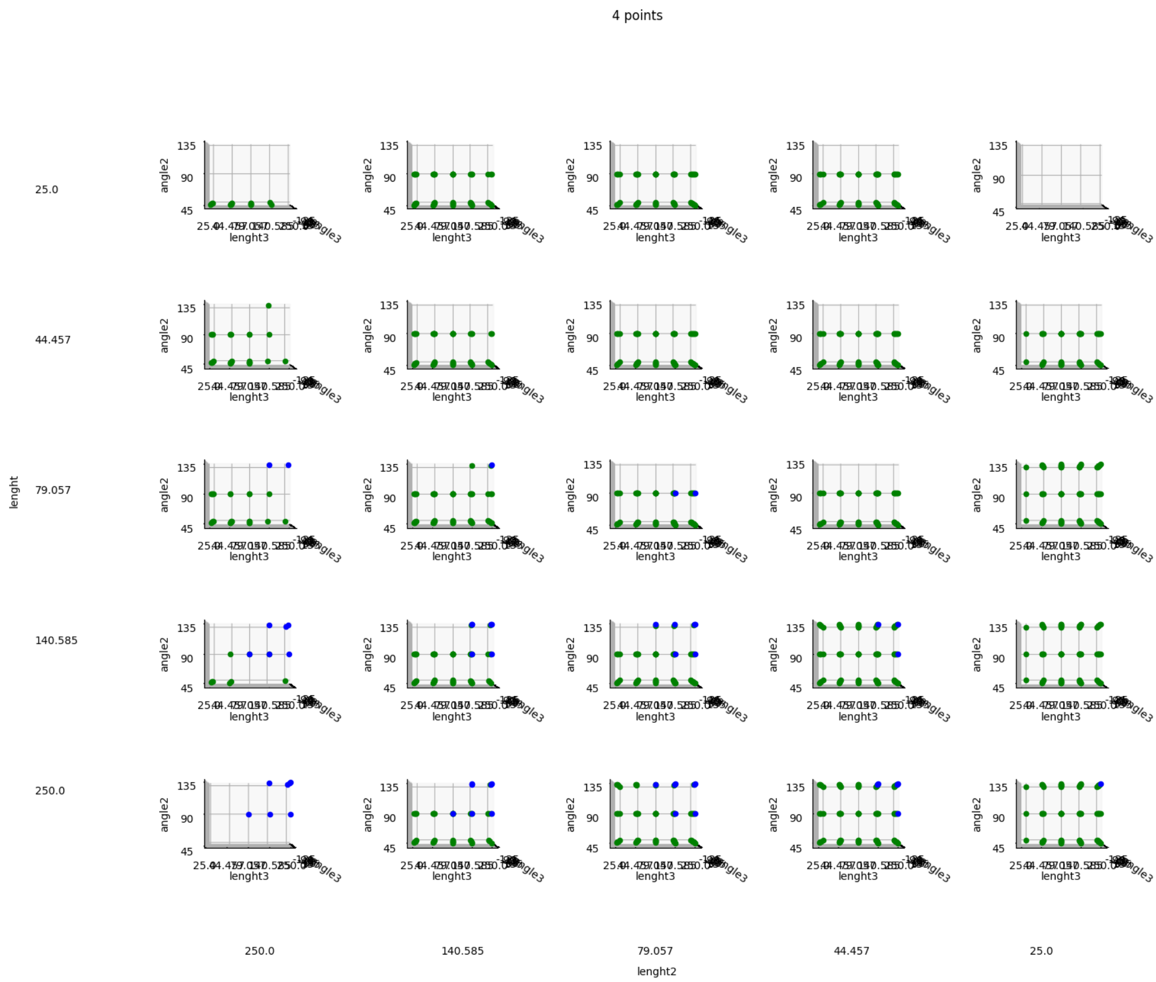


Figure A.15: 4-points roads test results - third segment length and second angle

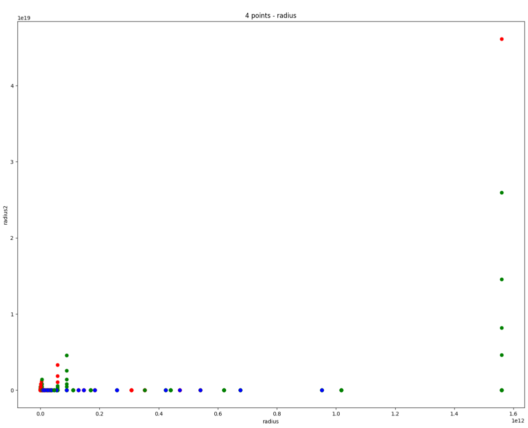


Figure A.16: 4-points roads curvature radius

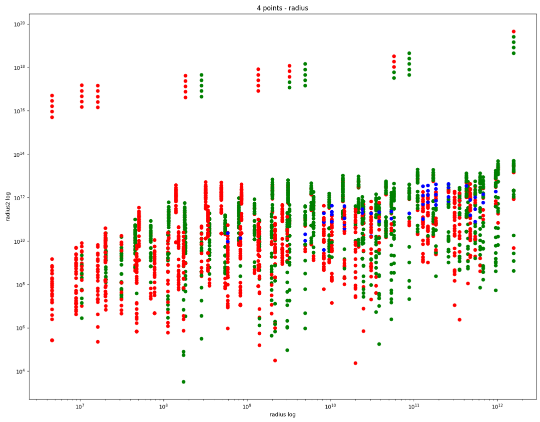


Figure A.17: 4-points roads curvature radius - logarithmic scale

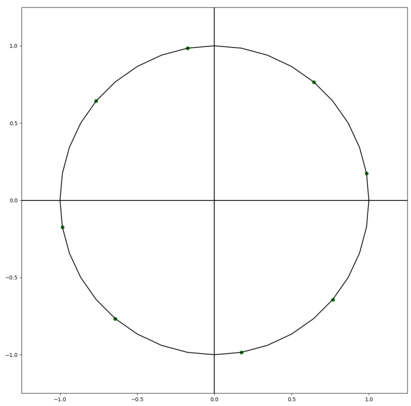


Figure A.18: 4-points roads direction coverage

indices

	lenght1	angle1	lenght2	angle2	lenght3	angle3	result	radius1	radius2
count	2625.000000	2625.0	2625.000000	2625.000000	2625.000000	2625.000000	2625.000000	2.625000e+03	2.625000e+03
mean	107.819800	0.0	107.819800	90.000000	107.819800	0.000000	0.526857	1.187125e+11	4.823267e+16
std	81.243587	0.0	81.243587	36.749347	81.243587	90.017148	0.538303	2.701159e+11	1.095805e+18
min	25.000000	0.0	25.000000	45.000000	25.000000	-135.000000	0.000000	4.615225e+06	0.000000e+00
25%	44.457000	0.0	44.457000	45.000000	44.457000	-90.000000	0.000000	1.836318e+08	1.770103e+09
50%	79.057000	0.0	79.057000	90.000000	79.057000	0.000000	1.000000	3.101947e+09	3.273424e+10
75%	140.585000	0.0	140.585000	135.000000	140.585000	90.000000	1.000000	5.737635e+10	4.670432e+11
max	250.000000	0.0	250.000000	135.000000	250.000000	135.000000	2.000000	1.559651e+12	4.614039e+19

