UNIVERSITY OF SALERNO

DEPARTMENT OF INFORMATION ENGINEERING, ELECTRICAL ENGINEERING AND APPLIED MATHEMATICS

PROJECT REPORT

# Autonomous Vehicle Driving

prof. Vento Mario
prof. Greco Antonio

*Group 07*

| Students | Id number |
| --- | --- |
| Alberti Andrea | 0622702370 |
| Attianese Carmine | 0622702355 |
| Capaldo Vincenzo | 0622702347 |
| Esposito Paolo | 0622702292 |


Academic Year 2024 – 2025

# CONTENTS

# ABSTRACT

This project presents the design, implementation, and evaluation of an autonomous driving system developed within the CARLA simulation environment. Starting from a baseline agent equipped with essential components such as global planner, behavior planner, local planner, and PID/Stanley controllers, we performed an extensive analysis to identify the system's critical limitations in handling complex driving scenarios.

The improved system introduces a set of significant enhancements, including advanced pedestrian detection, vehicle handling, lane invasion management, cyclist and obstacle handling, and proper reaction to traffic lights, stop signs, and intersections. A modular architecture inspired by Autoware ensures system scalability and maintainability. Special attention has been dedicated to calibrating both longitudinal and lateral controllers to achieve smooth and stable vehicle dynamics.

The system was evaluated on three predefined routes designed to cover various challenging scenarios, including road narrowing, overtaking maneuvers, intersections, and interactions with dynamic actors such as vehicles, pedestrians, and cyclists. Results show a significant reduction in infractions compared to the baseline, with full route completion and improved compliance with traffic rules.

Despite these improvements, the system presents some known limitations. The maximum speed is limited to 30 km/h to maintain safety margins, causing repeated infractions for failing to respect the minimum speed requirement. Increasing the vehicle maximum speed would require further improvements in trajectory control and navigation accuracy. Additionally, rare minor lateral collisions may occur during road narrowing or bike overtaking in extremely constrained spaces. Finally, as the system has been tested on a limited number of routes, its behavior on untested scenarios remains to be verified through further experimentation.

# CHAPTER 1

# INTRODUCTION

The purpose of this project is to design, implement and evaluate an autonomous driving system that respects as much as possible the road rules. The project is based on the CARLA simulator and, as a starting point, we rely on a baseline agent that serves both as a reference and initial implementation. The baseline already implements the main components of an autonomous driving system, that are the global planner, the behavior planner, the local planner, along with PID controller for longitudinal control and Stanley controller for lateral control. Our goal is to analyze the behavior of the baseline system and enhance its performance by addressing specific scenarios that were either not managed or poorly handled. To achieve this goal, we first analyzed the baseline agent to understand its limitations. Based on this analysis, we developed an improved agent designed to better comply with the road code, minimizing infractions and handling critical driving scenarios more effectively.

## 1.1 Route analysis

To evaluate the performance of the autonomous agent, we have three predefined routes: Route0, Route1 and Route4. These routes are designed to test various aspects of driving behavior, including navigation through intersections, interaction with static and dynamic obstacles, and compliance with traffic rules. Specific scenarios include road work zones, traffic lights, stop signs, and interactions with other vehicles and pedestrians.

| Route | Description | Illumination | Weather |
|---|---|---|---|
| Route0 | Urbanization route focused on junction-crossing vehicles. | Daylight, good illumination. | Light cloudiness, no rain, dry road, mild wind. |
| Route1 | Curved road route, focusing on overtaking stopped vehicles or obstacles. Bumpy road. | Night, poor illumination. Transition from good to bad lighting due to worsening weather. | Severe conditions: from moderate rain to heavy rain, high wetness and deposits, strong wind, dense fog. |
| Route4 | Rural route focused on crossing actors. | Daylight, good illumination. | Moderate to strong precipitation, dry road, wind intensity increasing. Constant fog density. |

Table 1.1: Overview of evaluation routes and corresponding environmental conditions.

The selected routes may involve a range of challenging situations typically encountered in real-world driving, such as:

- Lane merging
- Lane changing
- Negotiations at traffic intersections
- Negotiations at roundabouts
- Handling traffic lights and traffic signs
- Yielding to emergency vehicles
- Coping with pedestrians, cyclists, and other dynamic elements

In particular, these are the events, in detail, that happen within the routes:

| **Event** | **Description** | **Route(s)** |
|---|---|---|
| *AccidentTwoWays* | An accident is simulated. Vehicles remain stationary and must be overtaken to continue the route. | 1–4 |
| *BlockedIntersection* | The intersection is blocked by a stationary vehicle. | 1 |
| *ControlLoss* | The vehicle skids due to irregularities in the road surface. | 1 |
| *ConstructionObstacleTwoWays* | A traffic warning signal blocks the lane, making it impassable. | 1 |
| *DynamicObjectCrossing* | A pedestrian suddenly crosses the lane. | 1–4 |
| *HardBreakRoute* | The leading vehicle suddenly stops, remains still, then resumes movement. | 1 |
| *HazardAtSideLaneTwoWays* | Cyclists are driving in the lane and stop at some point. The vehicle must overtake them. | 1 |
| *InvadingTurn* | Oncoming vehicles partially invade the ego lane due to road narrowing. | 1 |
| *NonSignalizedJunctionRightTurn* | The vehicle encounters an unsignalized intersection. | 1–4 |
| *ParkedObstacleTwoWays* | A parked vehicle on the roadside must be overtaken. | 1 |
| *VehicleTurningRoute* | After a turn, the vehicle encounters a cyclist. | 4 |
| *VehicleTurningRoutePedestrian* | After a turn, the vehicle encounters a pedestrian. | 4 |

Table 1.2: Description of the events occurring in the evaluation routes.

## 1.2 Evaluation Criteria

The performance of the autonomous agent is assessed through a set of metrics designed to reflect its ability to complete routes safely and in compliance with traffic rules:

- **Driving Score**: represents the overall performance of the agent. It is defined as:

$$\text{Driving Score} = \text{Route Completion} \times \text{Infraction Penalty}$$

- **Route Completion**: the percentage of the total route distance successfully completed by the agent. A higher percentage indicates better coverage of the planned route.
- **Infraction Penalty**: a multiplicative penalty based on the infractions committed during the route. The score starts at 1.0 and is reduced for each infraction according to predefined weights.

Finally, each evaluation metric is averaged over all completed routes to produce the final performance indicators, referred to as **Global Metrics**. The **Global Driving Score** is the primary metric used to rank the overall performance of the agent.

