Autonomous Blocking in F1TENTH Racing Using Multi-Raceline Planning and Vision-Based Opponent Detection

* Group 6

1st Okan Arif Güvenkaya
Mechatronics, Robotics and Biomechanical Engineering
Technical University of Munich (TUM)
Munich, Germany
okanarif.guvenkaya@tum.de

2nd Marc Warschat
Automotive Engineering
Technical University of Munich
Munich, Germany
m.warschat@tum.de

3rd Verina Guirgis
Mechatronics, Robotics and Biomechanical Engineering
Technical University of Munich
Munich, Germany
verina.guirgis@tum.de

***Abstract*—This project investigates the problem of autonomous blocking in the context of the F1TENTH autonomous racing platform. The goal is to enable an autonomous 1:10 scale vehicle to actively prevent a following opponent from overtaking on a closed race track. The approach is based on the generation of multiple racelines and the use of a Pure Pursuit controller to track predefined paths. Opponent detection is supported by a rear-mounted camera, allowing reactive defensive behavior. The results demonstrate that strategic raceline selection can be used to achieve effective blocking while maintaining stable and safe vehicle control.**

***Index Terms*—Autonomous systems, autonomous racing, autonomous blocking, F1TENTH, raceline switching, trajectory planning, opponent detection, ROS2, perception-based decision making**

I. Introduction

Autonomous racing research typically focuses on minimizing lap times, with recent work establishing standardized platforms, benchmarks, and control pipelines for autonomous competition [1]. This project addresses Autonomous Blocking, where an ego vehicle actively prevents an opponent from overtaking on a closed track. The task requires integrating perception, decision-making, and control to react to a dynamic opponent in real-time.

The system is implemented on the F1TENTH 1:10 scale autonomous racing platform, a widely used benchmark in autonomous driving research, using a modular ROS 2 architecture [1].. The hardware setup includes a 2D LIDAR for global localization and a rear-facing ZED Stereo Camera for opponent detection using state-of-the-art deep learning–based object detection. Recent advancements in the YOLOv8 architecture have demonstrated improved accuracy and real-time performance, making it well suited for autonomous racing applications [2]. . The software relies on a Finite State Machine (FSM) that analyzes bounding box data to classify the opponent's position. Based on this classification, the system triggers a Pure Pursuit controller to switch between three pre-calculated racelines (Inner, Middle, Outer), physically blocking the overtaking maneuver while maintaining track compliance.

II. System Architecture

The autonomous system is architected as a modular pipeline within the ROS 2 framework. This design prioritizes a clear separation of concerns, ensuring that perception, localization, decision-making, and control operate as independent yet synchronized modules. The overall software stack is illustrated in Fig. 1. The data flow follows a sequential pipeline: raw sensor data are ingested and processed into state estimates, which inform high-level strategic decisions that are translated into low-level actuation commands.

A. Sensor Interface

The vehicle perception relies on two primary sensors. A 2D LIDAR provides 270-degree scan data published to the /scan topic, which is essential for mapping and localization. Simultaneously, a ZED Stereo Camera mounted at the rear captures RGB image streams, publishing to /camera/image raw at 30 Hz. This visual feed is the primary input for identifying opponent vehicles and their relative positions.

B. Perception Stack

Opponent vehicle detection is performed using a real-time, deep learning–based object detection approach operating on stereo camera images. The purpose of this module is to identify the presence of an opponent vehicle and to extract its image-space location, which is then used by the decision-making system to trigger blocking behavior.