correlation

	lenght1	angle1	lenght2	angle2	lenght3	angle3	result	radius1	radius2
lenght1	1.000000e+00	NaN	5.732205e-16	-1.663697e-15	-1.980751e-16	3.412357e-18	0.162575	7.110266e-01	0.061254
angle1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
lenght2	5.732205e-16	NaN	1.000000e+00	1.114471e-16	-6.574496e-17	4.739385e-18	0.142797	2.152730e-01	0.061255
angle2	-1.663697e-15	NaN	1.114471e-16	1.000000e+00	-1.145816e-16	-3.248053e-18	-0.319022	-1.066110e-01	-0.003886
lenght3	-1.980751e-16	NaN	-6.574496e-17	-1.145816e-16	1.000000e+00	-1.137452e-18	0.144953	1.940095e-17	-0.027018
angle3	3.412357e-18	NaN	4.739385e-18	-3.248053e-18	-1.137452e-18	1.000000e+00	0.002477	3.428537e-18	0.045609
result	1.625747e-01	NaN	1.427973e-01	-3.190216e-01	1.449529e-01	2.477392e-03	1.000000	1.551683e-01	0.001195
radius1	7.110266e-01	NaN	2.152730e-01	-1.066110e-01	1.940095e-17	3.428537e-18	0.155168	1.000000e+00	0.182129
radius2	6.125448e-02	NaN	6.125478e-02	-3.885707e-03	-2.701848e-02	4.560877e-02	0.001195	1.821286e-01	1.000000

direction coverage
21.622%

Figure A.19: 4-points roads analysis

- 5-points roads: from the analysis of the curvatures there does not seem to be a correlation (see figures A.20, A.21, A.22, A.23, A.24, A.25, A.26, A.27, A.28 and A.29)

