Squadra Corse Driverless

Autonomous System

Project Configuration

Politecnico di Torino

October 21, 2024

Contents

1 Introduction **3**
1.1 Autonomous system 4

2 ROS2 pipeline **5**

3 Packages and nodes **6**

4 Interfaces **7**
4.1 Notes 8
4.1.1 sensor_msgs/PointCloud2 8
4.1.2 sensor_msgs/PointCloud2 9

5 Topics **10**

6 Parameters tables **11**
6.1 Launch Parameters (AS only) 11
6.2 Simulation parameters (SIM only) 12
6.3 General parameters 12
6.4 Perception simulation parameters (SIM only) 12
6.5 Camera driver parameters (AS only) 13
6.6 Ground removal parameters 14
6.7 Clustering Parameters 15
6.8 YOLO Parameters 15
6.9 Sensor Fusion Parameters 16
6.10 SLAM Parameters 17
6.11 Local path planning parameters 18

6.12 Path planning parameters 19
6.13 Path tracking parameters 20
6.14 Vehicle model parameters 22
6.15 Visualization Parameters 23

1 Introduction

The purpose of this document is to provide a detailed description of the autonomous system and simulator ROS2 projects describing:

- **Autonomous system**: overview of the autonomous system logical block diagram.
- **Pipeline**: description of nodes implementations compared to logical blocks.
- **Packages and nodes**: how implemented algorithms are managed inside the ROS2 project.
- **Interfaces**: definitions of custom message interfaces.
- **Topics**: topics used for inter-process communication.
- **Parameters tables**: description of all launch parameters.

1.1 Autonomous system

In Figure 1, the logical block diagram of the autonomous system is presented. LiDAR and camera data are processed to gather informations about the environment, which are then merged using a sensor fusion process. A SLAM (Simultaneous Localization And Mapping) algorithm is responsible for mapping the track in real time and localizing the vehicle, optimizing both the positions of the cones and the vehicle state.

Each time the map is updated, a path to follow is generated by the path planning process, which is subsequently refined by a trajectory optimizer. A velocity profiler determines the desired velocity to track at each point along the path. Finally, the path tracker computes the commands the vehicle should follow, including steering angle, and velocity reference or throttle input.

