ReliablE in-Vehicle pErception and decisioN-making in complex environmenTal conditionS

Grant Agreement Number: 101069614

D4.2 Motion planning

Document Identification			
Status	Final	**Due Date**	30/08/2024
Version	1.0	**Submission Date**	11/11/2024
Related WP	WP4	**Document Reference**	D4.2
Related Deliverable(s)	D4.1	**Dissemination Level**	PU
Lead Participant	TECN	**Document Type:**	Other
Contributors	TECN, TUD, HIT	**Lead Authors**	Leonardo González, (TECN)
		Reviewers	Barys Shyrokau (TUD)
			Fabio Tango (CRF)

Funded by the European Union

This project has received funding under grant agreement No 101069614. It is funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Document Information

Author(s)		
First Name	Last Name	Partner
Leonardo	Gonzalez	TECN
Asier	Arizala	TECN
Oscar	de Groot	TUD
Alberto	Bertipaglia	TUD
Alireza	Ahrabian	HIT

Document History			
Version	Date	Modified by	Modification reason
0.1	15/09/2024	TECN	Empty document with section structure
0.2	15/10/2024	HIT	Included contributions from Hitachi first version
0.3	17/10/2024	TECN	Included contributions from TECN first version
0.4	18/10/2024	TUD	Included contributions from TUD
0.5	25/10/2024	TECN	Included contributions from TECN second version
0.6	01/11/2024	CRF	First review
0.7	01/11/2024	TUD	Second review
0.8	07/11/2024	TECN	Final review
0.9	08/11/2024	ICCS	Final review and plagiarism check
1.0	11/11/2024	ICCS	Final submission

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Leonardo Gonzalez (TECN)	07/11/2024
Quality manager	Panagiotis Lytrivis (ICCS)	08/11/2024
Project Coordinator	Angelos Amditis (ICCS)	08/11/2024

Executive Summary

This document presents the final deliverables for Task 4.1, Motion Planning, within the EVENTS project. It encompasses the finalized designs and implementations of the motion planning module, detailing its development and integration into various experimental setups. Each experiment includes a dedicated description of how the module is adapted to meet its unique requirements. The algorithms and methodologies used for motion planning have been specifically tailored and optimized for each experiment. These are concurrently integrated within both the simulation environment and the real-world platform, forming the core of ongoing work in WP5. Full quantitative outcomes, aimed at validating the module's performance, are anticipated in WP6, which will focus on the evaluation phase and cost-efficient sensor suite optimization. Thus, this report represents a comprehensive overview of progress in Task 4.1.

Table of Contents

List of Tables

List of Figures

Abbreviations & Acronyms

Abbreviation / acronym	Description
ACC	Adaptive Cruise Control
AD(F)	Autonomous Driving (Function)
AI	Artificial Intelligence
AL	Alert Limit
AV	Automated Vehicle
BP	Behavioural Planner
CA	Consortium Agreement
CAM	Cooperative Awareness Message
CAV	Connected Automated Vehicle
CPM	Collective Perception Messages
DDT	Dynamic Driving Task
DENM	Decentralized Environmental Notification Message
DM	Decision Making
EC	European Commission
EXPs	Experiments
FIS	Fuzzy Inference System
FoV	Field of View
FTP	Fail-degraded Trajectory Planning
GA	Grant Agreement
IR	Integrity Risk
ISO	International Organization for Standardization
I/O	Input(s) / Output(s)
LiDAR	Light Detection and Ranging
MDP	Markov Decision Process
MPC	Model Predictive Control
MRM	Minimum Risk Manoeuvre
MOP	Moving Object Prediction

Abbreviation / acronym	Description
MOT	Multi-Object Tracking
ODD	Operational Design Domain
PE	Position Error
PL	Protection Level
PP	Perception Platform
RADAR	RAdio Detecting And Ranging
REQs	Requirements
RL	Reinforcement Learning
SAE	Society of Automotive Engineers
SMD	Safety-mode Decision
SoTA	State of the art
SPaT message	Signal Phase and Timing message
SPECs	Specifications
TOR	Take Over Request
TP	Trajectory Planner
TSs	Target Scenarios
UCs	Use Cases
VRU	Vulnerable Road User
WP	Work Package

1. Introduction

1.1 Project aim

Driving is a challenging task. In our everyday life as drivers, we are facing unexpected situations we need to handle in a safe and efficient way. The same is valid for Connected and Automated Vehicles (CAVs), which also need to handle these situations, to a certain extent, depending on their automation level. The higher the automation level is, the higher the expectations for the system to cope with these situations are.

In the context of this project, these unexpected situations where the normal operation of the CAV is close to being disrupted (e.g., ODD limit is reached due to traffic changes, harsh weather/light conditions, imperfect data, sensor/communication failures, etc.), are called “events”. EVENTS is also the acronym of this project.

Today, CAVs are facing several challenges (e.g., perception in complex urban environments, Vulnerable Road Users (VRUs) detection, perception in adverse weather and low visibility conditions) that should be overcome to be able to drive through these events in a safe and reliable way.

Within our scope, and, to cover a wide area of scenarios, these kinds of events are clustered under three main use cases: a) Interaction with VRUs, b) Non-Standard and Unstructured Road Conditions and c) Low Visibility and Adverse Weather Conditions.

Our vision in EVENTS is to create a robust and self-resilient perception and decision-making system for AVs to manage different kinds of “events” on the horizon. These events result in reaching the AV ODD limitations due to the dynamically changing road environment (VRUs, obstacles) and/or due to imperfect data (e.g., sensor and communication failures). The AV should continue and operate safely no matter what. When the system cannot handle the situation, an improved minimum risk manoeuvre should be put in place.

1.2 Deliverable scope and content of the Document

Task 4.1 of the EVENTS project is dedicated to motion planning, encompassing the design and integration of algorithms to generate trajectory references for both longitudinal and lateral control. This document reports the final developments in Task 4.1, focusing solely on motion planning, which will be implemented and tested across selected experiments in the project. Results and quantitative analyses are planned in future deliverables.

In this document, not every experiment is discussed in the project EVENTS. After all, the development of both motion and behavioural planning depends on each

experiment's focus. Motion planning will be developed and tested for EXP1, EXP2, EXP4, and EXP8.

The document is structured as follows:

- Chapters 2 to Chapter 5 go in-depth into the developments of each experiment regarding the motion planning module.
- Finally, Chapter 6 presents the conclusions of the progress mentioned in the previous chapters as a summary of the document.

1.3 Motion Planning

Motion planning is a critical component of Automated Vehicles (AV). As AVs move through dynamic and unpredictable environments, their ability to plan and execute precise paths is essential for maintaining safety and efficiency. The core objective of the motion planning system is to create a feasible and comfortable trajectory that the vehicle's control system can follow. This planning relies on input from perception, communication, and internal acquisition and self-assessment modules, as illustrated in the EVENTS architecture (Figure 1). This architecture has served as a reference throughout the whole EVENTS project and has guided the interaction between different modules developed within the project.

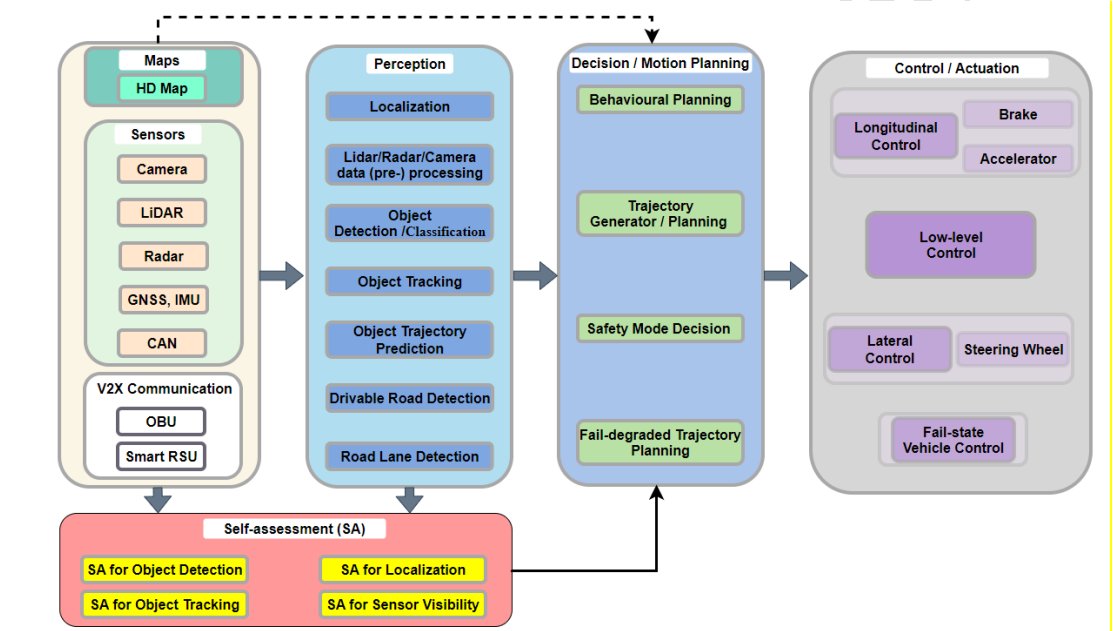


Figure 1 EVENTS high-level Architecture and Interfaces ("Master Architecture") [1]

The motion planning framework proposed in EVENTS addresses the complex challenges of autonomous navigation through different methods for each of the experiments selected. Optimization methods play a key role, where the motion planner formulates each movement as a local optimization problem, minimizing a cost function that balances road-following accuracy, comfort, obstacle avoidance, and steering smoothness. The vehicle dynamics are modelled with a jerk-controlled kinematic bicycle model, ensuring smoother and more comfortable trajectories by penalizing high jerks and rapid wheel angle changes. Several cost components contribute to this objective: a curvature-aware contouring cost to track the reference path, an adaptive velocity term to handle sharp turns, a repulsive force for obstacle avoidance, and a Huber loss to maintain robustness against outliers.

Various trajectory generation methods were evaluated, such as Bézier curves, splines, and optimization-based approaches, each offering unique advantages. Quintic Bézier curves were emphasized for their smoothness and adaptability, while splines provided a cost-effective solution for trajectory smoothing. An optimization-based method utilizing a kinematic Ackermann model was implemented within a Model Predictive Control (MPC) framework to enhance trajectory reactivity, particularly in slippery conditions. Performance indicators were established to quantify the trade-offs between accuracy, comfort, and computational efficiency, focusing on vehicle dynamics, trajectory adherence, and responsiveness.

In a complementary effort, high-definition (HD) map updates were proposed based on detected roadwork bollards. This method seeks to enhance real-time map accuracy by integrating data from camera and GPS sources to inform motion planning. The specific operational design domain (ODD) was defined as a two-lane road with one lane blocked by roadworks. This lane definition was later used as input for one of the trajectory generation methods developed for the project.

2. EXP1 (TUD)

Experiment 1 (EXP1) focuses on the "Interaction with Vulnerable Road Users (VRUs) in a Complex Urban Environment". The objective is to achieve safe, comfortable, and time-efficient automated driving in a complex urban environment while engaging with VRUs such as pedestrians and cyclists.

2.1 SoTA

The state-of-the-art methods for motion planning were discussed in D4.1 “Initial version of motion planning and behavioural decision-making components”. Based on the analysis, the optimization-based approach has been selected.