### 1.2.1 Infraction Penalty Weights

The Infraction Penalty is computed as the product of all penalty factors corresponding to the infractions committed by the agent during a route. Each infraction type is assigned a penalty factor between 0 and 1. The lower the penalty factor, the more critical the infraction is considered. Since the final penalty is computed multiplicatively, severe infractions (with lower factors) have a significantly larger negative impact on the overall Driving Score. This design ensures that the evaluation prioritizes both safety and compliance with traffic regulations. The table below summarizes the penalty factors assigned to each type of infraction.

| **Infraction Type** | **Penalty Factor** |
|---|---|
| Collision with pedestrian | 0.50 |
| Collision with vehicle | 0.60 |
| Collision with static object | 0.65 |
| Running a red light | 0.70 |
| Running a stop sign | 0.80 |
| Scenario timeout (stuck for 4 minutes) | 0.70 |
| Failure to maintain minimum speed | 0.70 |
| Failure to yield to emergency vehicle | 0.70 |
| Off-road driving | No addition to route completion |

Table 1.3: Infraction penalty factors.

# CHAPTER 2

# BASELINE ANALYSIS

This chapter presents an analysis of the baseline agent, a reference implementation of the autonomous driving system that relies on a simple, rule-based decision-making strategy without advanced perception or planning capabilities. The chapter begins by describing the main components of the system, followed by an overview of its Operational Design Domain (ODD). Subsequently, we report the results of the baseline evaluation across multiple routes, highlighting the system's limitations in handling complex scenarios. A detailed analysis of the most frequent and penalized infractions provides the foundation for identifying the key areas of improvement to be addressed in the next development phases.

## 2.1 Component Description

The baseline agent is composed of few high-level and low-level components, each responsible for a specific aspect of the vehicle's behavior, planning, and control.

- **Global Planner**: the global planner operates on a global scale. It computes a high-level route from the starting point to the destination using waypoints defined by the map.
- **Local Planner**: the local planner refines the global route into detailed, temporally feasible trajectories. It generates short-term waypoint targets and velocity profiles for the controller to follow.
- **Behaviour Agent**: this is the central decision-making component of the baseline system. It implements its behaviour through a rule based approach for managing different situation. In order of priority, from higher to lower, it includes the following:
    1. **traffic light management**: takes effect when the ego vehicle encounters a red traffic light;
    2. **pedestrian avoidance**: allows the ego vehicle to stop when it is too close to a pedestrian;
    3. **vehicle avoidance**: allows the ego vehicle to stop when it is too close to a vehicle;
    4. **car following**: the ego vehicle follows the leading vehicle if it is at a safety distance, otherwise it starts decelerate;
    5. **intersection behavior**: takes effect in proximity of intersections, slowing down the vehicle;
    6. **normal behavior**: the ego vehicle follows the preset max speed or the speed limit when none of the listed situations happen.
- **Controller**: is responsible for translating the planned trajectory into low-level actuator commands. It includes two key controllers: PID Controller for longitudinal control and Stanley Controller for lateral control.

## 2.2 Operational Design Domain (ODD)

The Operational Design Domain (ODD) defines the specific conditions under which the autonomous agent is intended to operate safely and effectively. It includes environmental, infrastructural, and dynamic constraints that bound the system's capabilities.

| Category | Sub-domain | Description of Capabilities and Limitations |
|---|---|---|
| **Scenery** | **Roadway Types** | The system can operate on urban roads with one or two lanes, as well as on rural or suburban roads with a single lane. In two-lane roads featuring tight curves, the vehicle may encroach into the adjacent lane. |
| | **Fixed Road Structures** | The system can detect and obey traffic lights. However, it fails to recognize most traffic signs, such as stop signs, and does not follow traffic rules at intersections, entering them prematurely. |
| | **Temporary Road Structures** | The system is unable to detect temporary static objects such as roadworks, roadside parked vehicles, or temporary warnings. These scenarios typically result in collisions. |
| | **Speed Limits** | The system is capable of retrieving the current road's speed limit and maintains its speed below that threshold. |
| **Dynamic Elements** | **Traffic** | The system is able to operate in low to moderate traffic conditions. It does not implement overtaking logic: if the road is blocked, the vehicle remains stationary. In the presence of cyclists in the lane, the vehicle either fails to detect them (leading to a collision) or remains stopped behind them, obstructing traffic. It also fails to adapt in scenarios involving lane narrowing, where oncoming vehicles partially encroach into the ego lane. The system does not adjust its lateral position to avoid the obstacle and remains centered in the lane, resulting in side collisions. |
| | **Pedestrian Crossing** | The system poorly manages scenarios in which a pedestrian crosses the road unexpectedly outside designated crossing points, particularly near or after intersections, often resulting in collisions. |
| **Environmental Conditions** | **Illumination** | The system can operate during both daylight and nighttime conditions, handling both good and poor lighting scenarios. |
| | **Weather** | The system can handle cloudy or wet weather conditions. However, in rainy scenarios, it may lose control during emergency braking maneuvers. |

Table 2.1: Baseline Operational Design Domain (ODD) and its limitations

## 2.3 Baseline evaluation

| | Route0 | Route1 | Route4 | Global |
|---|---|---|---|---|
| **Driving Score** | 77.444 | 0.7158 | 1.2902 | 26.4833 |
| **Route Completion** | 100.00 | 12.14 | 100.00 | 70.7133 |
| **Infraction Penalty** | 0.7744 | 0.0589 | 0.0129 | 0.2820 |
| **Infractions** | | | | |
| Collision with pedestrian | 0 | 0 | 3 | 3 |
| Collision with vehicle | 0 | 4 | 2 | 6 |
| Collision with static object | 0 | 1 | 0 | 1 |
| Running a red light | 0 | 0 | 0 | 0 |
| Running a stop sign | 1 | 0 | 4 | 5 |
| Scenario timeout (stuck for 4 minutes) | 0 | 1 | 1 | 2 |
| Failure to maintain minimum speed | 1 | 0 | 0 | 1 |
| Failure to yield to emergency vehicle | 0 | 0 | 0 | 0 |
| Off-road driving | 0 | 0 | 0 | 0 |
| **Total** | 2 | 6 | 10 | **18** |

Table 2.2: Overview of the baseline performance on the routes.