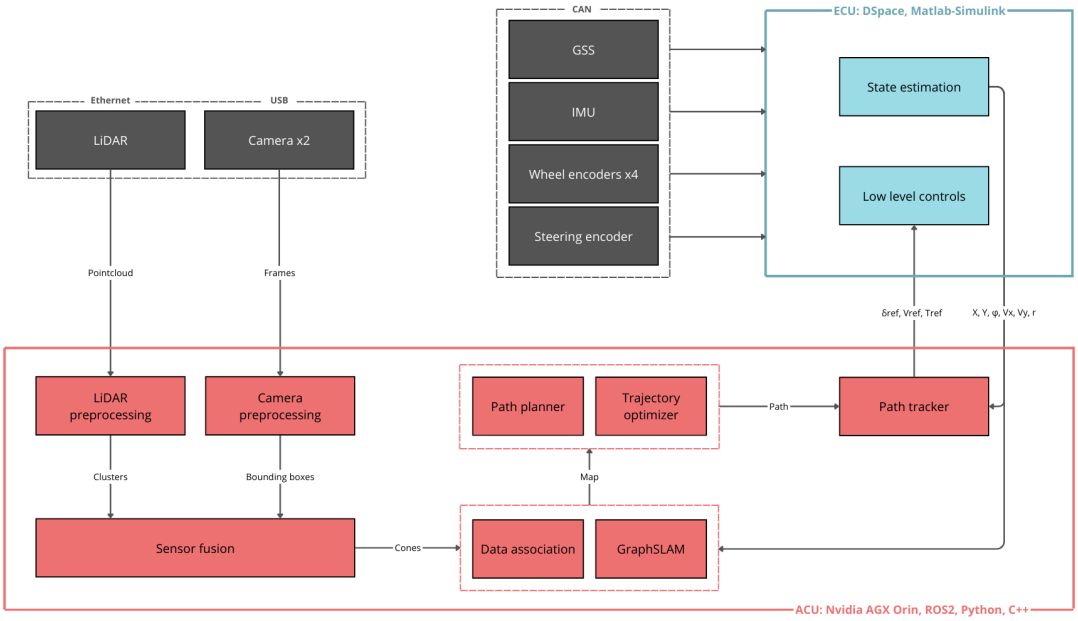

Figure 1: Autonomous system overview

2 ROS2 pipeline

Figure 2 illustrates the ROS2 pipeline, highlighting the communication topics between various package nodes. Below is a legend explaining the classification of the packages:

- **Green**: Perception packages, responsible for processing sensor data and understanding the environment.
- **Red**: Packages for SLAM, path planning, and path tracking, which handle mapping, trajectory generation, and control.
- **Orange**: Finite state machine, logger, TCP inter-process communication, and visualization packages, which are accessible by all nodes.
- **Yellow**: Perception simulation and vehicle model packages, primarily used for simulation purposes.
- **Light blue**: ECU tasks, managing low-level vehicle control and state estimation.