The implementation of the optimization-based planner, path tracking is accomplished with Model Predictive Contouring Control (MPCC) [2], which introduces a cost function that allows the vehicle to follow the road (i.e., the reference path) and stay within the road boundaries. However, in the study [3], it was noted that the default formulation (e.g., as used in [2]) does not consider the curvature of the reference path. Therefore, the following issues have been noted for the improvement:

1. When tracking a velocity, that velocity is not in the direction of the path, possibly resulting in speeding when overtaking.
2. A planned trajectory could violate the road boundary constraints while giving back a feasible solution.

To resolve these issues, the proposed planner uses Curvature-Aware MPC to resolve these problems in favour of MPCC to improve planning performance.

2.2 Architecture

The proposed motion planner is integrated with Autoware to replace the default scenario planner, as shown in Figure 2. It uses the existing mission planner to plan a high-level route from the vehicle position to the user-specified target position. The high-level route is processed to provide path information for the motion planner. The trajectory computed by the motion planner is passed to the control module. This control module then computes control inputs to track the trajectory as closely as possible.

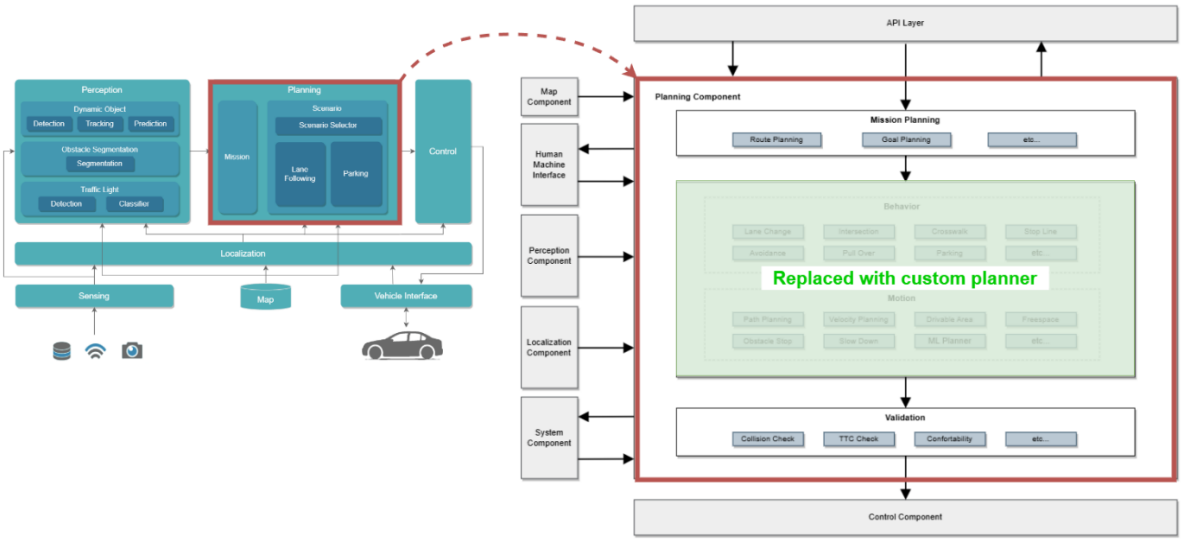


Figure 2 The motion planner replaces the scenario planner in the Autoware ROS stack.

The architecture of the motion planning and decision-making components is shown in Figure 2, in which the motion planning without behavioral decision-making is represented in the local planner step (marked in blue) in which P collision-free trajectories are planned in parallel. A high-level guidance planner computes global guidance trajectories. Each guidance trajectory is passed to a set of local planners that further refine the trajectories and ensure they are dynamically feasible. The executed trajectory is determined by comparing the locally planned trajectories. The motion planning part (local planning) is discussed in this deliverable.

The local planners are implemented with Model Predictive Control. Each local planner solves a nonlinear MPC problem in real time, computing a locally optimal trajectory.

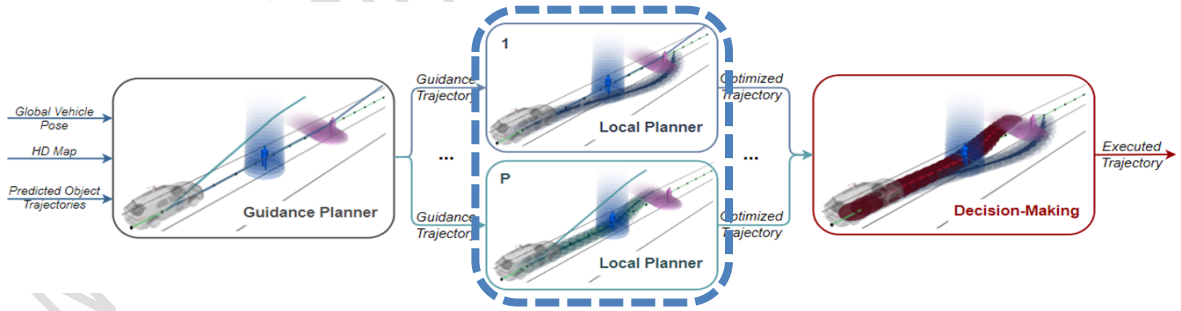


Figure 3 Motion planning architecture

2.3Algorithmic Approach

The generic formulation of the optimal control problem is presented in D4.1, "Initial version of motion planning and behavioural decision-making components". For automated driving, each of the local optimisation problems that the motion planner solves online is given by:

$$\begin{aligned} &\min_{u\in U, x\in X} \sum_{k=0}^{N_t} J(x_k, u_k) \\ &\text{s.t.} \quad x_{k+1} = f(x_k, u_k), \forall k \\ &\qquad x_0 = x_{init} \\ &\qquad g(x_k, o_k^j) \le 0, \forall k, j \end{aligned} \tag{2.1}$$

where J represents the objective to be minimised, f is the vehicle model and g represents the collision avoidance and road boundary constraints.

The vehicle model is a jerk-controlled kinematic bicycle model with control inputs. With l_r, l_f the distances from the front and rear axles to the centre of gravity of the vehicle, δ the wheel road angle, x, y the vehicle position, v its forward velocity, θ the heading angle, a, j the acceleration and jerk and ω the road wheel angle rate, the vehicle model is given by

$$\beta = \tan^{-1}\left(\frac{l_r}{l_r + l_f} * \tan(\delta)\right)$$

$$\dot{x} = v\cos(psi + \beta), \dot{y} = v\sin(psi + \beta) \tag{2.2}$$

$$\dot{\theta} = \frac{v}{l_r}\sin(\beta)$$

$$\dot{v} = a, \dot{a} = j, \dot{\delta} = \omega.$$

Changes in jerk are known to result in discomfort. By controlling the jerk, penalizing high jerks leads to smoother and more comfortable trajectories. The rate of road wheel angle is penalised to make steering input sufficiently smooth.

The objective consists of the following components:

Road Following (Curvature-Aware Contouring [3]). This cost function follows a reference path $[0, S] \to \mathbb{R}^2$ (mapping distance to x and y positions) and it aims to track a reference velocity in the direction of the path. The cost is computed as:

$$J_{CA-MPC} = q_s(\dot{s_k} - \dot{s_t})^2 + q_c(\epsilon_k^c)^2, \tag{2.3}$$

where $\dot{s_k}$ and $\dot{s_t}$ are the current and target velocity in the direction of the path, ϵ_k^c is the curvature-aware contouring error, and q_s and q_c are tuning parameters. Compared to MPCC, CA-MPC has one less tuning parameter, as no lag error is required. The complexity in the implementation of CA-MPC is in the computation of $\dot{s_k}$, that considers the curvature of the reference path. The reference path is obtained from the lanelets in the path of Autoware's default mission planner. It is processed to form a sequence of cubic splines that are smoothed together, ensuring first-order continuity of the reference path. In addition to the stage-wise contouring cost, a

terminal cost is added on the deviation of the vehicle orientation from the direction of the reference path and on the regular contouring costs.

Adaptive Reference Velocity: The term $\dot{s}_t$ in our implementation is adaptive in that we define it as a function $[0, S] \rightarrow \mathbb{R}^+$, that maps distance along the reference path to a velocity. This, for example, allows the vehicle to stop at the end of the path and can accommodate for tight corners by scaling back the reference velocity.

Huber Loss (a loss function used in robust regression, that is less sensitive to outliers in data than the squared error loss): Instead of quadratic penalties (e.g., x^2) we formulate our cost function regarding the path reference and velocity as the Huber loss of the component. The Huber loss is only quadratic up to a threshold and linear after this, preventing any cost from dominating the behaviour of the vehicle by reducing the costs that are far from the reference (e.g., preventing aggressive acceleration from standstill).

Obstacle Repulsive Forces: To encourage the vehicle to keep its distance from VRUs, we include this cost term:

$$J_{repulsive} = q_r \sum_{i=1}^{N} \frac{1}{(\Delta x_k^i)^2 + (\Delta y_k^i)^2 + \gamma}, \tag{2.4}$$

where $\Delta x_k, \Delta y_k$ represent the distances to each of the obstacles (superscript i) and γ is a constant for numerical stability. Parameter q_r weights how strong the vehicle moves out from obstacles.

Acceleration, Jerk and Rotational velocity penalties: Comfort is promoted by the cost term:

$$J_{input} = q_a a^2 + q_\omega \omega^2 + q_j j^2, \tag{2.5}$$

where a, ω and j are the acceleration, yaw rate and jerk, and q_a, q_w, q_j represent their weights respectively.

Slack penalty: For ensuring feasibility, some constraints can be violated by a fixed amount and the resulting violation d is penalized with a cost:

$$J_{slack} = q_{slack} d \tag{2.6}$$

The constraints consist of the following components:

Probabilistic dynamic obstacle avoidance: Obstacles and their future motion are predicted with a Gaussian probability distribution for each time instance. In this probabilistic context, a collision is a probabilistic event that happens with a probability. The proposed motion planner constrains the probability of collision $\epsilon, 0 \leq \epsilon < 1$, that is, the probability that a collision happens at any point in its trajectory.

These constraints are imposed using Chance Constrained MPC [4]. Modelling the vehicle with several discs and each obstacle as a single disc, a linear constraint $A_k p_k \leq b_k$ can be formulated to describe how far from the obstacle the vehicle discs can be. The cchance-constrainedMPC imposes the constraints:

$$A_k^{\top}(\Delta p_k) - r \geq \mathrm{erf}^{-1}(1-2\epsilon)\sqrt{2A_k^{\top}\Sigma_k A_k^{\top}}, \tag{2.7}$$

With Δp_k the difference in position between the obstacle and the vehicle disc, Σ_k the variance of the distribution and $\mathrm{erf}^{-1}(\cdot)$ the inverse standard error function. These constraints ensure that the probability of collision at time k is smaller than ϵ and are imposed for each obstacle, each vehicle disc and each time step.

Homotopy class constraints: To ensure that the homotopy class of the global guidance trajectory is respected (i.e., the direction in which it passes obstacles) an additional constraint is inserted per obstacle. This constraint is of the form $A_k p_k \leq b_k$, but where p_k is taken from the guidance trajectory at the time step k and where the obstacle is given zero radius as the probabilistic collision avoidance constraint is responsible for collision avoidance. An example is visualised in Figure 4.