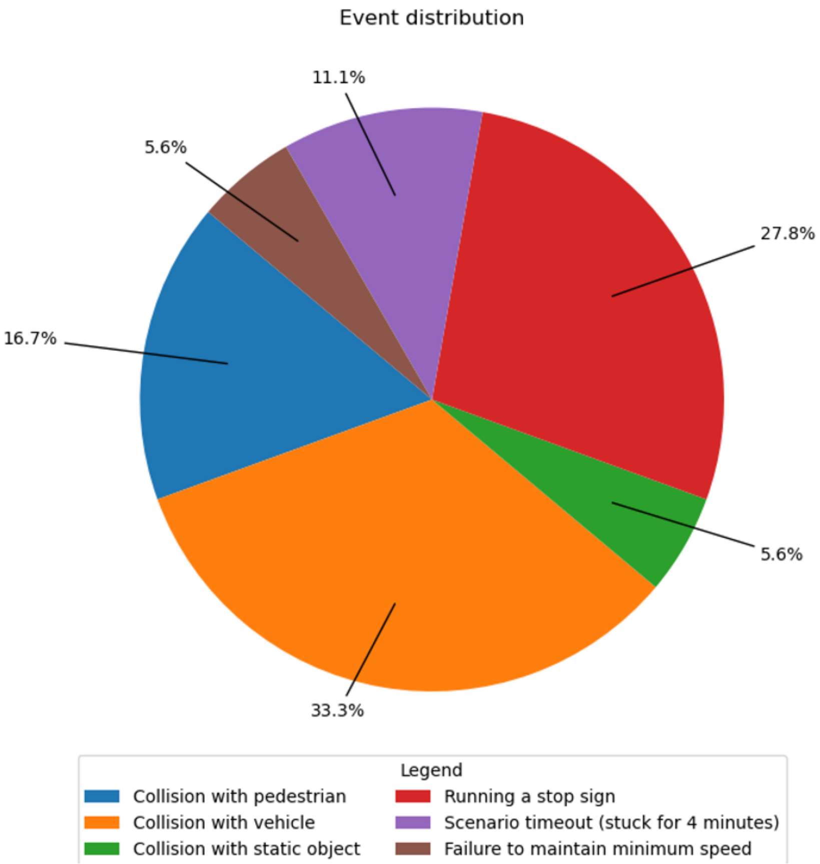



Figure 2.1: Infraction Distribution

## 2.4 Discussion of Baseline Results

The results clearly show that the baseline is unable to handle complex events. Although it successfully completes Route0 and Route4 with a 100% completion rate, it commits severe infractions that significantly lower the Driving Score. The most frequent and impactful infractions involve collisions, particularly with vehicles and pedestrians, as well as failures to respect stop signs, which the baseline is unable to manage. A detailed analysis of the main infractions reveals the following:

- **Collision with pedestrians:** this is the most heavily penalized infraction (0.5) and occurs in 16.7% of the cases. The baseline doesn't always stop in the presence of pedestrians, without adopting appropriate behavior for such scenarios.
- **Collision with vehicles:** this is one of the most heavily penalized infractions (0.6), accounting for 33.3% of all infractions. It results from the baseline's inability to properly handle:
  - parked vehicles;
  - lane narrowing situations;
  - interactions with other vehicles when crossing intersections.
- **Collision with static objects**: this type of infraction represents only 5.6% of the total, but it still has a significant impact on the Driving Score (penalty 0.65). It is caused by the baseline's inability to detect nearby static objects.
- **Running a stop signs**: the baseline does not include stop sign handling, resulting in systematic failure in these scenarios.
- **Scenario timeout**: this may cause the route to be interrupted, negatively affecting both the infraction penalty and the completion percentage.
- **Minimum speed infraction**: this represents a marginal portion of the infractions, with relatively low impact on the Driving Score (0.7).

In order to improve the baseline, our strategy will focus primarily on addressing the most heavily penalized and frequent infractions, as these have the greatest impact on the Driving Score. Specifically, efforts will be concentrated on improving pedestrian management, ensuring correct interaction with other vehicles, at intersections, enhancing the management of static objects, and implementing stop sign handling. Less critical aspects, such as minimum speed violations, will be addressed in a later phase.

# CHAPTER 3

# SYSTEM IMPROVEMENTS

In this chapter, we describe and analyze the enhancements introduced to the initial baseline. We begin with the calibration of both longitudinal and lateral controllers, followed by an investigation of the key events, introduced in the previous section, that led to modifications in the vehicle's behavior.

We created a **Custom driving mode** other then the three options outlined in the baseline (*Cautious*, *Normal* and *Aggressive*). This style includes the following predefined parameters:

| **Parameter** | **Value** |
|---|---|
| Max Speed | 30 |
| Speed Limit Distance | 3 |
| Speed Decrease | 10 |
| Safety Time | 3 |
| Min Proximity Threshold | 10 |
| Braking Distance | 3 |
| Tailgate Counter | 0 |

Table 3.1: Parameters and their values

## 3.1 Controller Calibration

Controllers play a central role in maintaining the stability and responsiveness of an autonomous vehicle. Their calibration is essential for ensuring smooth operation, and thus we began our improvement process by refining both longitudinal and lateral controllers. The longitudinal dynamics are managed by a **PID controller**, while lateral dynamics are handled by a **Stanley controller**.

The calibration process was initially conducted on a minimal test route specifically designed to simplify the analysis of the controllers' behavior. This route featured two curves, one right and one left, and three straight segments (*route_0.xml*). The purpose of this setup was to isolate the fundamental dynamics of the vehicle without interference from complex maneuvers or unpredictable traffic interactions.

Since the simulation is run with a fixed time step, specifically *fixed_delta_second = 0.05*, we set the time step $dt$ for both controllers accordingly.

### 3.1.1 Longitudinal Control

The evaluation of the longitudinal controller, tasked with speed regulation, was conducted primarily during the first straight segment of the route, under ideal conditions: absence of traffic, pedestrians, and without the influence of lateral dynamics. This allowed for a clean and isolated analysis of the controller's behavior in response to a simple speed-following task.
The baseline configuration used the following parameters:

| **Parameter** | **Value** |
|---|---|
| $K_P$ | 0.888 |
| $K_I$ | 0.0768 |
| $K_D$ | 0.05 |
| $dt$ | 0.05 |

Table 3.2: Baseline PID Parameters

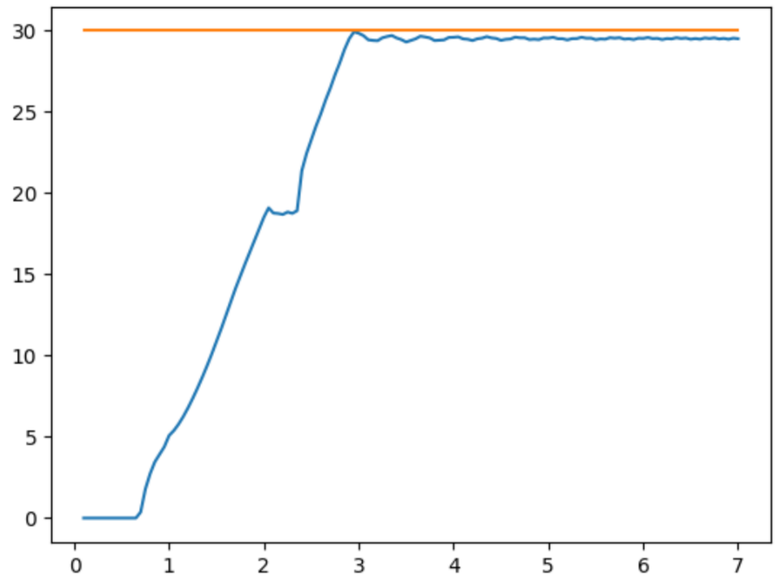



Figure 3.1: Baseline speed profile

The results obtained with the baseline configuration of the PID controller, as shown in Figure 3.1, were already quite satisfactory. The system was able to reach the target cruising speed of 30 km/h with no overshoot and a relatively comfortable acceleration profile. However, small oscillations and a noticeable steady-state error were observed as the speed approached the reference value. These signs suggested that the PID gains could be further optimized to improve stability and accuracy.