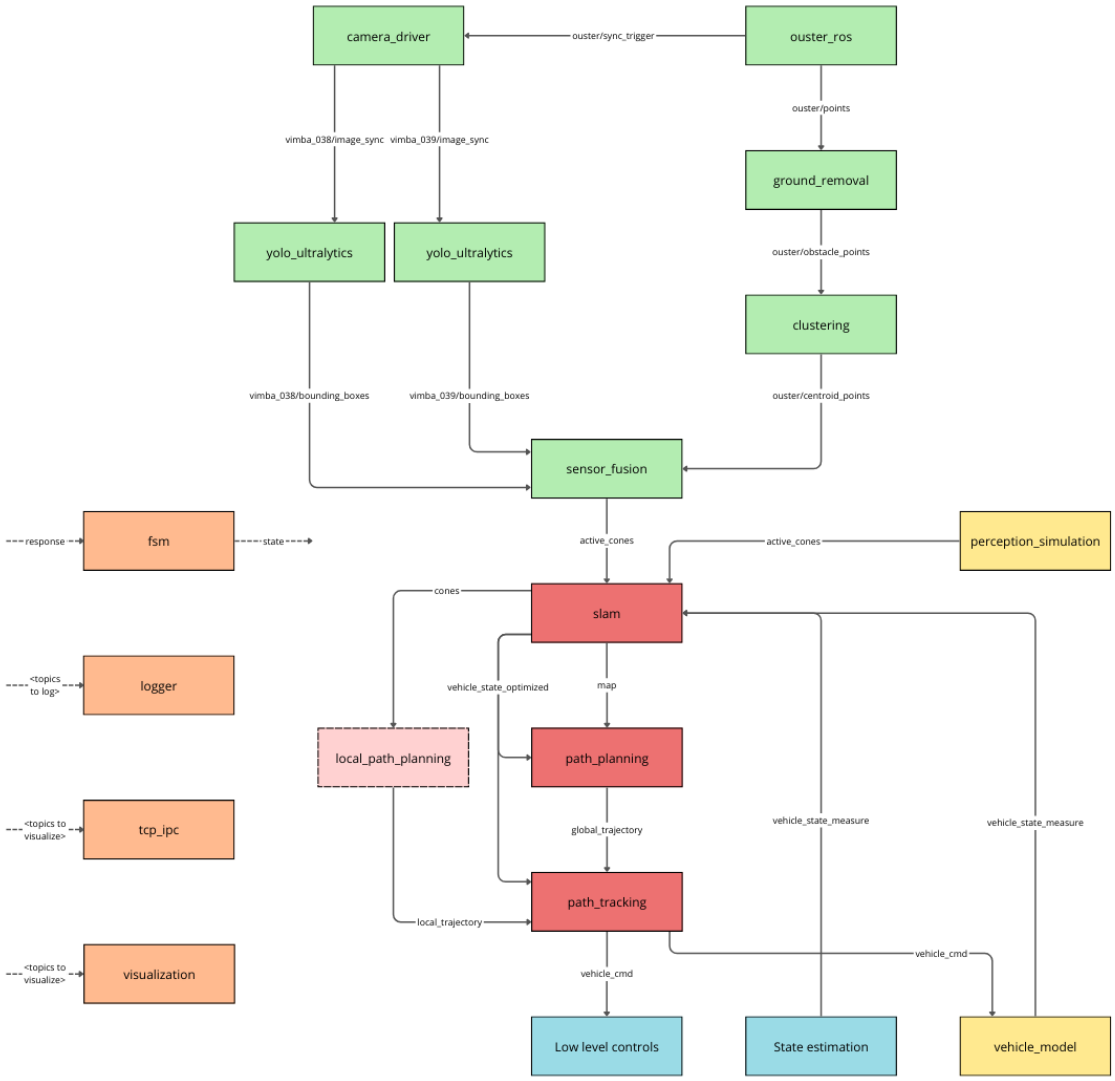

Figure 2: ROS2 packages pipeline

3 Packages and nodes

Package	Node	Description
camera_driver	vimba_driver	Manages camera connection and data streaming.
yolo_ultralytics	yolo	YOLO object detection, publishes cone bounding boxes.
ouster_ros	os_cloud	Publishes LiDAR point cloud.
ouster_ros	os_image	Publishes LiDAR-based reconstructed image.
ouster_ros	os_sensor	Publishes sensor data from LiDAR.
ground_removal	ground_removal	Executes ground plane removal from point cloud.
clustering	clustering	Performs DBSCAN clustering.
sensor_fusion	sensor_fusion	Synchronizes and merges LiDAR and cameras data.
slam	graph_slam	Handles data association and SLAM.
path_planning	global_planner	Wrap global path planner, trajectory optimizer and velocity profiler for autocross and trackdrive events.
global_path_planning	ransac_planner	Path planner for the acceleration event.
local_path_planning	rrt_planner	Previous local path planner (deprecated).
local_path_planning	discrete_tree_planner	Previous local path planner (deprecated).
path_tracking	pure_pursuit	Manages lateral dynamics in the first lap.
path_tracking	mpcc	Controls both longitudinal and lateral dynamics in later laps.
fsm	fsm	Manages state transitions for synchronized algorithms.
logger	bag_logger	Logs data into ROS2 bags.
tcp_ipc	tcp_server	Launches pipeline via TCP from simulator/external sources.
tcp_ipc	tcp_client	Sends visualization data to the simulator/external sources.
visualization	rviz_plotter	Node for visualization using RViz2.

Table 1: Packages and nodes description

4 Interfaces

Interface	Fields	Description
sensor_msgs/PointCloud2	height: uint32, width: uint32, fields PointField[ ], is_bigendian: bool, point_step: uint32, row_step: uint32, data: uint8[ ], is_dense: bool	Describe the point cloud. Height and width are respectively the number of channels and the number of acquisitions per channel. The field fields describe the point structure with metadata about point data to be extracted correctly from data field. Fields point_step and row_step are respectively the offsets in the field data to reach the next point and the next row of the point cloud. If there are not invalid points is_dense is set to true.
sensor_msgs/PointField	name: string, offset: uint32, datatype: uint8, count: uint32	Describe a field of a point by name, offset from start of point structure, datatype (INT8, UINT8, INT16, UINT16, INT32, UINT32, FLOAT32, FLOAT64), and count of elements in the field.
sensor_msgs/Image	height: uint32, width: uint32, encoding: string, is_bigendian: uint8, step: uint32, data: uint8[ ]	Image data message format.
interfaces/BoundingBoxArray	data: BoundingBox[ ]	Array of bounding boxes.
interfaces/BoundingBox	class_id: int16, class_name: string, probability: float64, xmin: int64, ymin: float64, xmax: float64, ymax: float64	Bounding box describing a class referenced with class_id and class_name. The confidence is described as probability and the bounding box is described by pixel dimensions (xmin, xmax) and (ymin, ymax) into the image.