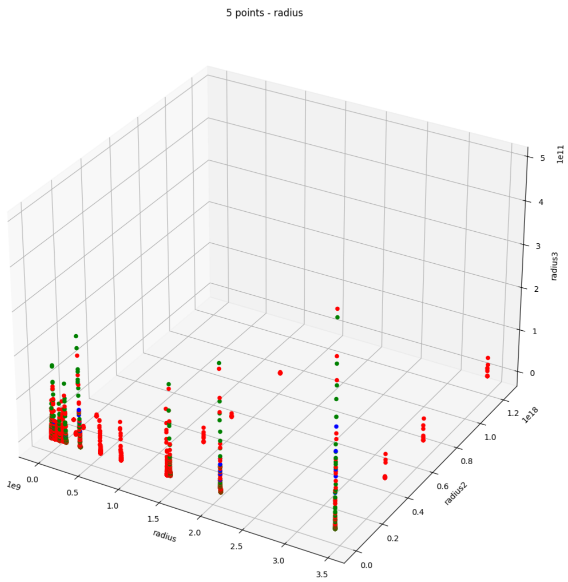


Figure A.20: 5-points roads curvature radius

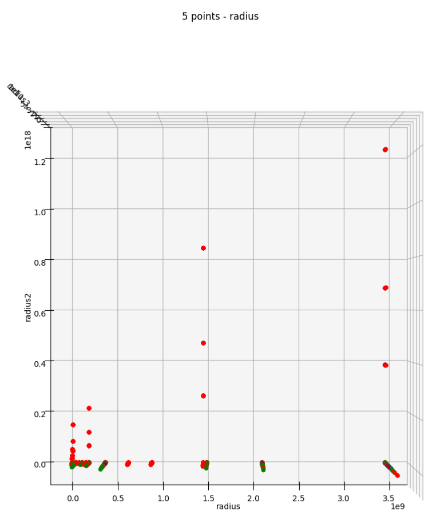


Figure A.21: 5-points roads curvature radius - first radius and second radius

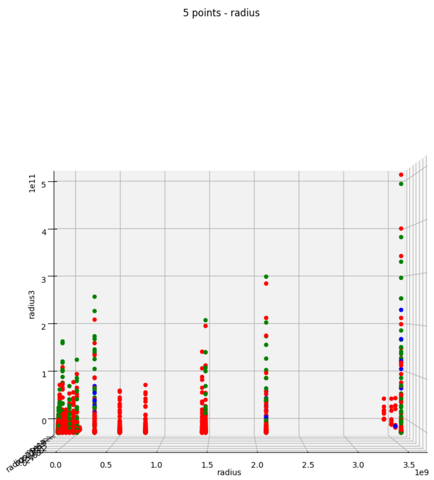


Figure A.22: 5-points roads curvature radius - first radius and third radius

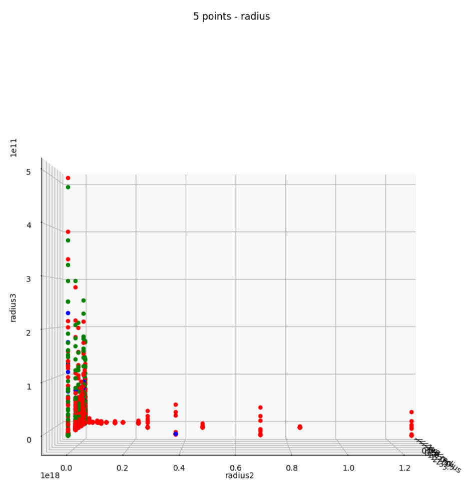


Figure A.23: 5-points roads curvature radius - second radius and third radius

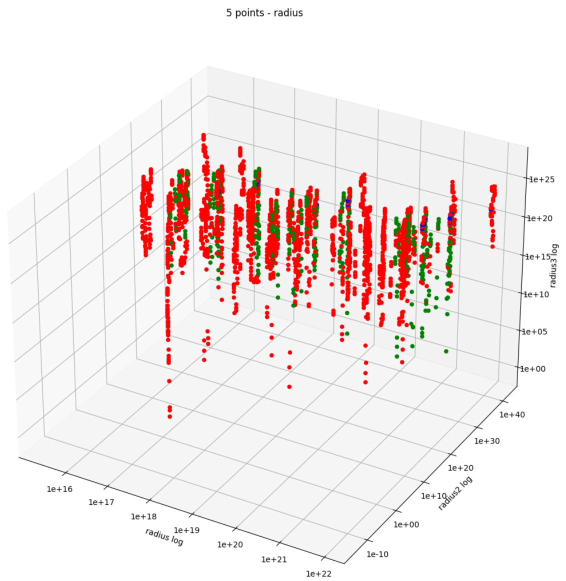


Figure A.24: 5-points roads curvature radius - logarithmic scale

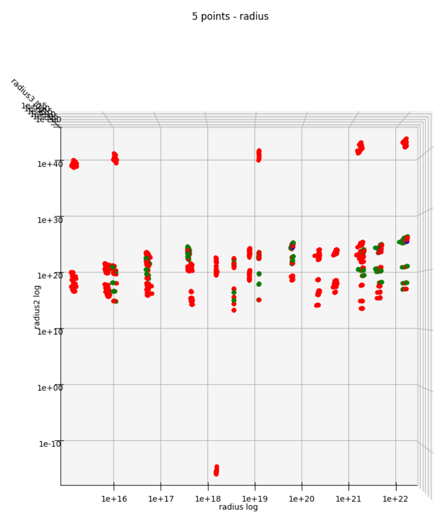


Figure A.25: 5-points roads curvature radius - first radius and second radius - logarithmic scale

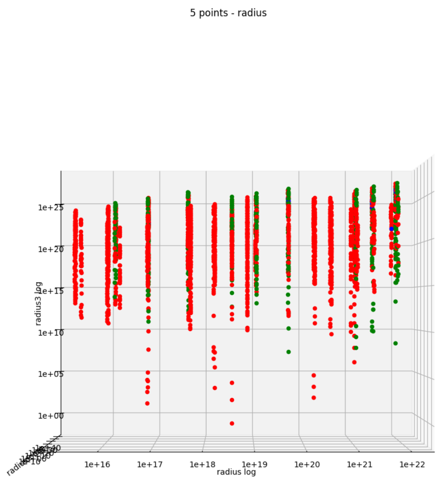


Figure A.26: 5-points roads curvature radius - first radius and third radius - logarithmic scale