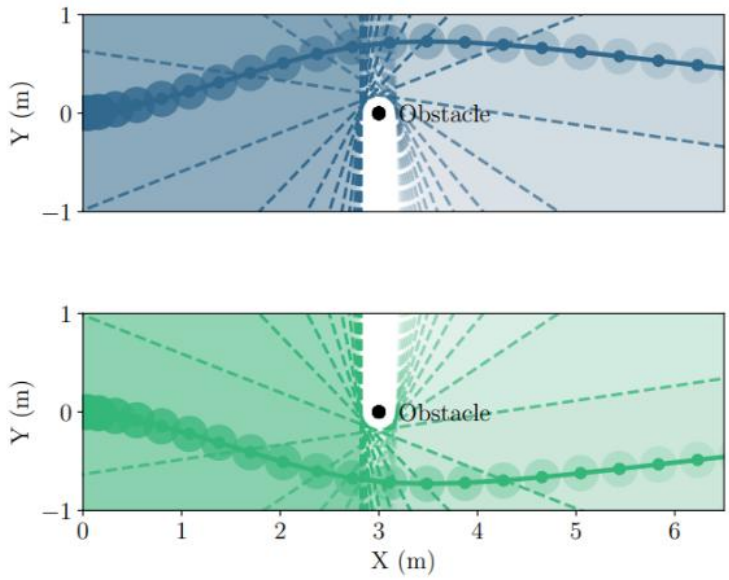


Figure 4 Example of two locally optimized trajectories (blue / green)

In Figure 4 each trajectory uses a homotopy distinct global guidance trajectory to construct a set of homotopy class constraints. This example considers a static obstacle and indicates that the homotopy constraints do not allow the vehicle to pass the obstacle on the other side.

Road boundary constraints: The vehicle is kept within the road boundaries by constraining the curvature-aware contouring error as:

$$b_{l,k}(s_k) \leq \epsilon_k^c \leq b_{r,k}(s_k), \tag{2.8}$$

where $b_{l,k}$ and $b_{r,k}$ represent the road width on the left and right side, respectively, as a function of the current position of the vehicle along the path. These functions are obtained by first retrieving the lanelets from Autoware's mission planner. For each point on the reference path, a representative point on the boundary is computed. Over these points, a set of cubic splines are fitted and passed to the optimisation problem at runtime.

State and input constraints: Finally, states and inputs are constrained to match their physical application.

The road boundary and collision avoidance constraints are formulated with a slack variable that allows small violations ($d \leq 0.1$) that allows the optimisation to remain feasible when the perception or control components deviate from their expected behaviour.

Note that, using the global guidance from behavioural decision-making, the planner usually solves multiple instances of the local optimisation problem concurrently, ultimately comparing their solutions.

The overall planner parameters are listed in Table 1 Planner parameters, values and their description

.

Parameter	Value	Description
q_s	0.075	Contouring weight
q_c	0.015 (terminal: 0.075)	Velocity tracking weight
q_r	0.05	Obstacle repulsion weight
q_a	0.2	Acceleration weight
q_ω	0.05	Rotational velocity weight
q_j	0.01	Jerk weight
q_{slack}	10	Slack weight
$\dot{s}_t$	(Adaptive, default: 2.5m/s)	Reference velocity
γ	10^{-5}	Numerical stability constant (for obstacle repulsion)
ϵ	0.05 (5% risk)	Probability of collision
$n_{segments}$	5	Number of cubic splines used for optimization
$n_{obstacles}$	4	Number of obstacles
N	35	Planning horizon
Δt	0.2s	Time between discrete time steps of the prediction
T	7.0s	Planning time horizon
t_{loop}	$0.1s$	Planning time

Table 1 Planner parameters, values and their description

The MPC planner is implemented within the Autoware framework, using ROS and C++. The optimization problem is solved with Forces Pro [5]. The implemented software first generates C++ code from a python-based problem description. This generated C++ code is used by the developed C++ software that solves the planning problem using real-time data from the vehicle.

To define good initial guesses for the parameters in Table 1 Planner parameters, values and their description

, high-level performance indicators are optimised with an open-source optimisation framework, Optuna, that implements hyperparameter tuning with Bayesian optimisation. The resulting weights are then manually adjusted to fine-tune the motion planner.

The qualitative behaviour of the developed motion planner is illustrated in Figure 5. The first figure shows that the planner allows the vehicle to corner smoothly (top left), following the reference path. In the second figure (top right), the vehicle stays within the road boundaries and avoids a pedestrian, thanks to the constraints. The bottom figures show that the vehicle performs the scenario safely and efficiently.

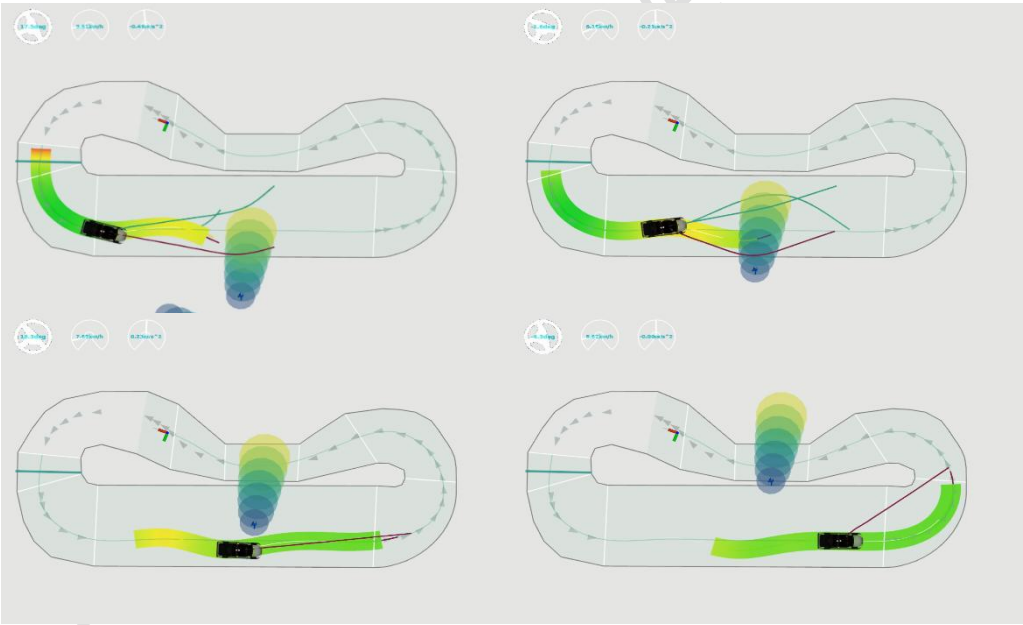


Figure 5 Behaviour of the planner in a scenario with one crossing pedestrian.

In the Figure 5 the green / yellow area denotes the planned and past trajectory of the vehicle, with yellow (slow) to green (fast) marking speeds. Pedestrian motion is predicted forward in time with a Gaussian distribution illustrated with blue-yellow circles. In the top images, two trajectories are optimized (green/red short lines) that pass the obstacle in different ways. The red trajectory is executed (it has a lower cost), causing the vehicle to pass behind the pedestrian.

2.4 Future work

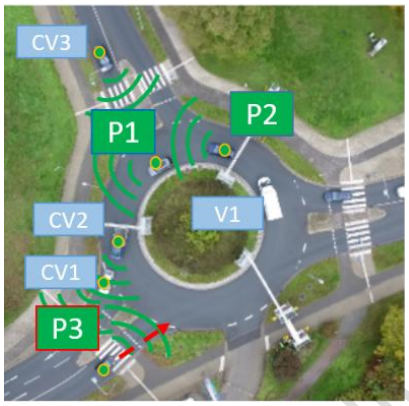
The next step is quantitively validating the proposed motion planner in the Autoware simulation and comparing its performance against the default Autoware planner. Validation metrics are focused on safety, time efficiency and comfort. This quantitative evaluation will be included in D6.2 “Technical evaluation results”.

3. EXP2

In the EXP2, a platoon formed by a leader and two follower vehicles is split due to the interruption of an outside vehicle in a roundabout. The goal is to reconnect the platoon again so they can cross the roundabout together. Therefore, the leader may decide that it must drive around the roundabout again, changing the initial path calculated by the global planner. This leads to the conclusion that followers, if they want to keep being part of the platoon, cannot have a static trajectory generation.

In this case, the trajectory generation of the follower vehicles is defined by the position of the leader that they receive through communication. However, the trajectory created will not be smooth or easy to follow.

In this task, we want to know if a dynamic trajectory generation could improve the comfort measured as the lateral acceleration and, if so, what the cost is.



- Green dot denotes V2X capability of the traffic agent

P1: CAV platoon leader
P2: CAV platoon follower #1
P3: CAV platoon follower #2
(←this is the subject vehicle which tries to reconnect with the platoon via merging into the roundabout right after/before P1, P2, details of P1, P2, P3 choreography so that a platooning reconnection is realized to be specified later)
CV1, CV2, CV3: Connected vehicles able to share CAM, DENM, CPM info
V1: not connected vehicle

Figure 6 The ODD being considered in EXP 2, where three vehicles in a platoon

In the next sections the simulation conditions, each trajectory in the benchmark used and the metrics will be explained in detail.

3.1 Trajectory test simulation conditions

First of all, we need to define some static parameters for every experiment. The scenario is the Town03 of the standard map set of CARLA simulator; this scenario has been chosen as the better fit in representing urban environments among the default maps provided by the CARLA simulator developers; including signalized and unsignalized intersections, as well as roundabouts, which are the focus on EXP2. Here we have defined a path with variable curvature that can be seen in the Figure 7, the starting point being in A and the ending point being in B.

The vehicle model chosen for this test is the Tesla Model 3 that is available in the Carla blueprint library.

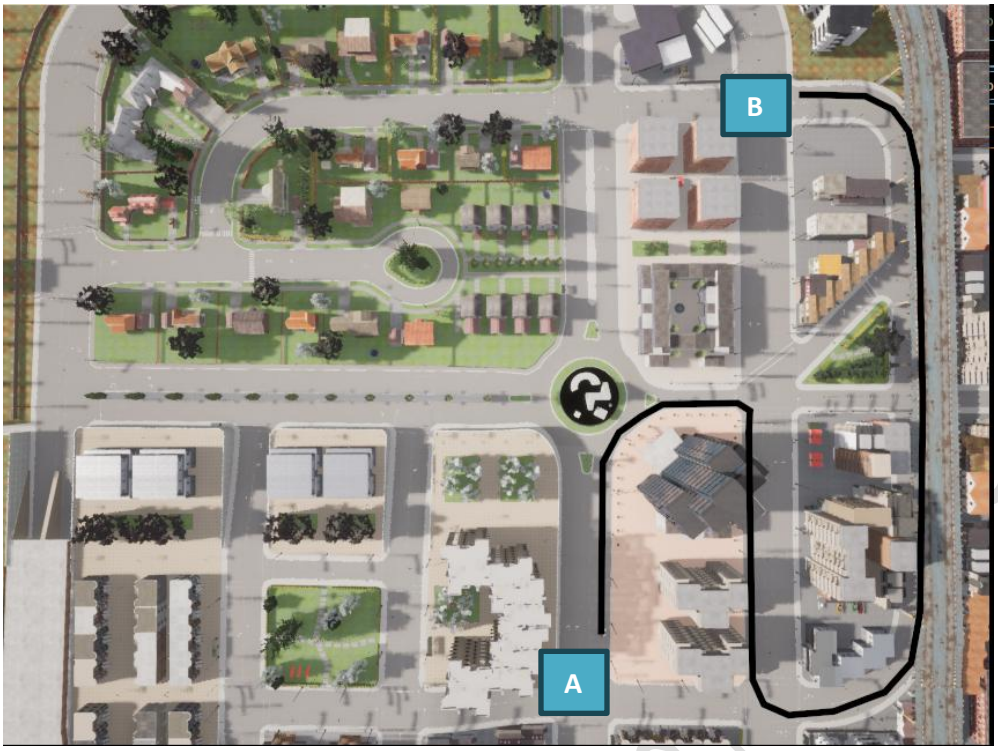


Figure 7 Trajectory to be performed by the vehicle

Since the objective is to study the impact of the dynamic trajectory generation in the lateral accelerations the speed set point will be fixed, in this case, to 8.3 m/s, which is the mandatory speed in urban road with a single track per driving direction in Spain. The longitudinal controller will be the same. A PID that uses speed error as input and throttle and brake values as output. PID values are *Kp=1.0, Ki=0.03* and *Kd=0.05.*

Every vehicle will share the lateral controller too. In this case, a simple PID is applied to the lateral error calculated between the closes point in the trajectory and a control point at 2.5 m in front of the rear axle of the vehicle. PID values in this case are *Kp=1.0, Ki=0.07* and *Kd=0.5.*

The length of every trajectory will be fixed to 40m.