Table 2: Interfaces description

interfaces/ConeArray	data: Cone[ ]	Array of cones.
interfaces/Cone	x: float64, y: float64, color: int16, id: int16	Cone described by its position (x, y) in the plane, color (0: unknown, 1: yellow, 2: blue, 3: orange, 4: big orange) and id.
interfaces/WaypointArray	data: Waypoint[ ]	List of trajectory waypoints.
interfaces/Waypoint	x: float64, y: float64, vel_ref: float64	Waypoint described by its position (x, y) in the plane and velocity reference for tracking.
interfaces/VehicleStateArray	data: VehicleState[ ]	List of vehicle states.
interfaces/VehicleState	x: float64, y: float64, yaw: float64, v_x: float64, v_y: float64, yaw_r: float64, s: float64, v_s: float64, delta: float64, d: float64	State described by the values (x, y, yaw) and derivatives (v_x, v_y, yaw_r), along with cumulative displacement s and derivative v_s, current steering delta and current throttle d.
interfaces/VehicleCmd	type: uint8, delta: float64, d: float64, vs: float64	Vehicle control command expressed by type (VELOCITY, THROTTLE), steering delta, velocity reference vs and throttle reference d.
interfaces/State	data: uint8	FSM state (INITIALIZATION, EXPLORATION, RACING, STOP, ERROR)
interface/Response	data: uint8	FSM response (SUCCESS, ERROR)

Table 3: Interfaces description

4.1 Notes

4.1.1 sensor_msgs/PointCloud2

LiDAR points are described by sensor_msgs/PointCloud2 messages, in particular ouster_ros package deliver point clouds with the following characteristics:

- **height**: 64 channels
- **width**: 1024 points per vertical channel
- **point_step**: 48 bytes
- **row_step**: 49152 bytes

- **is_bigendian**: false

4.1.2 sensor_msgs/PointCloud2

Here are described the fields describing a point:

Field	**Offset**	**Type**	**Size**
x	0 bytes	float32	4 bytes
y	4 bytes	float32	4 bytes
z	8 bytes	float32	4 bytes
intensity	16 bytes	float32	4 bytes
t	20 bytes	uint32	4 bytes
reflectivity	24 bytes	uint16	2 bytes
ring	26 bytes	uint16	2 bytes
ambient	28 bytes	uint16	2 bytes
range	32 bytes	uint32	4 bytes

Table 4: Point fields in ouster point cloud

5 Topics

Topic	Interface	Description
ouster/sync_trigger	std_msgs/bool	LiDAR to cameras trigger signal.
ouster/points	sensor_msgs/PointCloud2	LiDAR point cloud.
ouster/obstacle_points	sensor_msgs/PointCloud2	Ground filtered point cloud.
ouster/centroid_points	sensor_msgs/PointCloud2	Point cloud of cluster centroids.
vimba_038/image_sync	sensor_msgs/Image	Image from vimba_038 camera.
vimba_039/image_sync	sensor_msgs/Image	Image from vimba_039 camera.
vimba_038/bounding_boxes	interfaces/BoundingBoxArray	Bounding boxes from vimba_038 camera.
vimba_039/bounding_boxes	interfaces/BoundingBoxArray	Bounding boxes from vimba_039 camera.
active_cones	interfaces/ConeArray	Cones perceived in local frame.
cones	interfaces/ConeArray	Mapped cones in local frame.
map	interfaces/ConeArray	All mapped cones in global frame.
local_trajectory	interfaces/WaypointArray	Trajectory in local frame.
global_trajectory	interfaces/WaypointArray	Trajectory in global frame.
vehicle_state	interfaces/VehicleState	Real vehicle state (simulation).
vehicle_state_measure	interfaces/VehicleState	Measured vehicle state.
vehicle_state_optimized	interfaces/VehicleState	SLAM optimized vehicle state.
vehicle_cmd	interfaces/VehicleCmd	Vehicle controls command.
state	interfaces/State	Finite state machine state.
response	interfaces/Response	Response to finite state machine.