The tuning process began with a reduction of the proportional gain $K_p$, which led to a decrease in the oscillations observed near the steady state, resulting in a smoother convergence toward the target velocity. Subsequently, the integral gain $K_i$ was increased to reduce the persistent steady-state error, helping the system to better eliminate long-term deviations from the desired speed. The derivative gain $K_d$ was left unchanged, as the absence of overshoot indicated that its initial value was already appropriate.
The resulting configuration is:

| **Parameter** | **Value** |
|---|---|
| $K_P$ | 0.7 |
| $K_I$ | 0.2 |
| $K_D$ | 0.05 |
| $dt$ | 0.05 |

Table 3.3: Final PID Parameters

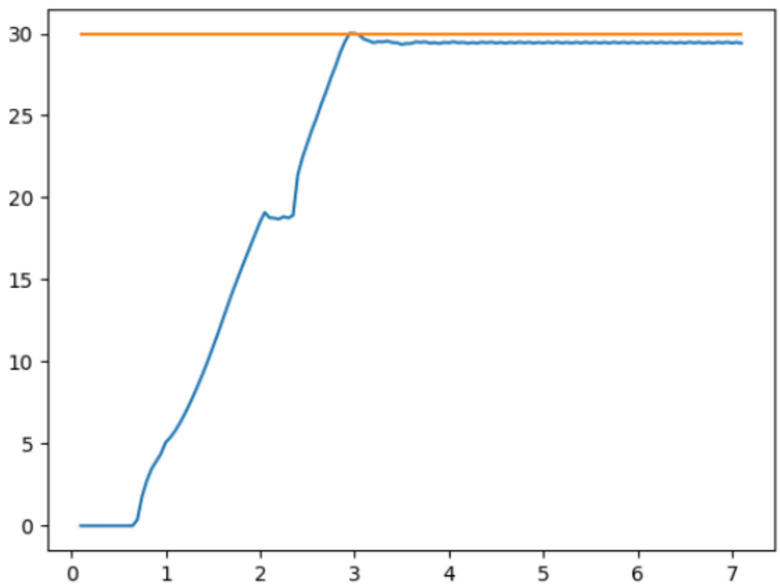



Figure 3.2: Speed profile after PID tuning

As shown in Figure 3.2, the controller, after tuning, demonstrates significantly improved performance: the speed closely follows the reference with a negligible steady-state error, no overshoot, and minimal oscillations. Additionally, the rise time remains acceptable, and the acceleration remains within comfortable limits for the driver, confirming the effectiveness of the tuning process.

### 3.1.2 Lateral Control

Unlike the longitudinal controller, the lateral controller, implemented through the Stanley method, required a more complex and iterative calibration procedure. Due to the nature of lateral control, the behavior of the vehicle must be validated not only in simple scenarios but also across a variety of curves and dynamic conditions. While the initial two curves in the test route allowed for a preliminary assessment of the turning response, they were not sufficient to ensure robust performance under more realistic conditions. Therefore, the calibration process was extended to a broader set of scenarios involving overtaking maneuvers and higher curvature roads. These tests were essential to fine-tune the parameters and ensure a stable and responsive steering behavior without excessive oscillations or understeering.
The baseline configuration was:

| **Parameter** | **Value** |
|---|---|
| $K_V$ | 4.0 |
| $K_S$ | 1.0 |
| $dt$ | 0.05 |

Table 3.4: Baseline Stanley Parameters

The final parameters were determined through iterative testing in diverse environments, leading to the following configuration:

| **Parameter** | **Value** |
|---|---|
| $K_V$ | 1.5 |
| $K_S$ | 3.0 |
| $dt$ | 0.05 |

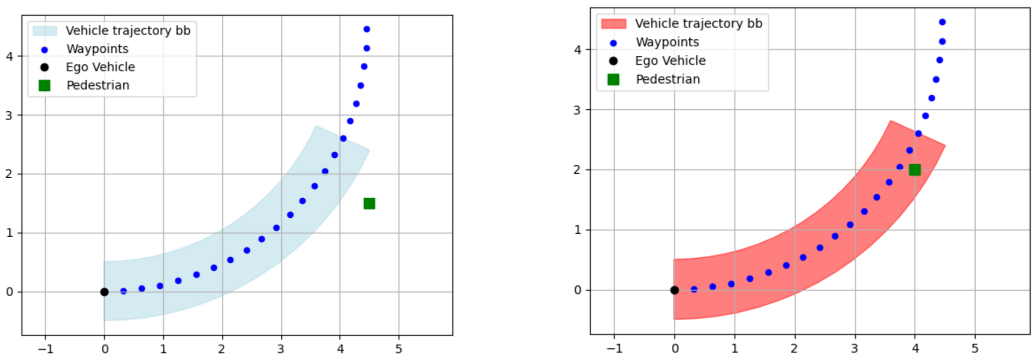
Table 3.5: Final Stanley Parameters

## 3.2 Pedestrian Handler

In our autonomous vehicle system, pedestrian management is handled through a perception and detection module specifically designed to identify potential collisions with pedestrians and react accordingly. The system continuously monitors the environment by retrieving all pedestrian actors present in the scene. It filters these actors based on their proximity to the vehicle, considering only those within a certain distance threshold, that in our case is set to 10 meters.

The core of the logic relies on determining whether a pedestrian is positioned along the vehicle's current or intended path, with different strategies applied depending on the driving context. When the vehicle is not located within a junction, pedestrian detection is based on two conditions: the pedestrian must be within the perception module's field of view and within a predefined safety distance. If both conditions are met, the pedestrian is considered a potential obstacle, and an emergency stop is performed.

However, when the vehicle is approaching or located inside a junction, a more precise evaluation is required due to the complexity of the scenario. In this case, the system performs a geometric reasoning process: it constructs a polygon representing the planned route of the vehicle and compares it to the pedestrian's position. If the pedestrian's position intersects with this route polygon, the system identifies the pedestrian as a potential obstacle and provides the necessary information, such as their distance and location. This logic is used for the perception of all the successive cases.




(a) Injuction without pedestrian.

(b) Injuction with pedestrian.

Figure 3.3: Figure 3.3a illustrates the vehicle's logic at an intersection when the pedestrian is outside the vehicle's trajectory. Figure 3.3b shows the case where a pedestrian intersects the projected path of the vehicle, triggering detection and notifying the high-level autonomous agent.