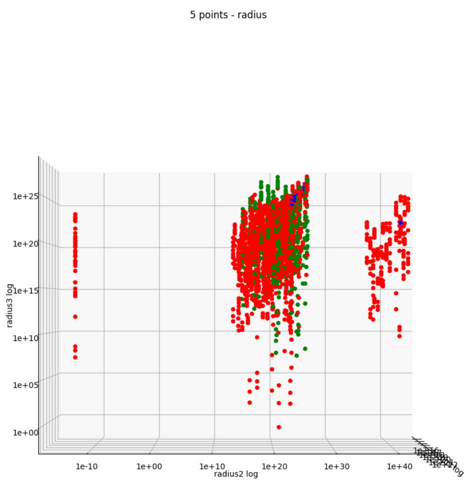


Figure A.27: 5-points roads curvature radius - second radius and third radius - logarithmic scale

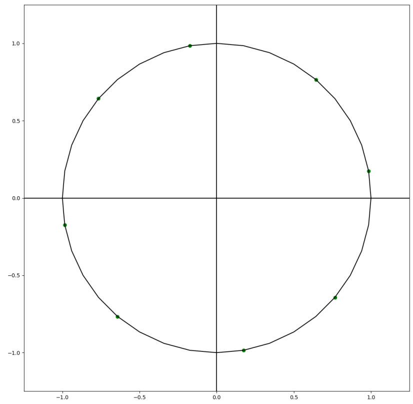
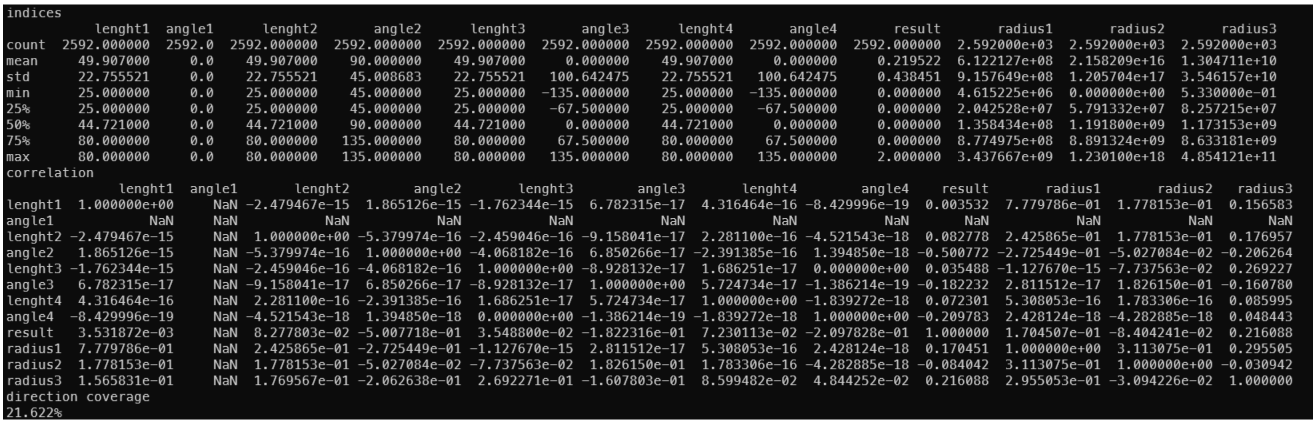


Figure A.28: 5-points roads direction coverage



indices

	lenght1	angle1	lenght2	angle2	lenght3	angle3	lenght4	angle4	result	radius1	radius2	radius3
count	2592.000000	2592.0	2592.000000	2592.000000	2592.000000	2592.000000	2592.000000	2592.000000	2592.000000	2.592000e+03	2.592000e+03	2.592000e+03
mean	49.907000	0.0	49.907000	90.000000	49.907000	0.000000	49.907000	0.000000	0.219522	6.122127e+08	2.158209e+16	1.304711e+10
std	22.755521	0.0	22.755521	45.008683	22.755521	100.642475	22.755521	100.642475	0.438451	9.157649e+08	1.205704e+17	3.546157e+10
min	25.000000	0.0	25.000000	45.000000	25.000000	-135.000000	25.000000	-135.000000	0.000000	4.615225e+06	0.000000e+00	5.330000e-01
25%	25.000000	0.0	25.000000	45.000000	25.000000	-67.500000	25.000000	-67.500000	0.000000	2.042528e+07	5.791332e+07	8.257215e+07
50%	44.721000	0.0	44.721000	90.000000	44.721000	0.000000	44.721000	0.000000	0.000000	1.358434e+08	1.191800e+09	1.173153e+09
75%	80.000000	0.0	80.000000	135.000000	80.000000	67.500000	80.000000	67.500000	0.000000	8.774975e+08	8.891324e+09	8.633181e+09
max	80.000000	0.0	80.000000	135.000000	80.000000	135.000000	80.000000	135.000000	2.000000	3.437667e+09	1.230100e+18	4.854121e+11