Table 5: Topics description

6 Parameters tables

6.1 Launch Parameters (AS only)

Parameter	Type	Description	Default
launch_fsm	bool	Launch FSM nodes.	True
launch_sbg_imu	bool	Launch SBG IMU nodes.	True
launch_gss	bool	Launch GSS nodes.	True
launch_lidar	bool	Launch LiDAR nodes.	True
launch_camera	bool	Launch camera nodes.	True
launch_ground_removal	bool	Launch ground removal nodes.	True
launch_clustering	bool	Launch clustering nodes.	True
launch_yolo	bool	Launch YOLO nodes.	True
launch_sensor_fusion	bool	Launch sensor fusion nodes.	True
launch_slam	bool	Launch SLAM nodes.	True
path_planning	bool	Launch local path planning nodes.	True
path_tracking	bool	Launch path tracking nodes.	True
launch_kvaser_interface	bool	Launch Kvaser interface nodes.	True
launch_logger	bool	Launch logger nodes.	True
launch_visualization	bool	Launch visualization nodes.	True

Table 6: Launch parameters description

6.2 Simulation parameters (SIM only)

Parameter	Type	Description	Default
interface	string	Simulation interface (external, rviz).	external

Table 7: Simulation parameters description

6.3 General parameters

Parameter	Type	Description	Default
task	string	Specifies the task to complete (skidpad, acceleration, autocross, trackdrive)	-
x_init	double	Initial x coordinate of the vehicle (m).	0.0
y_init	double	Initial y coordinate of the vehicle (m).	0.0
yaw_init	double	Initial yaw of the vehicle (rad).	0.0
v_x_init	double	Initial x coordinate of the vehicle velocity (m/s).	0.0
v_y_init	double	Initial y coordinate of the vehicle velocity (m/s).	0.0
yaw_r_init	double	Initial yaw rate of the vehicle (rad/s).	0.0

Table 8: General parameters description

6.4 Perception simulation parameters (SIM only)

Parameter	Type	Description	Default
track	string	Simulated track (skidpad, acceleration, cerrina, fsg19, fsonline20).	-
perception_rate	double	Perception rate (Hz).	10.0
std_dev	double	Standard deviation of the Gaussian noise applied to cone position.	0.05
fov	double	Distance of the field of view (m).	12.0
fov_clear	double	Distance of the field of view with clear color detection (m).	12.0

Table 9: Perception simulation parameters description

6.5 Camera driver parameters (AS only)

Parameter	Type	Description	Default
source_width	integer	Width of the source frame in pixels.	2464
source_height	integer	Height of the source frame in pixels.	1000
source_offset_x	integer	Horizontal offset of the source frame from the origin in pixels.	0
source_offset_y	integer	Vertical offset of the source frame from the origin in pixels.	1056
destination_scale	double	Scaling factor applied to the destination frame, affecting its resolution.	3

Table 10: Camera parameters description

6.6 Ground removal parameters

Parameter	Type	Description	Default
vehicle_hiding_len	double	Length threshold for filtering out vehicle points (m).	1.0
vehicle_hiding_width	double	Width threshold for filtering out vehicle points (m).	1.0
cone_height	double	Maximum allowable height for cone points (m).	0.4
sensor_height	double	Height of the sensor above the ground (m).	1.0
x_filter	double	Longitudinal distance filter for point consideration (m).	1.0
r_min	double	Minimum radius for radial filtering (m).	0.5
r_max	double	Maximum radius for radial filtering (m).	20.5
n_threads	integer	Number of threads to use for parallel processing.	4
n_bins	integer	Number of radial bins for filtering data.	80
n_segments	integer	Number of segments to divide each radial bin.	120
max_dist_to_line	double	Maximum distance a point can be from the fitted line to be considered for filtering (m).	0.06
max_slope	double	Maximum allowable slope for the fitting line.	0.3
min_slope	double	Minimum allowable slope for the fitting line.	-0.3
long_threshold	double	Threshold for determining long objects (m).	1.0
max_long_height	double	Maximum height deviation of points from the fitting line (m).	0.1
max_start_height	double	Maximum starting height of the fitting line (m).	0.1
line_search_angle	double	Angle used in line search algorithm (radians).	0.1
max_fit_error	double	Maximum permissible error in the fitting process (m).	0.05