The opponent detection node subscribes to the camera image stream published by the sensors module. Visual input is provided by a ZED stereo camera, in which images from the left and right cameras are concatenated horizontally into a single image frame. This configuration enables independent

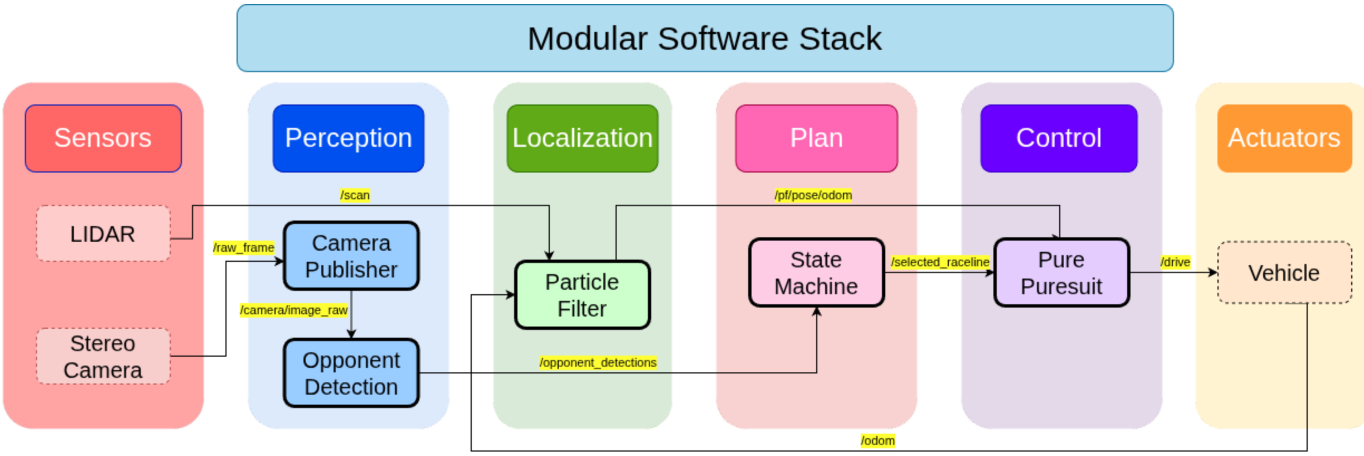

Fig. 1. System-level ROS2 communication architecture showing data flow between perception, localization, decision making, control, and actuation modules. Sensors are depicted as data sources and are not represented as ROS nodes.

perception from each camera viewpoint while maintaining a unified processing pipeline.

Detection is carried out using a YOLOv8 object detection model trained specifically to recognize F1TENTH opponent vehicles. Prior to deployment, a custom dataset was generated by recording videos of opponent vehicles on the track under varying viewpoints and conditions. These videos were converted into image frames and manually annotated in YOLO format using the MakeSense.ai tool. The resulting dataset was used to train the detection model offline, and the best-performing weights were selected for online inference.

During operation, the model processes incoming camera images to identify the opponent vehicle and publishes a Detection2DArray message to the opponent_detections topic. This message contains the bounding box coordinates (center x, y) and dimensions (width and height) in pixel units, which serve as proxies for the opponent's lateral position and longitudinal distance, respectively, as well as a confidence score indicating the detection reliability and a detection identifier labeled opponent_car.

C. Localization Stack

Precise global positioning is handled by a Particle Filter (Monte Carlo Localization). This node fuses the real-time LIDAR scans with a pre-built occupancy grid map and wheel odometry. It estimates the ego vehicle's global pose (x, y, θ) and publishes it to /pf/pose/odom. This accurate localization is critical for ensuring the vehicle adheres to the pre-calculated racelines within the track boundaries.

D. Planning Layer

The process of generating the race line starts by exporting the centre line from the race line optimisation module. The derivative of the reference race line is then computed to determine the local tangent directions at each point along the track. These tangents are then used to calculate the corresponding local normal directions, which serve as the basis for defining the track boundaries. Finally, the race lines are offset along the normal directions at specified percentages of the track width (e.g. 25 per cent for the inner line and 75 per cent for the outer line), resulting in a set of optimised paths that cover the racing corridor and allow for strategic positioning on the track.