correlation

	lenght1	angle1	lenght2	angle2	lenght3	angle3	lenght4	angle4	result	radius1	radius2	radius3
lenght1	1.000000e+00	NaN	-2.479467e-15	1.865126e-15	-1.762344e-15	6.782315e-17	4.316464e-16	-8.429996e-19	0.003532	7.779786e-01	1.778153e-01	0.156583
angle1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
lenght2	-2.479467e-15	NaN	1.000000e+00	-5.379974e-16	-2.459046e-16	-9.158041e-17	2.281100e-16	-4.521543e-18	0.082778	2.425865e-01	1.778153e-01	0.176957
angle2	1.865126e-15	NaN	-5.379974e-16	1.000000e+00	-4.068182e-16	6.850266e-17	-2.391385e-16	1.394850e-18	-0.500772	-2.725449e-01	-5.027084e-02	-0.206264
lenght3	-1.762344e-15	NaN	-2.459046e-16	-4.068182e-16	1.000000e+00	-8.928132e-17	1.686251e-17	0.000000e+00	0.035488	-1.127670e-15	-7.737563e-02	0.269227
angle3	6.782315e-17	NaN	-9.158041e-17	6.850266e-17	-8.928132e-17	1.000000e+00	5.724734e-17	-1.386214e-19	-0.182232	2.811512e-17	1.826150e-01	-0.160780
lenght4	4.316464e-16	NaN	2.281100e-16	-2.391385e-16	1.686251e-17	5.724734e-17	1.000000e+00	-1.839272e-18	0.072301	5.308053e-16	1.783306e-16	0.085995
angle4	-8.429996e-19	NaN	-4.521543e-18	1.394850e-18	0.000000e+00	-1.386214e-19	-1.839272e-18	1.000000e+00	-0.209783	2.428124e-18	-4.282885e-18	0.048443
result	3.531872e-03	NaN	8.277803e-02	-5.007718e-01	3.548800e-02	-1.822316e-01	7.230113e-02	-2.097828e-01	1.000000	1.704507e-01	-8.404241e-02	0.216088
radius1	7.779786e-01	NaN	2.425865e-01	-2.725449e-01	-1.127670e-15	2.811512e-17	5.308053e-16	2.428124e-18	0.170451	1.000000e+00	3.113075e-01	0.295505
radius2	1.778153e-01	NaN	1.778153e-01	-5.027084e-02	-7.737563e-02	1.826150e-01	1.783306e-16	-4.282885e-18	-0.084042	3.113075e-01	1.000000e+00	-0.030942
radius3	1.565831e-01	NaN	1.769567e-01	-2.062638e-01	2.692271e-01	-1.607803e-01	8.599482e-02	4.844252e-02	0.216088	2.955053e-01	-3.094226e-02	1.000000

direction coverage
21.622%

Figure A.29: 5-points roads analysis

Due to these artifacts in the Road component, some options were evaluated:

- to use a Support Vector Machine (SVM) to search for the boundary between roads with and without artifacts and then use a Neural Network (NN) to try to pull the road inside the class of roads without artifacts; this option was not used because visually there is no boundary between roads with and without artifacts and pulling the road means changing the road geometry
- for roads with points too sparse or too dense, to delete or to add some points; this option was not used because there is some weak evidence that this might be the problem (see curvatures), but there is no evidence of a sweet spot and adding or removing points might slightly change the road geometry (by Catmull-Rom spline definition)
- scale all the roads, there is some evidence that roads with larger curvature have fewer artifacts; this option was not used because it would have simplified the autonomous driving agents' tests. In extreme cases, this would be like transforming a mountain hairpin into a highway turn

therefore no actions were taken, but we remained confident that the problem will be solved by the developers.