3.2 Trajectory benchmark

This benchmark is composed by Bezier curves, splines, and optimization-based trajectories. In the Figure 8 a representation of every trajectory tested for a single state of the vehicle is presented.

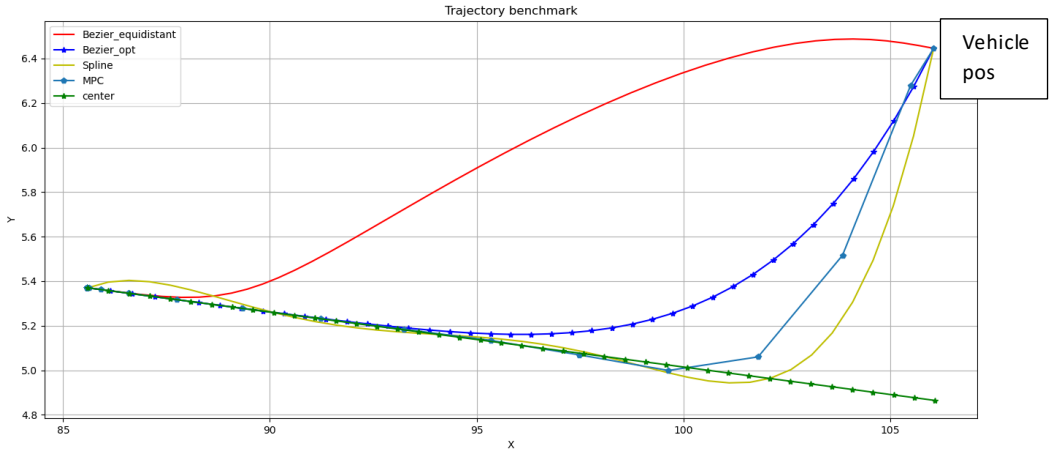


Figure 8 Trajectory benchmark applied to the same state

3.2.1 Bezier curves

Bezier curves are known for three interesting properties:

- The possibility of making it tangent in the beginning and in the end of the curve to two different directions.
- Being a curve contained in a convex shape.
- The fact of being derivable two times, ensuring a continuous curvature alongside the curve.

These curves are defined by the following equation:

$$B(t) = \sum_{i=0}^{n} b_{i,n}(t)\, P_i \quad , 0 \leq t \leq 1 \tag{3.1}$$

Where P_i are the control points and $b_{i,n}$ is the known Bernstein basis polynomial of the order n.

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad , i = 0, \ldots, n \tag{3.2}$$

The problem to be solved in formulating these types of curves, therefore, lies in the definition of the control points. Depending on the application, some choose to calculate them using geometric methods, as in the case of [6], where control points are used as boundaries to avoid collisions with other vehicles, or in [7], where the control points are defined based on the vehicle's speed to perform lane changes. On the other hand, [8] considers it as an optimization problem that may aim to minimize the derivative of the curvature, the second derivative of the curvature, or even a weighted sum of these.

This comparison aims to evaluate several methods found in the state of the art.

3.2.1.1 Equidistant points

One method for defining trajectories for lane changes in autonomous driving is to define a total of 6 control points to form 5th-order Bézier curves: 2 points in front of the vehicle and 2 points behind the endpoint, tangent to the trajectory. An example is shown in Figure 9. This approach allows the control of the curve's aggressiveness by adjusting the distance between the parallel points. It is a straightforward method for defining control points, with a computational cost that should be much lower than solving optimization problems that use iterative methods.

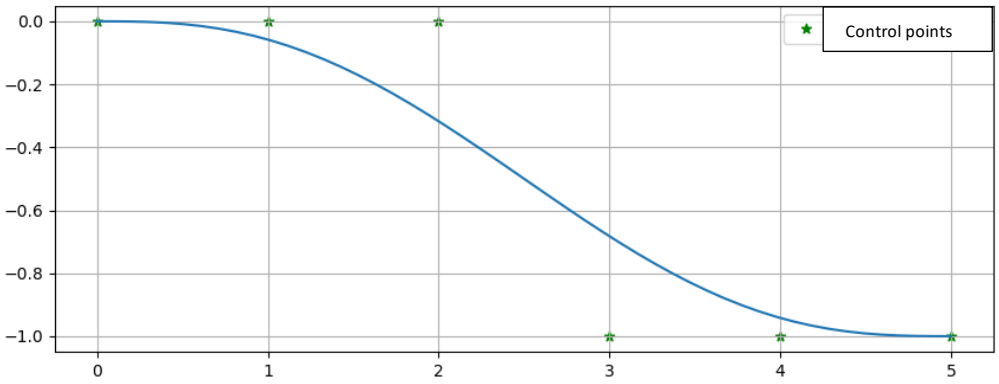


Figure 9 Quintic Bezier curve with equidistant points

This comparison aims to assess whether using this method dynamically for trajectory tracking is feasible. To do so, the distance between points is defined by two factors: a fixed distance and a dynamic one.

$$d_s = d_{fix} + d_{dyn} \tag{3.3}$$

Where d_{fix} in this case is 0.2 m and d_{dyn} can be calculated in three different ways. First, proportional to speed, second proportional to the lateral acceleration and third a pondered sum of both, speed, and lateral acceleration

$$d_{dyn} = K_v v \tag{3.4}$$

$$d_{dyn} = K_a a_{lat} \tag{3.5}$$

$$d_{dyn} = K_v v + K_a a_{lat} \tag{3.6}$$

In Figure 10 the representation of this trajectory can be seen integrated in a ROS environment running in parallel with the CARLA simulator.

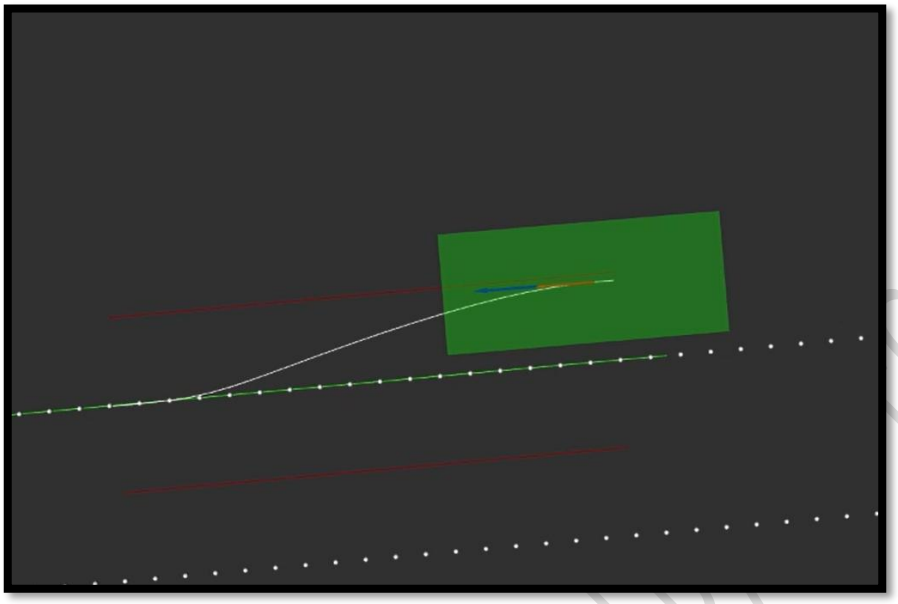


Figure 10 Bezier curve with equidistant control points on CARLA environment

3.2.1.2 Curvature optimization

We are going to implement a method based on the one shown in the work [8] which proposes the calculation of the control points as an optimization problem that minimizes the derivative of the curvature. Although the methods shown in the work use two consecutive optimization processes, we are going to use a single one due to the high computational time shown in the cited article.

The curvature in a quintic Bezier curve $c(t) = [x(t), y(t)]$ is defined by the following function:

$$\kappa = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\sqrt{(\dot{x}(t)^2 + \dot{y}(t)^2)^3}} \tag{3.7}$$

Based on the results of that work we chose to use the following cost function:

$$J = \int_{s_0}^{Sf} \dot{\kappa}(s)^2 ds \tag{3.8}$$

And lateral distance constrains to ensure the trajectory is within the road.

In the Figure 11, in white, there is an example of the Bezier curve calculated by using the curvature optimization method in the CARLA environment.

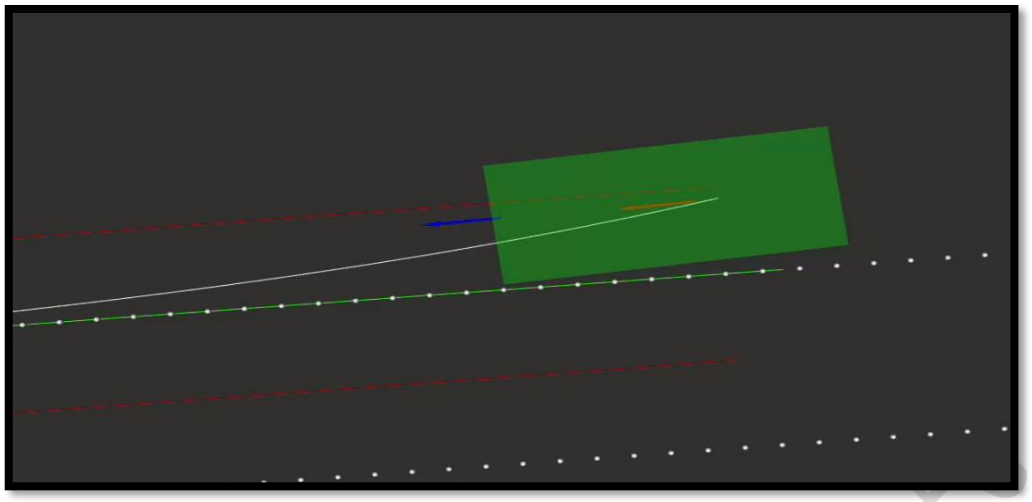


Figure 11 Bezier curve with curvature optimization method in CARLA environment

3.2.2 Splines

Splines are functions defined piecewise by polynomials. In automated driving, they are frequently used for trajectory generation, either by directly generating the trajectory or by smoothing other planning aspects, such as speed profiles or trajectory curvatures. Their strength it bounded by their low computational cost as well as their simplicity. In our case, we are going to use them to generate the trajectory itself.

Additionally, if we are trying to smooth a base path (the reference path in our case) it is possible to define how precise the spline should be.

In this case, we are going to test splines of different degrees as well as splines with different smoothing rate.

In Figure 12, in white, a trajectory generated using the spline approximation can be seen.