Table 11: Ground removal parameters description

6.7 Clustering Parameters

Parameter	Type	Description	Default
eps	double	Minimum distance required to consider two points as part of the same cluster (m).	0.2
min_points	integer	Minimum number of points required for a group of points to be classified as a cluster.	2
centroid_max_distance	double	Maximum allowable distance of points from their cluster centroid (m).	0.3
centering_diameter_ratio	double	Ratio of cone diameter that determines how much the cluster centroid is displaced towards the acquisition direction to fill the gap between the centroid and cone center.	0.18
cone_filter_score_threshold	double	Minimum score threshold for filtering cone shapes during the clustering process.	0.35

Table 12: Clustering parameters description

6.8 YOLO Parameters

Parameter	Type	Description	Default
yolo_model	string	Name of the YOLO model to be used for detection.	yolov8n.pt
device	string	Computing device to be utilized for running the model (e.g., 'cuda:0' for GPU or 'cpu').	cuda:0
probability_threshold	double	Minimum probability score for a detection to be considered valid and published.	0.6

Table 13: YOLO parameters description

6.9 Sensor Fusion Parameters

Parameter	Type	Description	Default
extrinsic_matrix_vimba_038	array	Extrinsic transformation matrix for the vimba_038 camera, defining its position and orientation relative to the world frame.	[0.4226, -0.9063, 0.0, 0.0453, 0.0, 0.0, -1.0, 0.1, 0.9063, 0.4226, 0.0, -0.0211, 0.0, 0.0, 0.0, 1.0]
extrinsic_matrix_vimba_039	array	Extrinsic transformation matrix for the vimba_039 camera, defining its position and orientation relative to the world frame.	[-0.4226, -0.9063, 0.0, -0.0453, 0.0, 0.0, -1.0, 0.1, 0.9063, -0.4226, 0.0, -0.0211, 0.0, 0.0, 0.0, 1.0]
intrinsic_matrix_vimba_038	array	Intrinsic camera matrix for the vimba_038 camera, describing its internal parameters such as focal length and optical center.	[820.0, 0.0, 430.0, 0.0, 700.0, -5.0, 0.0, 0.0, 1.0]
intrinsic_matrix_vimba_039	array	Intrinsic camera matrix for the vimba_039 camera, describing its internal parameters such as focal length and optical center.	[820.0, 0.0, 420.0, 0.0, 700.0, -5.0, 0.0, 0.0, 1.0]
max_association_pixel_distance	integer	Maximum pixel distance allowed between the mapped cluster centroid and the bounding box center for them to be considered associated.	80

Table 14: Sensor fusion parameters description

6.10 SLAM Parameters

Parameter	Type	Description	Default
x_sensor_offset	double	Sensor x offset in the vehicle frame (m).	0.0
y_sensor_offset	double	Sensor y offset in the vehicle frame (m).	0.0
yaw_sensor_offset	double	Sensor yaw offset in the vehicle frame (rad).	0.0
inner_origin_dist	double	Inner origin distance for stopping condition (m).	3.0
outer_origin_dist	double	Outer origin distance for stopping condition (m).	4.0
origin_cone_ids_size	int	Dimension of cone ids buffer to match for stopping condition.	16
filter_dist	double	Maximum distance of published cones (m).	10.0
min_opt_dist	double	Minimum vehicle displacement between optimization processes (m).	0.2
loc_only_acquisition_size	int	Dimension of optimization buffer in localization only mode.	50
end_lap	int	Override the number of laps until the end of the trackdrive mission.	-1
cov_obs	list	Observation measure covariance matrix.	[0.05, 0.0, 0.0, 0.05]
cov_odo	list	Odometry measure covariance matrix.	[0.05, 0.0, 0.0, 0.0, 0.05, 0.0, 0.0, 0.0, 0.0000005]