Bibliography

[1] Sadeq Almeaibed, Saba Al-Rubaye, Antonios Tsourdos, and Nicolas P. Avdelidis. Digital twin analysis to promote safety and security in autonomous vehicles. *IEEE Communications Standards Magazine*, 5(1):40–46, March 2021.

[2] ASAM e.V. *ASAM OpenDRIVE 1.8.0*. ASAM e.V., 2023. `https://www.asam.net/standards/detail/opendrive`.

[3] Ion Barosan, Arash Arjmandi Basmenj, Sudhanshu G. R. Chouhan, and David Manrique. Development of a virtual simulation environment and a digital twin of an autonomous driving truck for a distribution center. In Henry Muccini, Paris Avgeriou, Barbora Buhnova, Javier Camara, Mauro Caporuscio, Mirco Franzago, Anne Koziolek, Patrizia Scandurra, Catia Trubiani, Danny Weyns, and Uwe Zdun, editors, *Software Architecture*, pages 542–557, Cham, 2020. Springer International Publishing.

[4] BeamNG GmbH. BeamNG.tech. `https://www.beamng.tech/`.

[5] Matteo Biagiola, Andrea Stocco, Vincenzo Riccio, and Paolo Tonella. Two is better than one: digital siblings to improve autonomous driving testing. *Empirical Software Engineering*, 29(4):72, May 2024.

[6] Markus Borg, Raja Ben Abdessalem, Shiva Nejati, François-Xavier Jegeden, and Donghwan Shin. Digital twins are not monozygotic – cross-replicating adas testing in two industry-grade automotive simulators. In *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pages 383–393, April 2021.

[7] Markus Borg, Raja Ben Abdessalem, Shiva Nejati, François-Xavier Jegeden, and Donghwan Shin. Digital twins are not monozygotic – cross-replicating adas testing in two industry-grade automotive simulators. In *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pages 383–393, April 2021.

[8] Markus Borg, Raja Ben Abdessalem, Shiva Nejati, François-Xavier Jegeden, and Donghwan Shin. Digital twins are not monozygotic – cross-replicating adas testing in two industry-grade automotive simulators. In *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pages 383–393, April 2021.

[9] Vinton G. Cerf. A comprehensive self-driving car test. *Commun. ACM*, 61(2):7, January 2018.

[10] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[11] Michael G. Kapteyn, Jacob V. R. Pretorius, and Karen E. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Computational Science*, 1(5):337–347, 2021.

[12] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[13] Pascale Maul, Marc Mueller, Fabian Enkler, Eva Pigova, Thomas Fischer, Lefteris Stamatogiannakis. BeamNG.tech Technical Paper. `https://beamng.tech/blog/2021-06-21-beamng-tech-whitepaper/bng_technical_paper.pdf`.

[14] Omer San. The digital twin revolution. *Nature Computational Science*, 1(5):307–308, 2021.

[15] Andrea Stocco, Brian Pulfer, and Paolo Tonella. Mind the gap! a study on the transferability of virtual versus physical-world testing of autonomous driving systems. *IEEE Transactions on Software Engineering*, 49(4):1928–1940, April 2023.

[16] Omar Veledar, Violeta Damjanovic-Behrendt, and Georg Macher. Digital twins for dependability improvement of autonomous driving. In Alastair Walker, Rory V. O'Connor, and Richard Messnarz, editors, *Systems, Software and Services Process Improvement*, pages 415–426, Cham, 2019. Springer International Publishing.

[17] Heuijee Yun and Daejin Park. Simulation of self-driving system by implementing digital twin with gta5. In *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–2, Jan 2021.

[18] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.