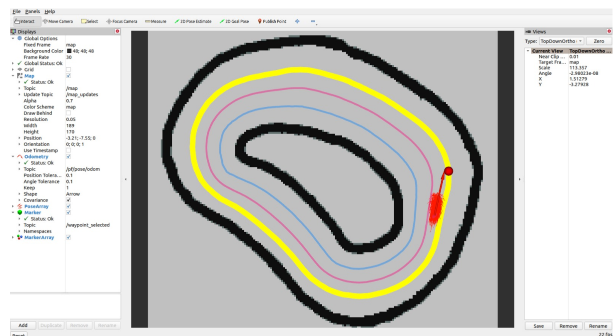

Fig. 2. RViz visualization showing the generated racelines and particle filter–based vehicle localization.

E. State Machine and Decision Logic

The core intelligence of the system resides in the Finite State Machine (FSM) node. This module orchestrates the blocking behavior by translating the visual data from the perception stack into discrete blocking actions.

1) Lateral Logic (Left/Right/Center): The FSM classifies the opponent's lateral position by analyzing the horizontal center (c_x) of the detected bounding box. The camera's field of view is logically divided into three zones using precise pixel thresholds calibrated to the physical track width:

- **Left Zone:** If $c_x <$ Left_Threshold, the opponent is attempting an overtake on the left. The FSM triggers a switch to the **Inner (Left)** raceline to block.
- **Right Zone:** If $c_x >$ Right_Threshold, the opponent is on the right. The FSM triggers a switch to the Outer (Right) raceline.

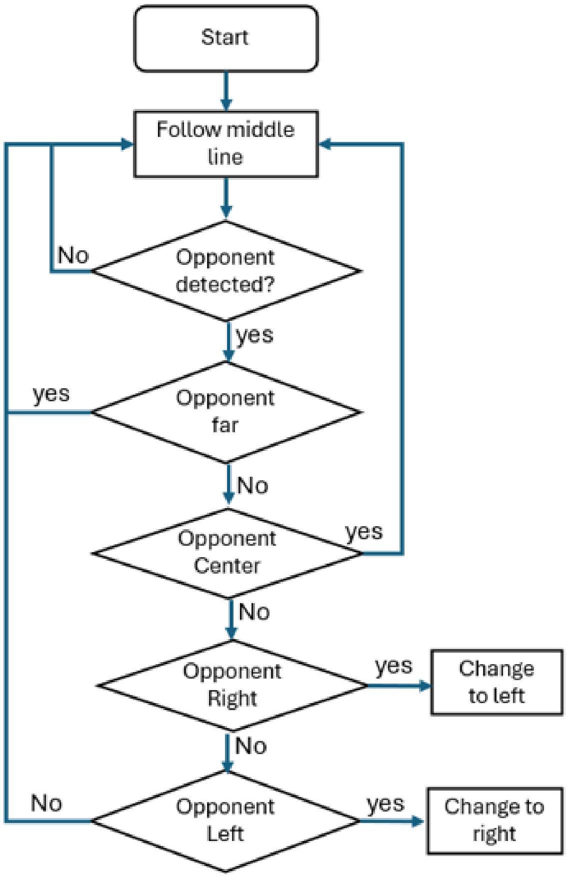

Fig. 3. Decision flowchart of the state machine used for autonomous blocking.

- **Center Zone:** If the opponent is between the thresholds, they are directly behind. The FSM maintains the current optimal line.

2) Longitudinal Logic (Close/Far): To prevent unnecessary maneuvering when the opponent is not a threat, the system estimates proximity using the bounding box height (h_{box}).

- **Close State:** If $h_{box} >$ Height_Threshold, the opponent is considered "Close" (danger zone), and blocking logic is activated.
- **Far State:** If $h_{box} \leq$ Height_Threshold, the opponent is "Far" (monitoring zone), and the vehicle follows the time-optimal center line to maximize speed.

Fig. 3 illustrates the decision flowchart of the proposed state machine. The flowchart summarizes the evaluation of opponent presence, proximity, lateral position, and temporal stability, which together determine whether a blocking maneuver should be executed or the current raceline should be maintained.