## 3.3 Vehicle Handler

Collisions with other vehicles represent one of the primary hazards for an autonomous vehicle and can originate from different situations. A rear-end collision may occur if the system fails to properly manage the following distance from a vehicle ahead. Similarly, a stationary vehicle on the roadside may not be correctly detected and avoided, leading to a potential impact. Furthermore, vehicles invading the ego-vehicle's lane, for example, due to road narrowing or irregular maneuvers, require adjustment on the original vehicle trajectory to prevent the collision.

To mitigate these risks, the system first performs vehicle perception tailored to different driving scenarios, ensuring appropriate detection based on the context. Tailgating situations are actively managed during lane-following operations to maintain safe following behavior. If a vehicle is detected ahead in the same lane, the system calculates the distance to that vehicle. If the distance is below the defined braking threshold, an emergency braking maneuver is triggered to avoid a collision. Otherwise, adaptive following logic is applied to maintain a safe distance. Additionally, the system monitors for situations where vehicles on the opposite lane, invade the ego-vehicle's lane and adjusts its behavior to safely handle these events. The following sections describe the strategies implemented to handle these scenarios and ensure safe navigation.

### 3.3.1 Lane invasion handling

A critical aspect of ensuring the safety of the autonomous vehicle is the ability to detect and react to vehicles from the opposite lane that invade the ego-vehicle's lane. This situation can arise due the road narrowing (*InvadingTurn* event). To handle this, the system continuously monitors the surrounding environment, paying particular attention to vehicles located on the adjacent lane respect to the ego-vehicle, which typically corresponds to vehicles traveling in the opposite direction on a standard two-way road.

The detection process begins by retrieving the position of nearby vehicles and determining the lane they occupy. If a vehicle is identified as being in the opposite lane, its position is analyzed in more detail to assess whether it poses a potential risk. To do this, the system calculates the relative position vector between the ego-vehicle and the detected vehicle. This vector is then projected onto two key directions defined by the ego-vehicle's current orientation: the forward direction, representing the longitudinal axis of travel, and the right direction, representing the lateral axis.

By projecting the relative position vector onto these axes, the system obtains the longitudinal distance (how far ahead the vehicle is) and the lateral distance (how far to the side the vehicle is, with positive values to the right and negative to the left). These two values allow the system to precisely determine the position of the intruding vehicle relative to the ego-vehicle (see the example at Figure 3.4).

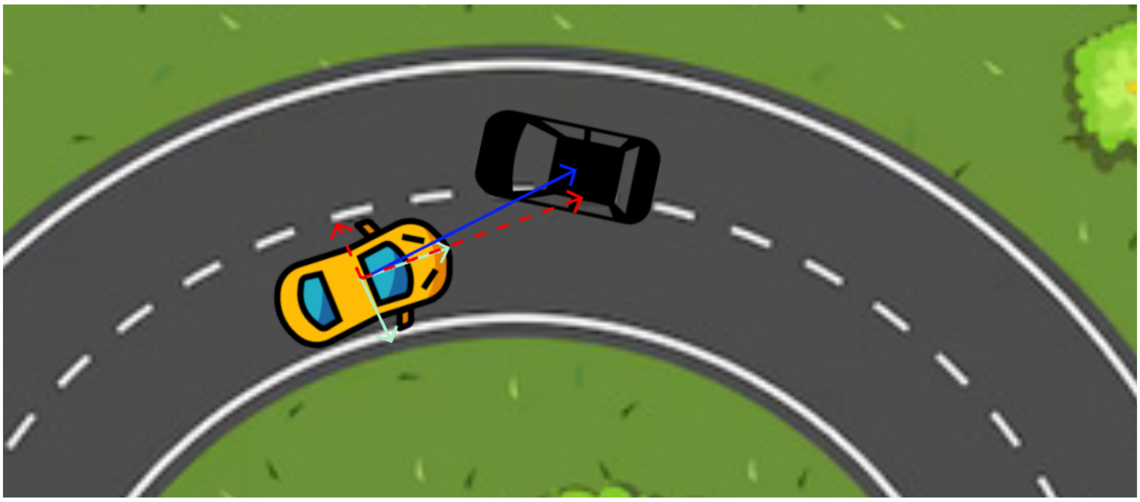


Figure 3.4: The green lines represent the unit vectors defining the forward direction and the right-hand direction of the autonomous agent. The blue vector indicates the relative position of the oncoming vehicle with respect to the ego-vehicle. The red vectors correspond to the projection of the blue vector onto the forward and right-hand unit vectors, effectively decomposing the relative position into longitudinal and lateral components. The evasive maneuver is applied only if a set of specific conditions regarding the lateral and longitudinal distances are satisfied.

If the vehicle is located ahead of the ego-vehicle within a predefined longitudinal threshold and the lateral distance is small enough to indicate a potential intrusion into the ego-vehicle's lane, the system considers the situation hazardous. In this case, a lateral correction is applied to the steering, designed to gently move the vehicle away from the threat. The intensity of this correction is proportional to how close the intruding vehicle is laterally, ensuring a smooth and controlled maneuver.

It is important to note that these corrections are disabled when the vehicle is inside a junction, as complex maneuvering logic already governs the vehicle's behavior in such contexts.

### 3.3.2 Car following behavior

In addition to collision avoidance with vehicles, the system uses a dedicated car following logic to safely manage vehicles ahead in the same lane. The goal is to maintain a safe distance and adjust speed dynamically to avoid rear-end collisions.

The logic is based on the estimation of the **Time to Collision (TTC)**

$$\text{ttc} = \frac{\text{distance\_from\_vehicle\_ahead}}{\Delta v}$$

which considers both the distance to the vehicle ahead and the relative speed difference ($\Delta v$). If the $TTC$ falls below a safety threshold, the system reduces the speed to maintain a safe distance. Specifically:

- **Critical Distance**: if the $TTC$ is below the configured safety time, the vehicle slows down significantly to avoid a potential collision.
- **Caution Zone**: if the $TTC$ is between two times safety threshold and the safety threshold, the vehicle adapts its speed to match that of the vehicle in front, ensuring a smooth following behavior.
- **Normal Conditions**: when the $TTC$ is large enough, the vehicle proceeds at its desired cruising speed.

## 3.4 Bike Handler

The autonomous agent implements a dedicated behavior to safely handle the presence of bicycles along its path. This mechanism aims to prevent dangerous situations when approaching cyclists, by dynamically adapting speed and trajectory based on their position.

First, the system constantly monitors the surroundings through a perception module capable of detecting bicycles ahead of the vehicle within a specific threshold. When a bicycle is detected in the same lane, the system immediately begins evaluating the distance to the cyclist.

If the distance to the bicycle falls below a critical threshold, the system applies specific behaviors depending on the severity of the situation. In particular:

- If the distance is lower then 3 meters and the adjacent lane is free, the system performs the overtaking with a small left steer.
- If the adjacent lane is not free and the distance is less than 1 meter, the vehicle executes an emergency braking maneuver to prevent collision.
- For distances between 1 and 4 meters, the vehicle adopts a car-following behavior, maintaining a safe distance behind the cyclist and adjusting speed accordingly.