Table 15: SLAM parameters description

6.11 Local path planning parameters

Parameter	Type	Description	Default
length	int	Number of path steps with low discretization.	7
fine_length	int	Number of path steps with high discretization.	0
step	double	Step length (m).	1.0
max_midpoint_dist	double	Maximum middle point distance from generating cones (m).	3.0
known_weight	double	Known color cones weight factor.	1.0
unknown_weight	double	Unknown color cones weight factor.	0.3
max_angle	double	Maximum angle between subsequent steps (degrees).	20.0
angle_step	double	Angle discretization step for low discretization (degrees).	10.0
fine_angle_step	double	Angle discretization step for high discretization (degrees).	5.0
min_leaf_target_dist	double	Distance from middle points in which target tree leaf nodes are selected to be evaluated (m).	0.8
cone_dist_limit	double	Cone distance from which the score of the path is reduced (m).	1.0
close_to_cone_factor	double	Cone distance path score reducing factor.	30.0
midpoint_dist_limit	double	Middle point distance from which the score of the path is increased (m).	1.5
close_to_midpoint_factor	double	Middle point distance path score increasing factor.	120.0
theta_factor	double	Smoothness factor.	15.0
iteration_num	int	Number of randomly generated steps.	1500
min_overload_dist	double	Minimum distance between generated tree nodes (m).	0.3
std_deviation	double	Standard deviation of the Gaussian distribution from which points are generated around target points (middle points or target cones).	1.5
min_cone_target_dist	double	Minimum cone distance from vehicle to be a target point (m).	6.0
max_cone_target_dist	double	Maximum cone distance from vehicle to be a target point (m).	10.0

Table 16: Local path planning parameters description

6.12 Path planning parameters

Parameter	Type	Description	Default
use_unknown_cones	bool	Use unknown cones information to compute the trajectory.	False
path_optimization	bool	Compute optimized path.	False

Table 17: Path planning parameters description

6.13 Path tracking parameters

Parameter	Type	Description	Default
fitting_step	double	Fitting step for spline path interpolation.	0.1
look_ahead	double	Look ahead to determine the waypoint to follow (m).	3.0
vel_ref	double	Maximum velocity reference (m/s).	7.0
lateral_acceleration_max	double	Maximum lateral acceleration (G).	0.3
ts_mpc	double	Discretization time for MPC (s).	0.05
n_sqp	int	Number of iterations of SQP algorithm for each control step.	3
n_reset	int	Number of SQP failures to get the last solution as current for each control step.	1000000000
sqp_mixing	double	Mixing parameter between last and current solution.	0.9
fitting_step_mpc	double	Fitting step for spline path interpolation.	0.05
r_in	double	-	0.4
r_out	double	-	0.4
max_dist_proj	double	-	3.0
e_long	double	-	1.25
e_eps	double	-	0.95
max_alpha	double	-	0.15
initial_velocity	double	-	0.0
s_trust_region	double	-	30.0
vx_zero	double	-	0.3
cost_q_c	double	-	0.8
cost_q_l	double	-	1000.0
cost_q_vs	double	-	30.0
cost_q_mu	double	-	0.1
cost_q_r	double	-	1e-4
cost_q_beta	double	-	0.0
cost_r_D	double	-	1e-1
cost_r_delta	double	-	1e-3
cost_r_vs	double	-	1e-1