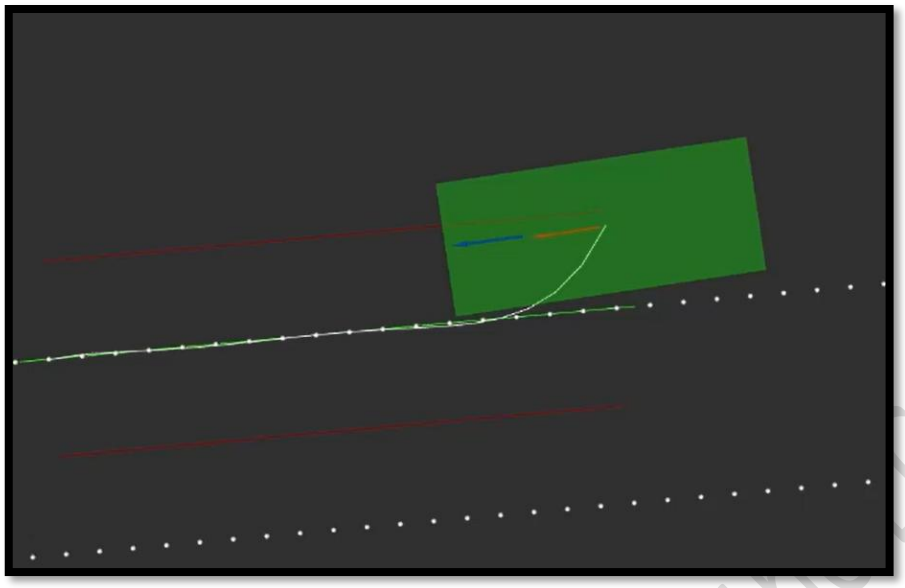


Figure 12 Spline-based trajectory on CARLA environment

3.2.3 Optimization based trajectories.

Model predictive control (MPC) is a well-known control method. However, it is possible to use it in this case to generate trajectories that consider the dynamics of the vehicle alongside other constraints. In our case, the model used will be the kinematic vehicle model (Figure 13), also known as the Ackermann model:

$$\dot{x} = v\cos(\varphi + \beta) \tag{3.9}$$

$$\dot{y} = v\sin(\varphi + \beta)$$

$$\dot{\varphi} = v\cos(\beta)\frac{\tan(\delta)}{L}$$

$$\beta = arctan\left(\frac{l_r\,\delta}{L}\right)$$

Were x and y are the coordinates in the reference system of the vehicle, v is the longitudinal speed, φ is the orientation of the vehicle, β is the angle between the rear axle and the control point measured on the center of the turning center (calculated as if the vehicle had 100% Ackermann configuration) and L and l_r are the wheelbase and the distance from the rear axle to the control point respectively. δ is the steering angle translated to the center of the front axle.

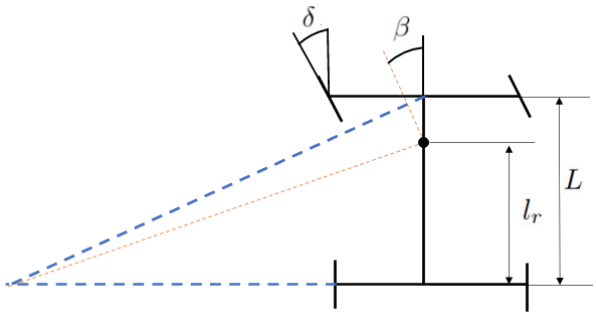


Figure 13 Kinematic model

The cost function is defined to minimize the distance between the trajectory and the reference path, as well as the control value change, which in this case would be the steering of the vehicle. Therefore, the function can be represented by the following equation:

$$J = \sum_{i=0}^{h}(e^2Q + \Delta u^2R) \quad (3.10)$$

Where Q and R are the values that we are going to change in the benchmark.

Additional constrains to the problem consider distances to the left and right borders of the road as well as steering minimum and maximum.

In Figure 14, in white, the trajectory generated by the optimization method can be seen.

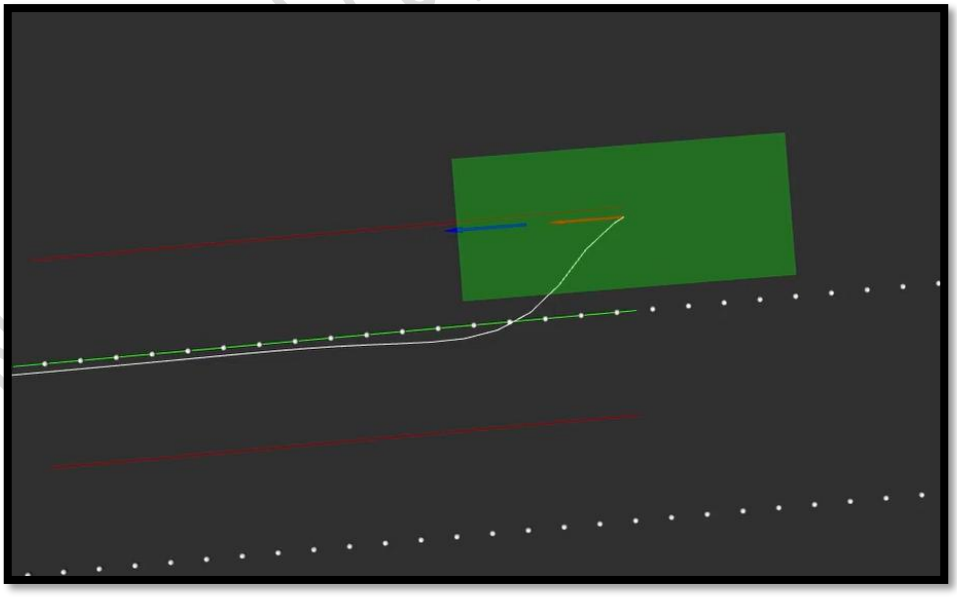


Figure 14 Model-based trajectory generation on CARLA environment

3.3 Comparison metrics

To measure the performance of the vehicle as well as the quality of the trajectories generated, we have established four groups of metrics or Performance Indicators (PI).

Metric	Description	Unit
a_{lat_max}	Maximum lateral acceleration	m/s^2
a_{lat_avg}	Average lateral acceleration	m/s^2
e_{lat_avg}	Average lateral error to traj	m
e_{ang_avg}	Average angular error to traj	rad
$e_{lat\_avg\_cent}$	Average lateral error to center	m
$e_{ang\_avg\_cent}$	Average angular error to center	rad
e_{lat_max}	Max lateral error to traj	m
e_{ang_max}	Max angular error to traj	rad
$e_{lat\_max\_cent}$	Max lateral error to center	m
$e_{ang\_max\_cent}$	Max angular error to center	rad
e_{avg}	Average error traj to center	m
e_{max}	Max error traj to center	m
k_{norm_max}	Normalized max curvature	$1/m$
k_{norm_avg}	Normalized average curvature	$1/m$
ek_{norm_max}	Normalized max curvature error	$1/m$
ek_{norm_avg}	Normalized average curvature error	$1/m$
t_{avg}	Average computation time	s
t_{max}	Max computation time	s

Table 2 Performance metrics

3.3.1 Dynamic performance indicators

Here, in the blue section of Table 2, we gather the PI related to vehicle dynamics. In this case, we consider the maximum and average lateral acceleration during the simulation (a_{lat_max} and a_{lat_avg}). Although comfort is a subjective matter, there are some studies that have tried to measure it. In the work [9] lateral accelerations between 3.1 and 7.5 m/s^2 are considered very uncomfortable. Therefore, one of the objectives will be to find a method that makes the vehicle perform under those values.

3.3.2 Vehicle error performance indicators

Here, we consider how the vehicle was able to follow the dynamic and global trajectories. For that, we will consider the maximum and average angular and lateral error to the dynamic trajectory (e_{lat_avg}, e_{ang_avg}, e_{lat_max} and e_{ang_max}) and the global path ($e_{lat\_avg\_cent}$, $e_{ang\_avg\_cent}$, $e_{lat\_max\_cent}$ and $e_{ang\_max\_cent}$). Therefore, we have a total of 8 PIs (lines in red of the Table 2).

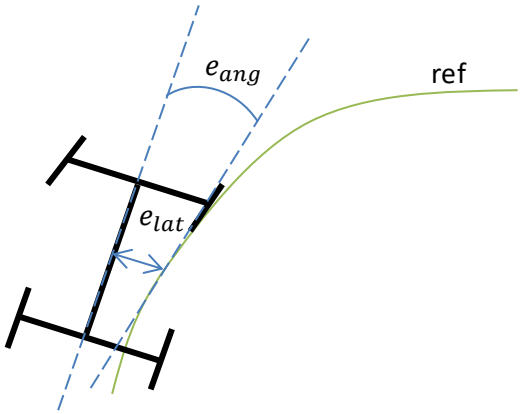


Figure 15 Lateral and angular error representation

3.3.3 Geometry error performance indicators

Here, we check how fit the trajectory is. First, we want to know the maximum and average curvature of the dynamic trajectories along the simulation. Then we have four PIs destined to compare the dynamic trajectories to the global path. We want to check the curvature difference between each trajectory and its static counterpart by measuring the maximum and average curvature difference. Also, we will compare the average and maximum distance between the dynamic trajectory and the global path (lines in green in the Table 2).

3.3.4 Time Performance indicators

Time performance indicators are a recursive study point when it comes to trajectory generation. After all, they define how often a new trajectory can be calculated and, therefore, how reactive the vehicle can be. An ideal dynamic trajectory generator should publish the trajectory at a higher frequency than the control. However, the computation demand is higher most of the time, so the usual approach is to either combine both trajectory generation and control (using an MPC, for example) or to increase the length of the path. This last approach can make the vehicle better at predicting the behavior of its surroundings, but it would lose maneuverability.

The PI considered here will be the average and maximum computation time (the purple lines in the Table 2) of each trajectory generation method tested. These values are measured from the receiving of the global trajectory until the dynamic trajectory is published in a ROS topic.

4. EXP4

4.1Introduction

The objective in EXP4 is to perceive and control a vehicle in the context of an unstructured road use case (specifically road works). The experiment first revolves around perceiving the environment to update the high-definition map. This information is then used by the motion planning module to follow a safe trajectory. The ODD considered in EXP4 is depicted in Figure 16, in which given a two-lane road, a single lane is blocked by traffic bollards, thus, the lane structure is modified according to the position of the bollards.

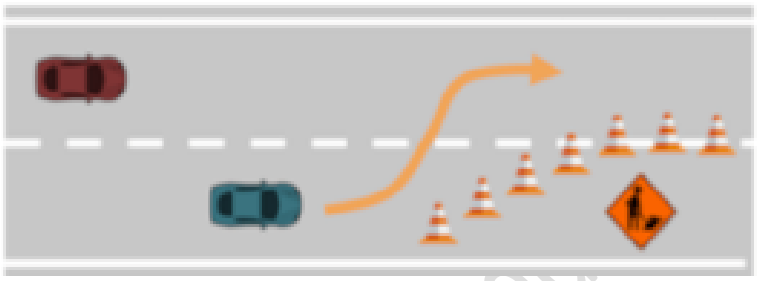


Figure 16 The ODD being considered in EXP4

4.2 HD-Map Update Using Detected Bollards

Our proposed system seeks to update the high-definition map in real-time based on the detected road work bollards in the scene. By updating the HD-map in real-time, we propose to enable more efficient motion planning that can constrain more effectively the candidate trajectories.

To this end, we propose specifically to update the HD map according to the observed roadworks bollards (using data obtained only from the camera and GPS). Our approach will assume a specific ODD, namely, a 2-lane road with one lane being blocked by road works. Our workflow is shown in Figure 17 and is composed of the following steps: 1) 2D detection of road work bollards, 2) 2D->3D estimation of the road work bollards, 3) generation of plausible lane boundaries and 4) update of HD-map based on the plausible lane boundary. A full description of the modules is given in D3.2.