In case the bike crosses the road, the vehicle performs an emergency braking avoiding the collision.

## 3.5 Object Handler

The autonomous agent includes a dedicated procedure to handle the presence of static obstacles along its path. Obstacles considered by this logic include stopped vehicles and other static props detected on or near the driving lane. The process begins with a perception step where the environment is scanned to identify both static objects and nearby vehicles within configurable distances.

If a stopped vehicle is detected in the ego lane, the system calculates the distance to the obstacle, get information about the detected obstacles and slow-down the speed. The agent computes whether there is enough distance and time to safely perform an overtaking maneuver (considering the speed of the oncoming vehicles on the overtaking lane), and if the space on the adjacent overtaking lane is clear the overtaking is performed. When overtaking is not possible—either due to limited space or traffic on the adjacent lane—the vehicle switches to a car-following behavior, maintaining a safe distance behind the vehicle.

The same logic applies to static obstacles that may obstruct the lane. The system identifies these objects, estimates their distance, and decides whether to slow down or overtake. The overtaking decision is subject to the same safety checks as before.

### 3.5.1 Overtake

The overtaking maneuver allows the autonomous vehicle to safely pass static obstacles and vehicle ahead, parked on the side of the road, when road conditions permit (i.e., traffic condition on the adjacent lane). The logic relies on dynamically generating a temporary trajectory that guides the vehicle into the adjacent lane, travels a configurable distance to complete the overtake, and then returns to the original lane.

To perform the overtaking maneuver when all the conditions are met, the system generates a temporary trajectory that guides the vehicle through the overtaking process and ensures a safe return to the original lane. This procedure is as follows:

1. **Trajectory Generation**: the system samples the current position and constructs a new path by:
   - Shifting to the adjacent lane using the available lane information.
   - Continuing along the adjacent lane for a predefined distance, ensuring enough space to safely complete the maneuver.
   - Returning to the original lane once the overtake is complete.
2. **Integration with Existing Plan**: after generating the overtaking path, it is merged with the remaining portion of the original global plan, ensuring continuity in navigation.

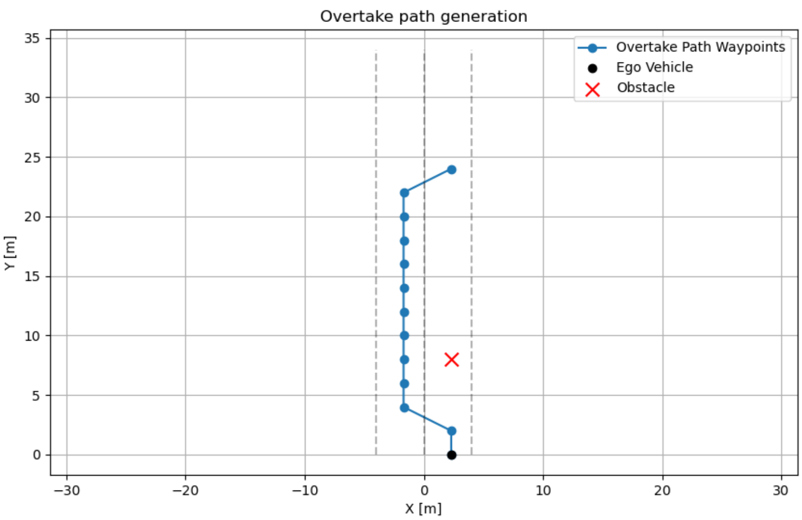



Figure 3.5: Example of path generated for the overtake. The length of the trajectory depends on the length of the obstacle.

## 3.6 Stop Sign Handler

The stop sign management module ensures that the autonomous vehicle correctly identifies and reacts to stop signs along its route. The system continuously monitors the environment, detecting all stop signs within a certain distance.

When a stop sign is detected within the relevant distance and positioned along the same road and driving direction as the vehicle, the system determines that the stop sign affects the vehicle's trajectory. To avoid redundant reactions to the same stop sign, each sign is identified by its unique ID, and previously handled signs are ignored. These checks are essential to ensure that the vehicle only reacts to stop signs intended for its direction of travel. For example, if there is a stop sign positioned after an intersection but directed towards other traffic flows (figure 3.6), the vehicle must ignore it, even if the sign falls within the distance threshold.

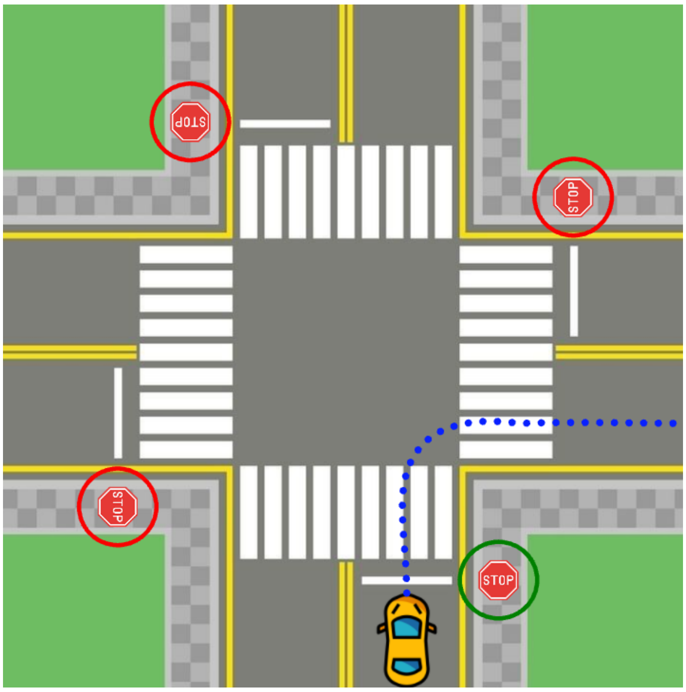



Figure 3.6: Example of stop sign management. The stop signs highlighted in red have no effect on the autonomous agent because they are either not located on the same road, not within the predefined distance, or not oriented towards the vehicle's driving direction. Only the green stop sign is considered. Thanks to these checks, stop signs placed after intersections but intended for vehicles coming from other lanes or directions are correctly ignored by the system.

Upon detecting a valid stop sign, the system initiates a controlled stop sequence. The vehicle applies braking until it comes to a complete stop. Once stopped, it remains stationary for a predefined amount of time. After this pause, the system allows the vehicle to safely resume its route, adjusting its speed accordingly.

## 3.7 Traffic Lights Handler

The traffic light management logic follows the same structure and reasoning as the stop sign handling module. The system continuously monitors the surrounding environment to detect traffic lights that may affect the vehicle's trajectory. As with stop signs, detection is based on the relative position, distance, and orientation of the traffic light with respect to the vehicle.