Table 18: Path tracking parameters description

Parameter	Type	Description	Default
cost_r_dD	double	-	1e-1
cost_r_dDelta	double	-	5e2
cost_r_dVs	double	-	1e-4
cost_q_c_N_mult	double	-	1000.0
cost_q_r_N_mult	double	-	10.0
cost_sc_quad_track	double	-	10000.0
cost_sc_quad_tire	double	-	6400.0
cost_sc_quad_alpha	double	-	6400.0
cost_sc_lin_track	double	-	100.0
cost_sc_lin_tire	double	-	80.0
cost_sc_lin_alpha	double	-	80.0
state_bound_X_l	double	-	-1000000000.0
state_bound_Y_l	double	-	-1000000000.0
state_bound_phi_l	double	-	-1000000000.0
state_bound_vx_l	double	-	0.0
state_bound_vy_l	double	-	-5.0
state_bound_r_l	double	-	-2.0
state_bound_s_l	double	-	-1.0
state_bound_D_l	double	-	-0.0
state_bound_delta_l	double	-	-0.35
state_bound_vs_l	double	-	0.0
state_bound_X_u	double	-	1000000000.0
state_bound_Y_u	double	-	1000000000.0
state_bound_phi_u	double	-	1000000000.0
state_bound_vx_u	double	-	15.0
state_bound_vy_u	double	-	5.0
state_bound_r_u	double	-	2.0
state_bound_s_u	double	-	1000000000.0
state_bound_D_u	double	-	1.0
state_bound_delta_u	double	-	0.35
state_bound_vs_u	double	-	50.0

Table 19: Path tracking parameters description

Parameter	Type	Description	Default
input_bound_dD_l	double	-	-20.0
input_bound_dDelta_l	double	-	-10.0
input_bound_dVs_l	double	-	-50.0
input_bound_dD_u	double	-	20.0
input_bound_dDelta_u	double	-	10.0
input_bound_dVs_u	double	-	50.0

Table 20: Path tracking parameters description

6.14 Vehicle model parameters

Parameter	Type	Description	Default
kp	double	PID kp for velocity control.	1.2
ki	double	PID ki for velocity control.	0.0048
kd	double	PID kd for velocity control.	0.0
ts	double	Discretization time (s).	0.001
Cm1	double	Motor model parameter.	5000.0
Cm2	double	Motor model parameter.	172.0
Cr0	double	Rolling resistance parameter.	180.0
Cr2	double	Drag parameter.	0.7
B	double	Pacejka tire model parameter.	10.0
C	double	Pacejka tire model parameter.	1.38
D	double	Pacejka tire model parameter.	1500.0
m	double	Mass of the vehicle (kg).	190.0
Iz	double	Vehicle inertia ($kg \cdot m^2$).	110.0
lf	double	Front distance from center of gravity (m).	1.22
lr	double	Rear distance from center of gravity (m).	1.22
g	double	Gravity constant (m/s^2).	9.81

Table 21: Vehicle model parameters description

6.15 Visualization Parameters

Parameter	Type	Description	Default
rviz_plotter_rate	float	Rate at which RViz updates the plotter, measured in Hertz (Hz). This defines how frequently the visual data is refreshed.	4.0
rviz_plot_lidar	bool	Flag to enable or disable the visualization of lidar data in RViz. If set to True, lidar point clouds will be displayed.	False
rviz_plot_camera	bool	Flag to enable or disable the visualization of camera data in RViz. If set to True, camera feeds and relevant data will be shown.	False
rviz_plot_lidar_processing	bool	Flag to enable or disable the visualization of lidar processing results in RViz. If set to True, processed lidar data will be displayed.	False
rviz_plot_camera_processing	bool	Flag to enable or disable the visualization of camera processing results in RViz. If set to True, processed camera data will be shown.	False
rviz_plot_sensor_fusion	bool	Flag to enable or disable the visualization of sensor fusion results in RViz. If set to True, fused data from multiple sensors will be displayed.	False
rviz_plot_local_frame	bool	Flag to enable or disable the visualization of the local frame in RViz. If set to True, the local coordinate frame will be shown.	False

Table 22: Visualization parameters description