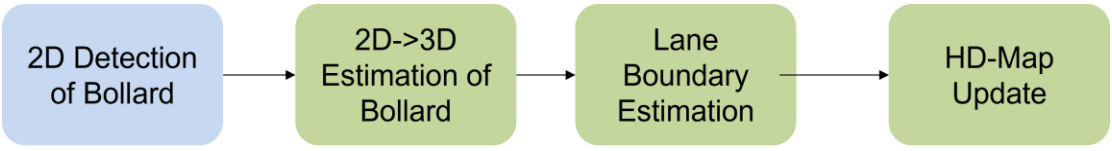


Figure 17 Workflow for detection of bollards to update HD-map.

The output produced by the perception platform is shown in Figure 18, where the road boundaries are updated according to the location of the bollards. In this example the

map frame (against the original map frame HD-map shown by the green lines). The red lines on the right panel are the left boundary of the drivable road, and the blue line is the right boundary of the drivable road. The light blue lines are the centerlines of the lane/s.

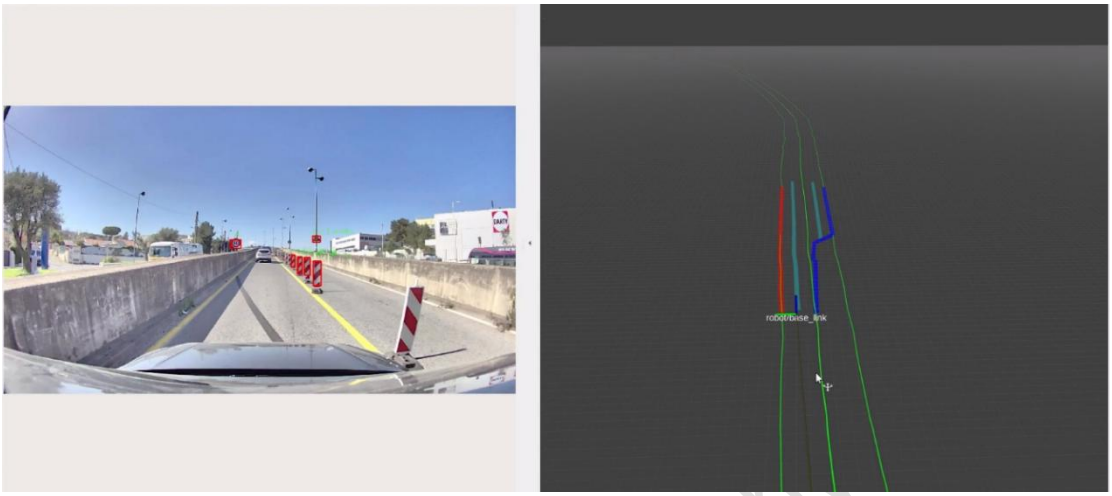


Figure 18 The environment and data gather (left). The lane boundary (red) based on HD Map(green) and Final system shows the updated HD-map

4.3 Dynamic Trajectory Generation with Adaptive Lane Boundaries

To enhance the motion planning framework, a dynamic trajectory generation method is required to effectively incorporate lane boundaries into the decision-making process. Optimization techniques based on quadratic errors are well-suited for this purpose, as they can accommodate dynamic constraints, allowing for real-time adjustments. In this context, we implemented a model-based predictive trajectory generator utilizing a kinematic vehicle model, which is a standard choice for developing Model Predictive Control (MPC) in low-speed scenarios. Notably, this method mirrors the approach studied in EXP2, reinforcing its applicability in various motion planning scenarios.

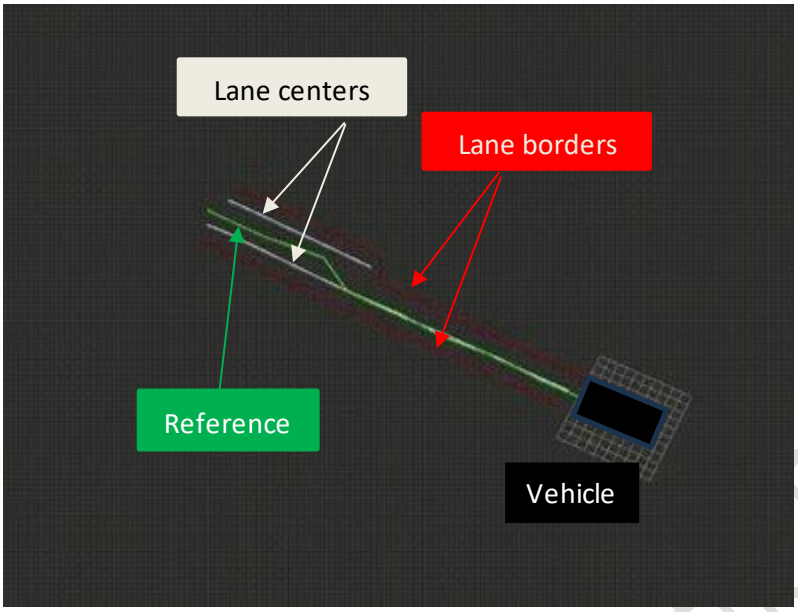


Figure 19 Trajectory generation using dynamic lanes from HD map

To integrate the dynamic lane boundaries, it is essential to discretize them in a manner consistent with the reference trajectory. This discretization ensures that the vehicle's motion planning can adapt to changes in the lane geometry, allowing for smoother navigation while adhering to safety regulations. By continuously updating the lane boundary data, the motion planner can dynamically adjust the vehicle's path in response to shifting lane conditions, optimizing both trajectory accuracy and vehicle stability. This approach not only enhances the vehicle's ability to navigate through varying environments but also ensures that it remains within the safe confines of the road, even in the presence of obstacles or changes in lane configuration.

5. EXP8 (TUD)

Experiment 8 (EXP8) focuses on the "Emergency evasion manoeuvre on slippery road under rain conditions". The objective is to avoid collisions (e.g., pedestrians or cyclists) in poor weather conditions on slippery roads.

5.1 SoTA

The state-of-the-art methods for motion planning were discussed in D4.1 “Initial version of motion planning and behavioural decision-making components”. Based on this analysis, the integration of motion planning, path tracking, and vehicle stability constraints has been selected using the Model Predictive Contouring Controller (MPCC). The MPCC employs an iterative approach to solving an optimal control problem, enabling the vehicle to maintain a low-risk, safe trajectory while successfully avoiding obstacles [10] [11].

Our recent research [10] showcases the application of the MPCC framework in high-speed collision avoidance scenarios, primarily when the vehicle is operating near its handling limits. This framework forms the basis of the proposed architecture [12] [1]. However, several enhancements are necessary to optimise the MPCC algorithm for emergency evasive manoeuvres on slippery roads, particularly under rainy conditions. These improvements are focused on three key areas: the vehicle prediction model, the tyre model, and the integration of uncertainties from the perception model into the optimal control problem formulation.

The vehicle prediction model has been refined to account for varying operational conditions encountered during evasive manoeuvres on slippery roads, such as the need for lower velocities due to a reduced road friction coefficient. The proposed tyre model is also more advanced, providing greater accuracy in representing combined slip conditions, which is critical for maintaining traction and stability in such scenarios. Furthermore, the obstacle representation within the MPCC has been enhanced by incorporating uncertainties derived from the perception module. This integration reduces the conservativeness typically associated with evaluating the distance between the vehicle and obstacles, leading to more precise and reliable collision avoidance strategies.

5.2 Architecture

The proposed MPCC for path tracking and motion replanning is integrated into Autoware as a replacement for the separated controllers for lateral and longitudinal dynamics, shown in Figure 20. The proposed controller tracks the trajectory of the motion planner when the vehicle is at a safe distance from the obstacles. As soon as the provided trajectory passes too close to the obstacle or it is unfeasible due to the

lower complexity of the prediction model, the proposed MPCC will perform a motion replanning to keep the vehicle at a safe distance from the obstacles.

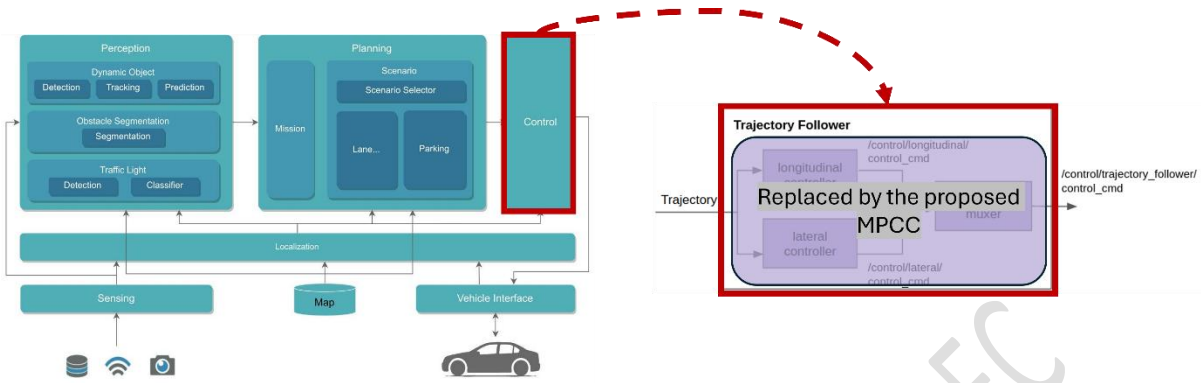


Figure 20 The architecture of the proposed MPCC in the Autoware ROS stack

5.3Algorithmic Approach

The proposed approach employs a nonlinear single-track vehicle model, which was selected in favour of the double-track vehicle model due to its balanced trade-off between accuracy and computational efficiency [13], [14], [15]. This model focuses solely on in-plane dynamics, omitting the effects of lateral weight transfer as well as roll and pitch dynamics to simplify the computational load. The vehicle's position is represented in a Cartesian reference frame through the state vector $x = [X, Y, \phi, v_x, v_y, r, \theta, \delta, F_x]$, which includes the longitudinal position (X), lateral position (Y), and the heading angle (ϕ) of the vehicle's center of gravity (CoG) relative to an inertial frame. The velocity components consist of the longitudinal and lateral velocities at the CoG, denoted by v_x and v_y, respectively, along with the yaw rate r. Additionally, an extra state variable θ is introduced to track the vehicle's travelled distance, which is utilized in the MPCC cost function to determine the vehicle's position relative to a reference path. The control inputs are the rate of the steering angle δ and the longitudinal force F_x. The corresponding state derivatives $\dot{x}$) are derived from the following equations, providing the necessary dynamics for the vehicle's motion.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\phi} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_x \cos\phi - v_y \sin\phi \\ v_x \sin\phi + v_y \cos\phi \\ r \\ \frac{1}{m}\left(F_x - F_{f,y} \sin\delta - F_{x,drag}\right) + v_y r \\ \frac{1}{m}\left(F_{r,y} + F_{f,y} \cos\delta\right) - v_x r \\ \frac{1}{I_{zz}}\left(F_{f,y} l_f \cos\delta F_{r,y} l_r\right) \\ \sqrt{v_x^2 + v_y^2} \end{bmatrix} \quad (5.1)$$

One issue with the dynamic bicycle model, $f_{dyn}(x,u)$, is that it becomes ill-defined at low velocities due to the dependence on slip angles. However, low velocities are relevant for an evasive manoeuvre on slippery roads, where the vehicle can significantly slow down due to the reduced friction coefficient. The most common solution is the implementation of kinematic models [16], which do not rely on slip angles. However, kinematic models are unsuitable for evasive manoeuvres on slippery roads because they do not account for the interaction between the tyres and the ground [17].