Only traffic lights positioned along the same road and facing the vehicle's direction of travel are considered. This prevents the vehicle from reacting to irrelevant traffic lights intended for other lanes or directions, such as those positioned after an intersection for cross traffic.

When a relevant traffic light is detected within the predefined distance, the system checks its state. If the light is red, the vehicle executes a controlled stop, remaining stationary until the light turns green. Once the traffic light permits movement, the vehicle safely resumes its route.

## 3.8 Junction Handler

In urban environments, managing intersections represents one of the most complex challenges for an autonomous vehicle. When the agent detects, via environmental perception, the presence of an intersection (i.e., the next waypoint falls within a zone classified as a *junction*), specific behavioral logics are activated. First, the vehicle checks for the presence of a stop sign. If such a sign is detected, the agent performs a controlled stop before proceeding, in accordance with traffic regulations. Subsequently, the system evaluates the need to yield to other vehicles. This check involves several strategies:

- The system analyzes the presence of stationary vehicles either in the same lane or in the lane the ego vehicle is about to enter.
- It considers the possibility that another vehicle may cross the ego vehicle's future trajectory. This verification is performed using a geometric analysis based on polygons that represent predicted trajectories: if an intersection is detected, the system estimates the arrival time at the intersection for both vehicles. If the other vehicle is expected to arrive within a dangerously short time window, the ego vehicle stops to ensure safety.
- Lastly, the system checks for the presence of nearby moving vehicles. In such cases as well, the agent adopts a cautious behavior and yields.

If no hazardous condition the agent proceeds along its path using the local planner. If the intersection involves a turn, a lower speed is set to ensure a smooth and safe maneuver.

This modular and hierarchical strategy allows the vehicle to handle intersections safely, efficiently, and in compliance with traffic rules, thereby reducing the risk of collisions and improving interaction with other road users.

## 3.9 Priority Hierarchy of Behavioral Handlers

The behavior agent is the central decision-making component of the system. It implements its behaviour through a rule based approach for managing different situation. In order of priority, from higher to lower, it includes the following:

1. Pedestrian handler
2. Stop sign handler
3. Traffic light handler
4. Junction handler
5. Vehicle handler
6. Bike handler
7. Obstacle handler

When none of the listed situations happen, the ego vehicle follows the preset max speed or the road speed limit.

# CHAPTER 4

# SYSTEM ARCHITECTURE

The architecture of our autonomous agent is inspired by the modular structure of Autoware. In figure 4.1 is shown the high-level system architecture, where the system is divided into clearly defined components: Perception, Behavior, Planning and Controller. Each component is responsible for a specific functionality, described in the previous chapter, allowing for a structured and scalable approach to decision-making and vehicle control.

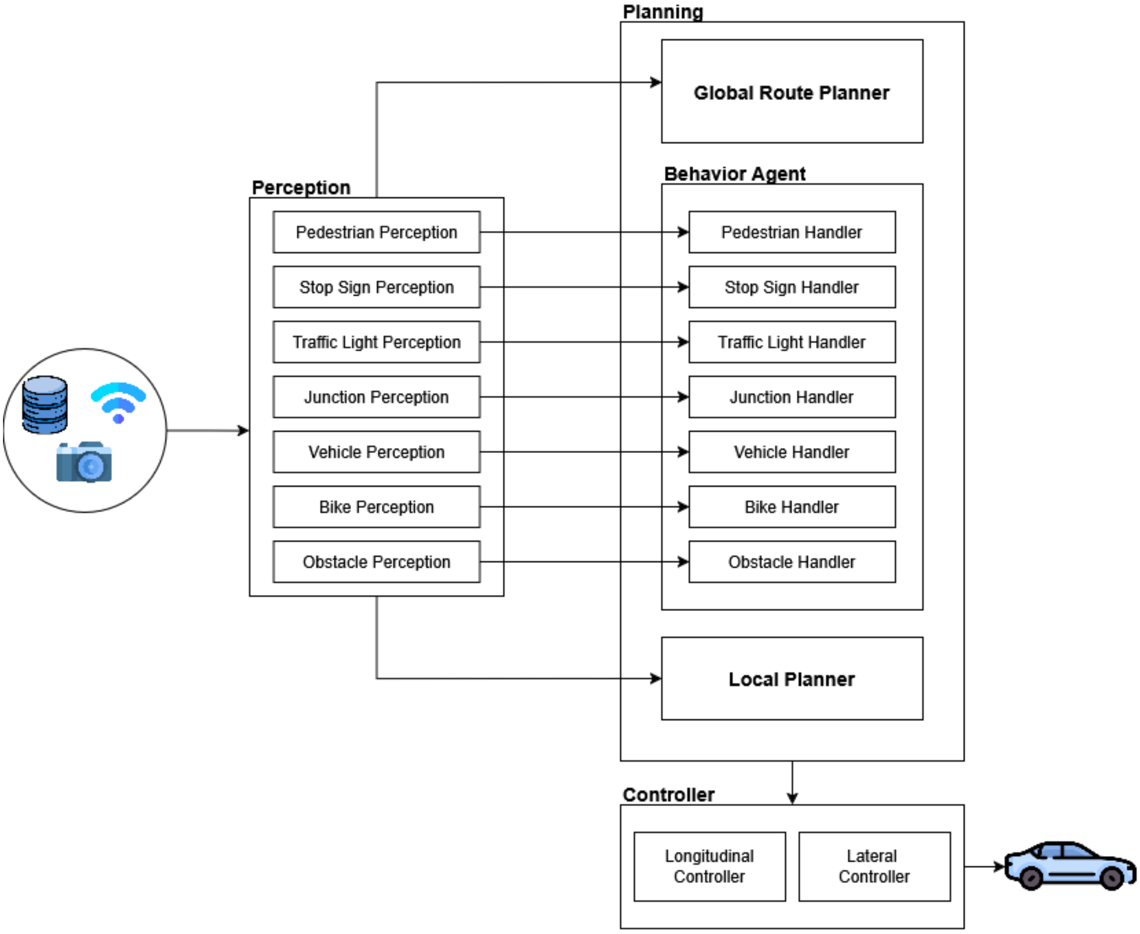



Figure 4.1: High Level System Architecture

# CHAPTER 5

# RESULTS

This chapter presents the final results obtained by the autonomous driving agent across the different routes. The results are analyzed in detail, with a focus on the agent's behavior in terms of safety, efficiency, and correctness of decision-making. Based on the observed results, a critical discussion is provided to highlight the system's limitations and the aspects that still require refinement. Finally, potential future improvements and directions for further development are outlined, aimed at increasing robustness, adaptability, and reliability of the overall system.