F. Control Layer

Trajectory execution is performed by a Pure Pursuit controller. This node subscribes to the selected raceline ID and loads the corresponding trajectory points. It calculates the necessary steering angle to track a dynamic lookahead point on the selected path. The lookahead distance scales dynamically with vehicle speed to ensure responsiveness at low speeds and stability at high speeds. The controller outputs Ackermann steering and velocity commands to the /drive topic, which interface directly with the vehicle's low-level actuators.

TABLE I
KEY CONFIGURATION PARAMETERS FOR PERCEPTION, DECISION MAKING, AND CONTROL

Module	Parameter	Configured Value
Opponent Detection	Detection confidence threshold	0.5
	Image resolution	1344×376 pixels
	Inference hardware	GPU-enabled
State Machine	Raceline change frame threshold	1 frame
	Distance threshold (bbox height)	50 pixels
	Lateral threshold (left boundary)	234 pixels
	Lateral threshold (right boundary)	438 pixels
Pure Pursuit	Minimum lookahead distance	1.2 m
	Maximum lookahead distance	3.0 m
	Lookahead ratio	8.0
	Steering gain K_p	0.25
	Velocity scaling factor	0.6

III. VALIDATION AND RESULTS

System reliability was verified through a progressive strategy moving from open-loop mechanical tests to static calibration of perception thresholds, and finally to dynamic interaction. Initial open-loop tests confirmed the vehicle's mechanical stability during aggressive lane changes triggered manually. Static tests allowed us to calibrate pixel-to-meter logic against ground truth using RViz.

Table I summarizes the key design parameters of the system.

The raceline generation and vehicle localization were successfully implemented. RViz visualizations of the generated racelines and particle filter–based localization are provided in the (Fig. 2).

In the final dynamic experiments, the system demonstrated effective blocking behavior. In baseline runs without an opponent, the vehicle adhered to the time-optimal center raceline. When an opponent attempted overtakes from the left or right, the perception stack accurately identified the lateral offset, triggering the State Machine to switch to the corresponding blocking raceline. Throughout these maneuvers, the Particle Filter maintained accurate localization, and the temporal stability buffer successfully filtered detection noise, resulting in smooth, stable defensive driving without erratic oscillations.

The opponent vehicle was detected with high confidence and accurately localized using well-defined bounding boxes. Based on the state machine logic, the relative position of the opponent vehicle, classified as far or close and left, right, or center,was correctly identified. Representative detection results are provided in the Appendix (Fig. 4–Fig. 7).

The successful real-time operation of the system can be observed in an accompanying video[1].

[1]https://syncandshare.lrz.de/getlink/fiKaXMXrHm1vuo3SxwJcX/FNJN8927.MP4

IV. Conclusion and Future Work

Experimental trials confirmed the system successfully blocks overtaking maneuvers. When an opponent approached from the left or right, the perception stack identified the offset, and the FSM triggered the correct raceline switch. The vehicle maintained localization and control stability during these transitions. In the absence of an opponent, the system correctly defaulted to the time-optimal center line. Future work will focus on integrating predictive motion models to enable anticipatory blocking strategies.

References

[1] Evans, Benjamin David, Raphael Trumpp, Marco Caccamo, Hendrik Willem Jordaan and Herman Arnold Engelbrecht. “Unifying F1TENTH Autonomous Racing: Survey, Methods and Benchmarks.” ArXiv abs/2402.18558 (2024): n. pag.

[2] S. Mupparaju, T. S. Ram, and C. V. R. Reddy, “A review on YOLOv8 and its advancements,” in *Proc. Int. Conf. on Data Intelligence and Cognitive Informatics*, Springer, Singapore, 2024.