To achieve the optimal integration of both models within a single framework, we propose a vehicle model that combines dynamic and kinematic elements [17]. This is possible because the kinematic model is formulated to account for the same state variables of the dynamic model. Thus, $\dot{v}_x$ and $\dot{v}_y$ within the kinematic model are derived by differentiating the variables $v_{y,kin}$ and r_{kin}, which are described as follows:

$$v_{y,kin} = l_r \, r_{tar} \tag{5.2}$$

$$r_{kin} = \frac{\tan(\delta)\, v_x}{l_f + l_r}$$

where r_{tar} corresponds to the steady state yaw rate, which is computed as follows:

$$r_{tar} = \frac{\delta v_x}{l_f + l_r} \tag{5.3}$$

The new vehicle kinematic model, $f_{kyn}(x,u)$, is computed as follows:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\phi} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_x \cos\phi - v_y \sin\phi \\ v_x \sin\phi + v_y \cos\phi \\ r \\ \frac{F_x - F_{x,drag}}{m} \\ \left(\dot{\delta}\, v_x + \delta\, \dot{v}_x\right) \frac{l_r}{l_r + l_f} \\ \left(\dot{\delta}\, v_x + \delta\, \dot{v}_x\right) \frac{1}{l_r + l_f} \\ \sqrt{v_x^2 + v_y^2} \end{bmatrix} \tag{5.4}$$

The resulting model, represented as $\dot{x} = f(x,u)$, is formulated using the same state variables for both models, enabling their integration. The final vehicle model is then created by linearly combining these two models as follows:

$$\dot{x} = \lambda f_{dyn}(x,u) + (1-\lambda) f_{kin}(x,u) = f(x,u) \tag{5.5}$$

$$\lambda = min\left(max\left(\frac{v_x - v_{x,min}}{v_{x,max} - v_{x,min}}, 0\right), 1\right)$$

The models are combined only within the velocity range v_x in $[v_{x,min}, v_{x,max}]$. For velocities below $v_{x,min}$, we rely exclusively on the kinematic model, while for velocities above $v_{x,max}$, we use only the dynamic model [17]. This work assumes that $v_{x,min} = 4$ m/s and $v_{x,max} = 5$ m/s to avoid continuous switches between the kinematic and dynamic models.

The vehicle parameters are reported in Table 3 Vehicle parameters

.

Parameters	Symbol	Value
Vehicle mass, kg	m	1564
Vehicle inertia around the z-axis, kg m^2	I_{zz}	2473
Distance between the front axle to CoG, m	l_f	1.050
Distance between the rear axle to CoG, m	l_r	1.651
Air density, kg/m^3	ρ	1.204
Drag coefficient, [-]	C_{d1}	0.25
Rolling resistance, N	C_{d0}	45
Vehicle frontal area, m^2	A_f	2.4

Table 3 Vehicle parameters

Extended Fiala Tyre Model

The challenging weather conditions require a high level of accuracy for the prediction model. Thus, in the EVENTS project, the lateral tyre forces of the single-track vehicle model are described by an extended Fiala tyre model [11]. This model builds upon the classic Fiala tyre model, which is based on pure slip conditions, meaning that longitudinal and lateral slips are decoupled. Since the combined dynamics should be considered, the Fiala tyre model has been enhanced by incorporating modifications that account for variations in cornering stiffness due to changes in longitudinal and vertical forces [18]. Additionally, the saturation region of the model is adjusted to include a negative gradient, which prevents the prediction model from overestimating the maximum lateral force when the tyre operates at high lateral slip angles, such as during evasive manoeuvres in poor weather conditions. The extended Fiala tyre model is defined as follows:

$$F_y(\alpha, F_x, F_z) = \begin{cases} -C_{ym}(F_x, F_z)\tan\alpha + \frac{C_{ym}^2(F_x,F_z)\tan\alpha\tan|\alpha|}{3F_{y,max}} - \frac{C_{ym}^3(F_x,F_z)\tan^3\alpha}{27F_{y,max}^2} & if\ |\alpha| \le \alpha_{thr} \\ \frac{2C_{ym}(F_x,F_z)(\zeta-1)\tan\alpha}{3} - \frac{C_{ym}^2(F_x,F_z)(\zeta-1)\tan\alpha\tan|\alpha|}{9F_{y,max}} + & |\alpha| > \alpha_{thr} \\ -F_{y,max}\zeta\ sign(\alpha) & \end{cases} \quad (5.6)$$

The tire slip angle, denoted by α, represents the angle between the direction the tire is pointing and the actual direction of travel. The tire cornering stiffness, C_{ym}, is a function of both the vertical F_z and longitudinal F_x tire forces. The maximum lateral tire force is represented by $F_{y,max}$, while α_{thr} denotes the slip angle threshold at

which the tire's lateral force peaks. The parameter ζ, which ranges between 0 and 1, characterizes the gradient of the tire force in the saturation region.

When $|\alpha| \leq \alpha_{thr}$, and apart from the influence of F_x (which affects even at low slip angles), the extended Fiala model behaves akin to the classic Fiala tire model [19]. In contrast, when $|\alpha| > \alpha_{thr}$, the saturation region is modified to better represent the reduction in maximum lateral force at large slip angles. Despite these modifications, the proposed model retains the advantages of the classic Fiala tire model, ensuring continuity and differentiability at $|\alpha| = \alpha_{thr}$.

The presence of a non-zero gradient in the saturation region benefits numerical optimization algorithms that rely on gradient calculations, preventing derivative vanishing and improving the optimization of the road wheel angle when the tire operates in the saturation region [20]. Moreover, the proposed model exhibits a positive gradient when ζ is within the range [1,2] and a negative gradient when ζ is within [0,1].

The adaptation of the cornering stiffness C_y based on the vertical force F_z [18] can be expressed using the following equation:

$$C_y(F_z) = c_1 F_{z,0} \sin\left(2 \operatorname{atan}\left(\frac{F_z}{c_2 F_{z0}}\right)\right) \tag{5.7}$$

In this equation, c_1 and c_2 are optimised constants and F_{z0} represents a nominal vertical force. The cornering stiffness is further adjusted to account for its dependency on the longitudinal force F_x as follows:

$$C_{ym}(F_x, F_z) = \frac{1}{2}(\ \mu F_z - F_x\ ) + \left(1 - \left(\frac{|F_x|}{\mu F_z}\right)^{c_3}\right)^{-c_3}\left(C_y(F_z) - \frac{1}{2}\mu F_z\right) \tag{5.8}$$

where c_3 is an optimised parameter, and μ denotes the friction coefficient. The $C_{ym}(F_x, F_z)$ is used to compute the tyre slip threshold α_{thr} as follows:

$$\alpha_{thr} = \frac{3\ F_{y,max}}{C_{ym}(F_x, F_z)} \tag{5.9}$$

The maximum lateral tyre force $F_{y,max}$ is limited by the tyre friction circle, defined as follows:

$$F_{y,max} = \sqrt{(\mu\ F_z)^2 - F_X^2} \tag{5.10}$$

All the parameters utilized in the extended Fiala tyre model are detailed in Table 4 Tyre parameters

. Their values were determined through a nonlinear optimization process [21], which involved aligning the lateral tyre forces predicted by the proposed extended Fiala tyre model with those from a high-fidelity Delft-Tyre model 6.2.

Parameters	Symbol	Value
Lateral cornering stiffness effect, [-]	c_1	53.3
Lateral cornering stiffness peak effect, [-]	c_2	6.8
Long. and Lat. tyre force coupling effect, [-]	c_3	3.5
Nominal vertical tyre force (as in .tir property file), N	F_{z0}	4300
Tyre friction coefficient, [-]	μ	0.5

Table 4 Tyre parameters

Obstacle Representation

The objective of the proposed MPCC is to perform collision avoidance in poor weather conditions on slippery roads. The obstacles (e.g. pedestrian or cyclist) are represented as circle which radius R_{obs} is provided by the vehicle perception unit. Utilizing data from 3D object detection, the R_{obs} is enlarged by the evaluated uncertainties, which are assumed Gaussian. The obstacle of the radius considering the perception uncertainties is computed as follows:

$$R_{obs+unc} = R_{obs} + \sqrt{\chi_2^2(p)\lambda_{max}(\Sigma_{xy})} \tag{5.11}$$

Where $\chi_2^2(p)$ represents the value of the chi-squared distribution corresponding to a maximum violation probability of $p = 0.95$ [22]. Regarding $\lambda_{max}(\Sigma_{xy})$, it refers to the maximum eigenvalue of the variance matrix associated with obstacle localization. This variance can either be treated as a constant or dependent on the distance between the vehicle and the obstacle. For instance, the variance decreases as the vehicle approaches the obstacle. In the context of the EVENTS project, the term $R_{obs+unc}$ is integrated into the MPCC cost function to encourage the vehicle to maintain a safer distance from the obstacle.

The qualitative behaviour of the developed controller is illustrated in Figure 21 and Figure 22. In both figures, the behaviour of the proposed MPCC in a scenario with two obstacles in a low friction road is depicted. The four instances of the evasive manoeuvre are visible in chronological order.

5.4Outlook and Future Works

The proposed controller can successfully work at high and low velocities in slippery conditions based on qualitative results. A detailed quantitative validation is planned, including a comparison with the default Autoware planner , which will be reported in D6.2 “Technical evaluation results”.