## 5.1 Our autonomous agent evaluation

| | **Route0** | **Route1** | **Route4** | **Global** |
|---|---|---|---|---|
| **Driving Score** | 94.72 | 81.90 | 100.00 | 92.20 |
| **Route Completion** | 100.00 | 100.00 | 100.00 | 100.00 |
| **Infraction Penalty** | 0.9472 | 0.8190 | 1.00 | 0.9220 |
| **Infractions** | | | | |
| Collision with pedestrian | 0 | 0 | 0 | 0 |
| Collision with vehicle | 0 | 0 | 0 | 0 |
| Collision with static object | 0 | 0 | 0 | 0 |
| Running a red light | 0 | 0 | 0 | 0 |
| Running a stop sign | 0 | 0 | 0 | 0 |
| Scenario timeout (stuck for 4 minutes) | 0 | 0 | 0 | 0 |
| Failure to maintain minimum speed | 2 | 5 | 0 | 7 |
| Failure to yield to emergency vehicle | 0 | 0 | 0 | 0 |
| Off-road driving | 0 | 0 | 0 | 0 |
| **Total** | 2 | 5 | 0 | **7** |

Table 5.1: Overview of our agent performance on the routes.

> Stopping the route

========= Results of RouteScenario_0 (repetition 0) ------ SUCCESS =========

| | |
|---|---|
| Start Time | 2025-06-27 09:23:36 |
| End Time | 2025-06-27 09:26:15 |
| Duration (System Time) | 158.61s |
| Duration (Game Time) | 48.4s |
| Ratio (System Time / Game Time) | 0.305 |

| Criterion | Result | Value |
|---|---|---|
| RouteCompletionTest | SUCCESS | 100 % |
| OutsideRouteLanesTest | SUCCESS | 0 % |
| CollisionTest | SUCCESS | 0 times |
| RunningRedLightTest | SUCCESS | 0 times |
| RunningStopTest | SUCCESS | 0 times |
| MinSpeedTest | SUCCESS | 128.33 % |
| InRouteTest | SUCCESS | |
| AgentBlockedTest | SUCCESS | |
| Timeout | SUCCESS | |

Figure 5.1: Table results of the Route0.

========= Results of RouteScenario_1 (repetition 0) ------ FAILURE =========

| | |
|---|---|
| Start Time | 2025-06-27 09:30:38 |
| End Time | 2025-06-27 09:57:47 |
| Duration (System Time) | 1628.52s |
| Duration (Game Time) | 507.85s |
| Ratio (System Time / Game Time) | 0.312 |

| Criterion | Result | Value |
|---|---|---|
| RouteCompletionTest | SUCCESS | 100 % |
| OutsideRouteLanesTest | SUCCESS | 0 % |
| CollisionTest | SUCCESS | 0 times |
| RunningRedLightTest | SUCCESS | 0 times |
| RunningStopTest | SUCCESS | 0 times |
| MinSpeedTest | FAILURE | 87.05 % |
| InRouteTest | SUCCESS | |
| AgentBlockedTest | SUCCESS | |
| ScenarioTimeoutTest | SUCCESS | 0 times |
| Timeout | SUCCESS | |

Figure 5.2: Table results of the Route1.

========= Results of RouteScenario_4 (repetition 0) ------ SUCCESS =========

| | |
|---|---|
| Start Time | 2025-06-27 07:41:52 |
| End Time | 2025-06-27 07:52:13 |
| Duration (System Time) | 620.52s |
| Duration (Game Time) | 205.6s |
| Ratio (System Time / Game Time) | 0.331 |

| Criterion | Result | Value |
|---|---|---|
| RouteCompletionTest | SUCCESS | 100 % |
| OutsideRouteLanesTest | SUCCESS | 0 % |
| CollisionTest | SUCCESS | 0 times |
| RunningRedLightTest | SUCCESS | 0 times |
| RunningStopTest | SUCCESS | 0 times |
| MinSpeedTest | SUCCESS | 122.75 % |
| InRouteTest | SUCCESS | |
| AgentBlockedTest | SUCCESS | |
| ScenarioTimeoutTest | SUCCESS | 0 times |
| Timeout | SUCCESS | |

Figure 5.3: Table results of the Route4.

## 5.2 Operational Design Domain (ODD)

The following section describes the specific Operational Design Domain (ODD) of our agent. Compared to the baseline, the system is capable of operating under fewer restrictions, while still ensuring safe and effective behavior in complex urban scenarios.

| Category | Sub-domain | Description of Capabilities and Limitations |
|---|---|---|
| **Scenery** | **Roadway Types** | The system can operate on urban roads with one or two lanes, as well as on rural or suburban roads with a single lane. In two-lane roads featuring tight curves, the vehicle may encroach into the adjacent lane. |
| | **Fixed Road Structures** | The system can detect and obey traffic lights and stop signs. |
| | **Temporary Road Structures** | The system is able to detect temporary static objects such as roadworks, roadside parked vehicles, or temporary warnings. |
| | **Speed Limits** | The system is capable of retrieving the current road's speed limit and maintains its speed below that threshold, but it's not able to always respect the minimum speed limit. |
| **Dynamic Elements** | **Traffic** | The system is able to operate in every traffic conditions (such as traffic jamming and road narrowing). |
| | **Pedestrian Crossing** | The system reliably detects pedestrians positioned along the intended driving path of the vehicle and performs controlled braking in advance to ensure collision avoidance. |
| **Environmental Conditions** | **Illumination** | The system can operate during both daylight and nighttime conditions, handling both good and poor lighting scenarios. |
| | **Weather** | The system can handle cloudy or wet weather conditions. However, in rainy scenarios, it may lose control during emergency braking maneuvers. |

Table 5.2: Our agent Operational Design Domain (ODD) and its limitations.

## 5.3 Conclusions and future improvement

The agent developed in this project showed significantly improved performance compared to the baseline, especially in terms of reducing the most critical and penalizing infractions, such as collisions with pedestrians, vehicles and obstacles, and violations of traffic rules. Nonetheless, some limitations remain, which impact the agent's efficiency and applicability.

The main limitation concerns the **minimum speed requirement**. The agent was designed to operate at a maximum speed of 30 km/h. While this conservative approach enhances safety margins and facilitates obstacle handling, it also results in repeated infractions for failure to maintain the minimum required speed. A future improvement would be on increasing the speed, this implies the need to simultaneously improve the vehicle's trajectory control and navigation accuracy, to avoid compromising safety when operating at higher speeds.

In addition to this, some limitations may arise in specific scenarios: rarely, collision happens during road narrowing or overtaking maneuvers. In these cases, the ego-vehicle may experience minor lateral collisions with vehicles in the opposite lane, especially when available space is limited.

Furthermore, although traffic light handling has been integrated into the system, the routes used for evaluation did not include situations that explicitly tested this functionality. As a result, the agent's behavior in traffic light-controlled intersections remains unverified and requires further dedicated testing.

Finally, the evaluation of the system has been conducted on a limited number of predefined routes (three in total). While the vehicle has demonstrated consistent and reliable performance across all tested routes, there is no certainty that the same level of performance would be achieved on different, untested routes.

# LIST OF FIGURES