Appendix A
Supplementary Results

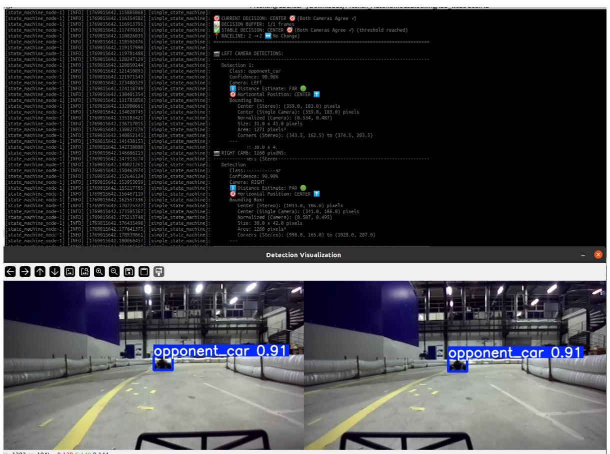

Fig. 4. Opponent vehicle detection in a far-range, center-aligned scenario. The confidence score and state classification are shown in the terminal output.

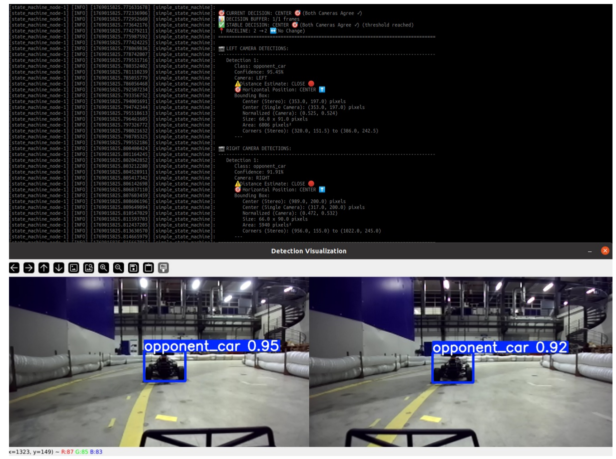

Fig. 5. Opponent vehicle detection in a close-range, center-aligned scenario. The bounding box, confidence score, and state classification are shown.

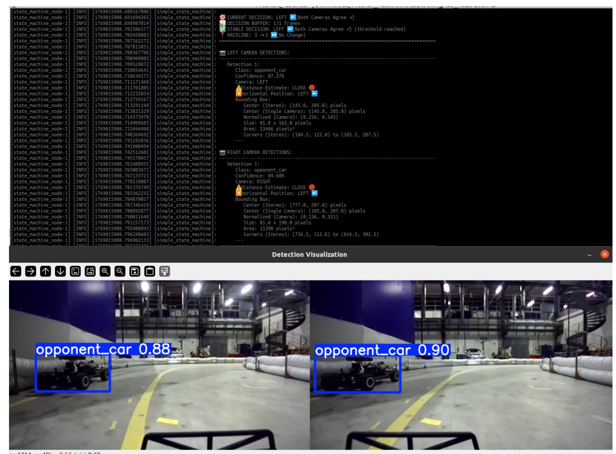

Fig. 6. Opponent vehicle detection in a close-range, left-aligned scenario. The bounding box and lateral state classification are visible in the terminal output.

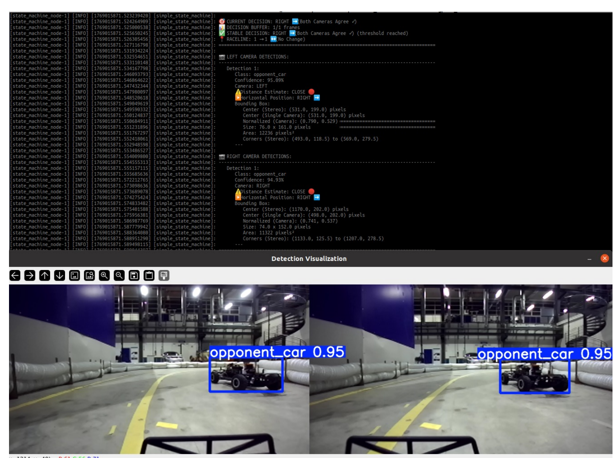

Fig. 7. Opponent vehicle detection in a close-range, right-aligned scenario. The bounding box and lateral state classification are visible in the terminal output.