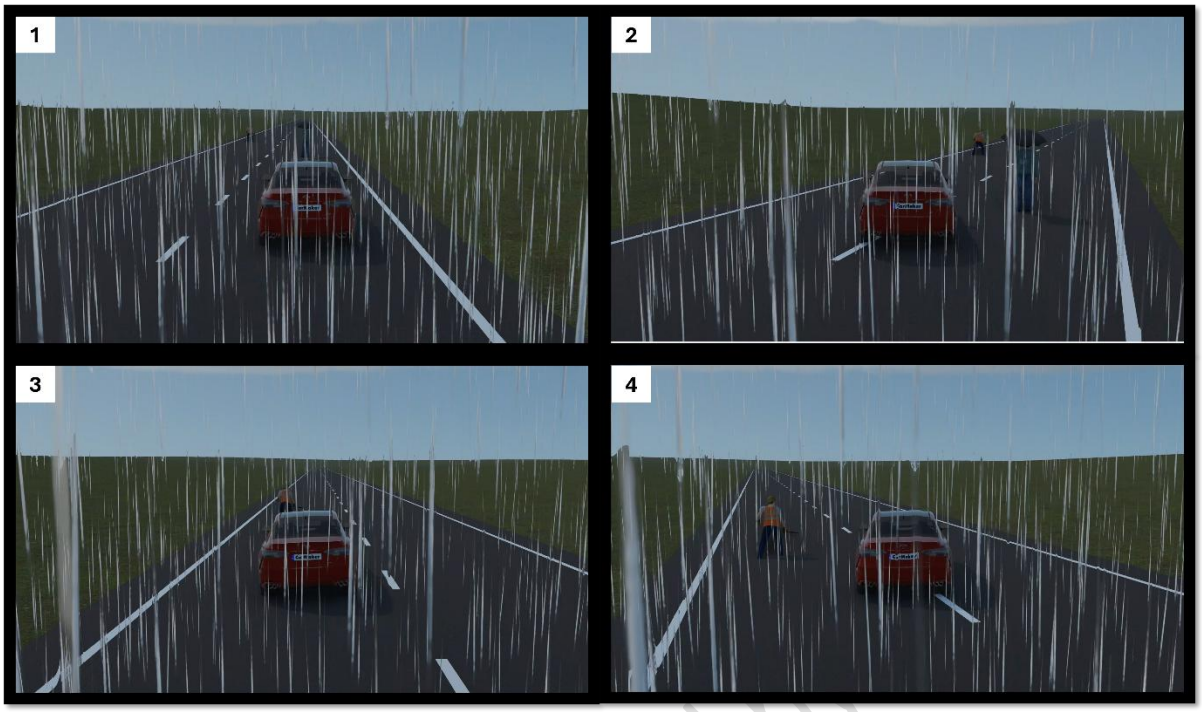


Figure 21 Behaviour of the proposed MPCC in evasive manoeuvre

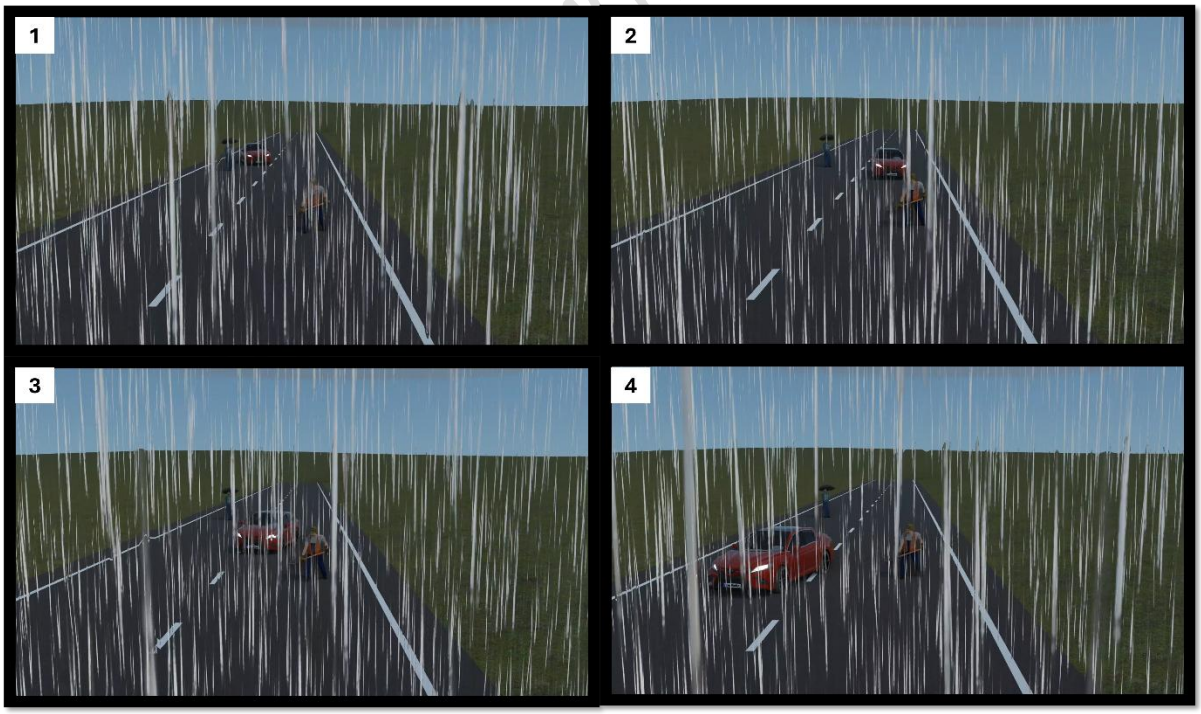


Figure 22 Behaviour of the proposed MPCC in poor friction condition

6. Conclusions

The aim of this document is to illustrate the final developments in Task 4.1 (T4.1), with a focus on motion planning, which will be implemented and tested across selected experiments in the project. Results and quantitative analyses are planned in future deliverables, specifically the ones in WP6 (particularly T6.2 dealing with the technical evaluation of the EVENTS systems in different scenarios and T6.3, aimed at performing the full quantitative outcomes and at validating the module's performance). In this context, D4.2 considers only some experiments (EXPs), because not everyone develops both motion and behavioural planning (depending on the goal of each EXP). Therefore, motion planning will be developed and tested for EXP1, EXP2, EXP4, and EXP8.

Each experiment includes a dedicated description of how the module is adapted to meet its requirements. The algorithms and methodologies used for motion planning have been specifically tailored and optimized for each experiment, detailing its development and integration into various experimental setups. It is worth noting here that these EXPs are concurrently integrated within both the simulation environment and the real-world platform, forming the core of ongoing work in WP5.

Considering the different descriptions of the motion planning algorithms and methods – as explained in Sections 2-5 – there are some common aspects that we want to highlight here. First, high computational time is a main concern in EVENTS applications because of the need to run these algorithms in real-time and online in specific demonstrators (independently of the fact they are prototype vehicles or driving simulators). Based on that, it was necessary to make some decisions in terms of the choice of algorithms, to ensure the trade-off between accuracy and computational efficiency. Secondly, the MPC approach has been considered in almost all the experiments, including its possible variances to solve specific problems (i.e., see EXP1 and EXP8). The rationale is that MPC is a well-known control method, with the possibility to use it to generate trajectories that consider the dynamics of the vehicle alongside other constraints. Finally, the methodologies described in each EXP have shown the importance of having a combined approach between data collected in simulation and in a real-world context. This is essential to have a strong design-development-testing loop of the different motion planning algorithms.

To sum up, deliverable D4.2 represents a comprehensive overview of WP4/T4.1 progress, pointing out the last next steps that will be finalized in WP5 (for the integration) and in WP6 (for the evaluation of the full E2E chain).

References

[1] *EVENTS Deliverable D2.2: Full Stack Architecture & Interfaces,* 2023.

[2] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman and D. Rus, "Safe Nonlinear Trajectory Generation for Parallel Autonomy With a Dynamic Vehicle Model," *IEEE Trans. Intelligent Transportation Systems,* vol. 19, no. 9, p. 2994–3008, 2018.

[3] L. Lyons and L. Ferranti, "Curvature-Aware Model Predictive Contouring Control," *2023 IEEE International Conference on Robotics and Automation (ICRA),* pp. 3204-3210, 2023.

[4] H. Zhu y J. Alonso-Mora, «Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments,» *IEEE Robot. Autom. Lett.,* vol. 4, nº 2, p. 776–783, 2019.

[5] A. Domahidi y J. Jerez, «FORCE-Pro Embotech,» 2014. [En línea]. Available: https://www.embotech.com/softwareproducts/prodriver/overview/.

[6] L. Zheng, P. Zeng, W. Yang, L. Yinong y Z. Zhan, «Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance,» *IET Intelligent Transport Systems,* vol. 14, nº 13, pp. 1882-1891, 2020.

[7] I. Bae, J. H. Kim y J. Moon, «Lane change maneuver based on Bezier curve providing comfort experience for autonomous vehicle users,» *2019 IEEE intelligent transportation systems conference (ITSC),* 2019.

[8] A. Artuñedo, J. Godoy y J. Villagra, «A primitive comparison for traffic-free path planning,» *IEEE Access,* 2018.

[9] Y. D. García Ramírez y D. Aguilar Cárdenas, «Passengers' comfort in horizontal curves on mountain roads: A field study using lateral accelerations,» *Revista Facultad de Ingeniería Universidad de Antioquia,* 2021.

[10] A. Bertipaglia, M. Alirezaei, R. Happee y B. Shyrokau, «Model predictive contouring control for vehicle obstacle avoidance at the limit of handling,» de *International Symposium on Dynamics of Vehicles on Roads and Tracks*, Ottawa, 2023.

[11] A. Bertipaglia, D. Tavernini, U. Montanaro, M. Alirezaei, R. Happee, A. Sorniotti y B. Shyrokau, «Model Predictive Contouring Control for Vehicle Obstacle Avoidance at the Limit of Handling Using Torque Vectoring,» de *International Conference on Advanced Intelligent Mechatronics*, Boston, 2024.

[12] *EVENTS Deliverable D2.1: User and system requirements for selected use cases,* 2023.

[13] M. Brown y C. Gerdes, «Coordinating tire forces to avoid obstacles using nonlinear model predictive control,» *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES,* vol. 5, nº 1, pp. 22-31, 2020.

[14] R. Hajiloo, M. Abroshan, A. Khajepour, A. Kasaiezadeh y S.-. K. Chen, «Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles,» *IEEE Transactions on Intelligent Transportation Systems,* vol. 22, nº 5, pp. 3167-3178, 2021.

[15] A. K. D. Kasinathan, A. Wong, A. Khajepour, S.-K. Chen y B. Litkouhi, «An optimal torque vectoring control for vehicle applications via real-time constraints,» *IEEE Transactions on Vehicular Technology,* vol. 65, nº 6, pp. 4368-4378, 2015.

[16] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu y L. Chen, «Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives,» *IEEE Transactions on Intelligent Vehicles,* vol. 8, nº 6, pp. 3692-3711, 2023.

[17] J. Kabzan, M. d. l. I. Valls, V. Reijgwart, H. F. C. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan, A. Dhall, E. Chisari y K. Napat, «AMZ Driverless: The full autonomous racing system,» *Journal of Fields Robotics,* vol. 37, pp. 1267-1294, 2020.

[18] J. K. Subosits y J. C. Gerdes, «Impacts of model fidelity on trajectory optimization for autonomous vehicles in extreme maneuvers,» *IEEE Transactions on Intelligent Vehicles,* vol. 6, nº 3, pp. 546-558, 2021.

[19] T. P. Weber y J. C. Gerdes, «Modeling and control for dynamic drifting trajectories,» vol. 9, nº 2, pp. 3731-3741, 2024.

[20] V. A. Laurense y J. C. Gerdes, «Long-horizon vehicle motion planning and control through serially cascaded model complexity,» *IEEE Transactions on Control Systems Technology,* vol. 30, nº 1, pp. 166-179, 2022.

[21] A. Bertipaglia, B. Shyrokau, M. Alirezaei y R. Happee, «A two-stage bayesian optimisation for automatic tuning of an unscented kalman filter for vehicle sideslip angle estimation,» de *IEEE Intelligent Vehicles Symposium*, Aachen, 2022.

[22] A. Bertipaglia, M. Alirezaei, R. Happee y B. Shyrokau, «A Learning-Based Model Predictive Contouring Control for Vehicle Evasive Manoeuvres,» de *15th International Symposium on Advanced Vehicle Control*, Kanagawa, 2022.

[23] HERE, "Maps for ADAS and HAD," HERE, [Online]. Available: https://www.here.com/platform/adas-had. [Accessed 4 December 2023].

[24] tomtom, «How we make our HD maps,» [En línea]. Available: https://www.tomtom.com/newsroom/behind-the-map/how-we-make-our-hd-maps/. [Último acceso: 01 09 2024].

[25] Q. Li, Y. Wang, Y. Wang y H. Zhao, «HDMapNet: An Online HD Map Construction and Evaluation Framework,» de *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 4628-4634.

[26] Y. Liu, T. Yuan, Y. Wang, Y. Wang y H. Zhao, «Vectormapnet: End-to-end vectorized hd map learning,» de *International Conference on Machine Learning*, PMLR, 2023, pp. 22352-22369.

[27] Y. Guo, J. Zhou, X. Li, Y. Tang y Z. Lv, «A Review of Crowdsourcing Update Methods for High-Definition Maps,» *ISPRS International Journal of Geo-Information,* vol. 13, nº 3, p. 104, 2024.