TED UNIVERSITY

Faculty of Engineering
Department of Computer Engineering
Project Name: F&D Autonomous Driving

Analysis Report

Team Members:
Taylan Kayalı
Can Soysal
Taha Ugan
Çağrı Şanlı
Mustafa Cem Hür

Supervisor:
Tolga Kurtuluş Çapın

Jury Members:
Deniz Cantürk
Hakkı Gökhan İlk

Overview

1. Introduction

1.1 Purpose of this Document

This Analysis Report presents analysis of the F&D Autonomous Driving System project. The report includes the system specification, conditions, cases, and provides a model of the system.

1.2 Project Context

Our team is developing an autonomous driving algorithm that operates in virtual environments. The system must optimize for energy efficiency while maintaining safety and meeting all requirements that are given in the requirements part of this report.

1.3 Scope

In Scope:

- Development of autonomous driving algorithms in Gazebo simulator
- AI/ML model training for perception, planning, and control
- Virtual environment creation and scenario testing
- Getting the most efficient outcome from our model.

2. Current System

Currently, there is no existing autonomous driving system. The system will be built up from scratch by us.

3. Proposed System

3.1 Overview

The F&D Autonomous Driving System is an autonomous vehicle control system. It is focused on energy efficiency. The system operates in a virtual environment.

1. **Environment**
    - Gazebo simulator for development and training
    - Realistic Physics
    - Sensors such as, ultrasonic distance sensor and camera.
    - Environmental conditions like weather
    - Drivable Track

F&D autonomous driving systems optimize safety, speed and energy efficiency.

3.2 Functional Requirements

FR-1: Path Planning

The Model must use the nodes given by the User and compute a path in 10 seconds maximum. The path must be on drivable areas only and the path must be recalculated in case blockages are encountered.

FR-2: Lane Keeping

The system shall detect lane markings and maintain the vehicle in its lane during driving with minimum deviation of 0.30m from lane center. The system must obey line markings legally.

FR-3: Local Path Planning

The system shall continuously generate short term trajectories to avoid any collision at maximum of 3 seconds. Trajectories shall prioritize energy efficiency.

FR-4: Obstacle Detection

The system shall classify objects matching the dimensions 1.10m (H) x 0.45m (D) with a blue color signature as "Track Obstacles" and the dimensions 1.0m (H) x 1.2m (W) x 1.0m (L) with a grey color signature as "Parking Obstacles."

FR-5: Collision Preventation

The system shall maintain safe distances from obstacles and prevent collisions through speed reduction and route adjustment with a success rate of 80%.

FR-6: Speed Control

The system shall track planned trajectories accurately with error of ≤ 0.25m. The system must control its speed accurately to legal traffic speed.

FR-7: Traffic Sign

The system shall detect and obey a standard red hexagon stop sign on yellow board, elevated at 0.5 - 1 meter from the road surface.

FR-8: Localization

The system shall estimate vehicle pose continuously, once every second, with average position error ≤ 0.50m

FR-9: Parking

The system shall identify the parking bay by detecting 0.15m wide blue lines bordered by white lines.

FR-10: Node Navigation

The system shall autonomously visit all target nodes within 3 meters of each node in order and The system must finish visiting nodes in the time limit that we defined for that situation.

FR-11: Road Compliance

The system shall not drive on non-drivable areas. The system shall follow and obey lane markings and road structures. The system shall drive with proper discipline in 85% of the driving time

3.3 Nonfunctional Requirements

NFR-1: Real-Time Performance

The system shall complete the timing requirements. Detection and Depth estimation shall run at ≥ 5 FPS. Short distance planning shall update at least once every second.

NFR-2: Safety Margins

The System shall keep the Lane boundary ≥ 0.20m to avoid crossing it. The vehicle shall emergency stop in ≥ 0.30m before the collision object

NFR-3: Robustness

The system shall handle environmental variations with stable behavior. The System shall tolerate a low level of camera noise.

NFR-4: Testability

The system shall support automated testing with different environments. The system shall track collision, lane violation, completion time, etc.

3.4 Pseudo Requirements

PR-1: Open Source Tools

Prefer open-source tools and frameworks to minimize licensing costs and enable community support.

Tools Selected:

- Gazebo (Simulator)
- ROS2 (Robotics middleware)
- Python with PyTorch or TensorFlow (ML frameworks)
- OpenCV and YOLO (Computer vision)

PR-2: Energy Efficiency Priority

Prefer the more energy-efficient approach,

- Energy-aware path planning
- Smooth trajectory planning
- Deciding how to take the maneuver more efficiently

3.5 System Models

3.5.1 Scenarios

Scenario 1: Basic Route Completion

Actors: Autonomous Vehicle Model
Precondition: Vehicle spawned at start position, route with 3 goal points defined
Normal Flow:

1. Model receives start command and goal point locations
2. Model computes global path visiting all goals
3. Model begins driving, following first path edge
4. Model detects lanes and maintains lane center
5. Model reaches at least 3m near of first goal point
6. Model continues to second goal point
7. Model adjusts path when an obstacle detected
8. Model reaches second goal point
9. Model continues to third goal point
10. Model reaches third goal and stops
11. Model reports completion

Postcondition: All goals visited, vehicle stopped safely, didn't break any traffic rules.
Exceptions:

- 7a: Obstacle blocks path entirely -> System re-calculates the route after updating the graph

Scenario 2: Traffic Sign Compliance

Actors: Autonomous Vehicle Model
Precondition: Vehicle approaching to a STOP sign
Normal Flow:

1. Model detects STOP sign at 10m distance
2. Vehicle reduces speed gradually
3. Vehicle comes to complete stop
4. Model waits 3 seconds
5. Model resumes motion to next target location
6. Vehicle accelerates to planned speed

Postcondition: Model detected STOP sign accurately, The trajectory updated correctly, The vehicle stopped completely, The vehicle resumed driving safely after parking.
Exceptions:

- 1a: STOP sign not detected until 3m -> Emergency brake.

Scenario 3: Obstacle Avoidance

Actors: Autonomous Vehicle Model
Precondition: Vehicle driving in lane at 25 km/h
Normal Flow:

1. Model detects an obstacle ahead at 20m
2. Vehicle reduces speed to 10 km/h
3. Model continues monitoring stopped obstacle
4. Model checks adjacent lane for passing opportunity
5. Adjacent lane clear, no lane markings prohibit passing
6. Vehicle executes lane change
7. Vehicle passes stopped vehicle
8. Vehicle returns to original lane when safe
9. Vehicle resumes to target location

Postcondition: Obstacle passed safely, vehicle in correct lane
Exceptions:

- 5a: Adjacent lane occupied -> System stops before colliding with the obstacle
- 6a: Solid lane marking (no passing) -> System stops behind obstacle
- 9a: Obstacle appears in original lane -> System remains in passing lane until clear

Scenario 4: Autonomous Parking

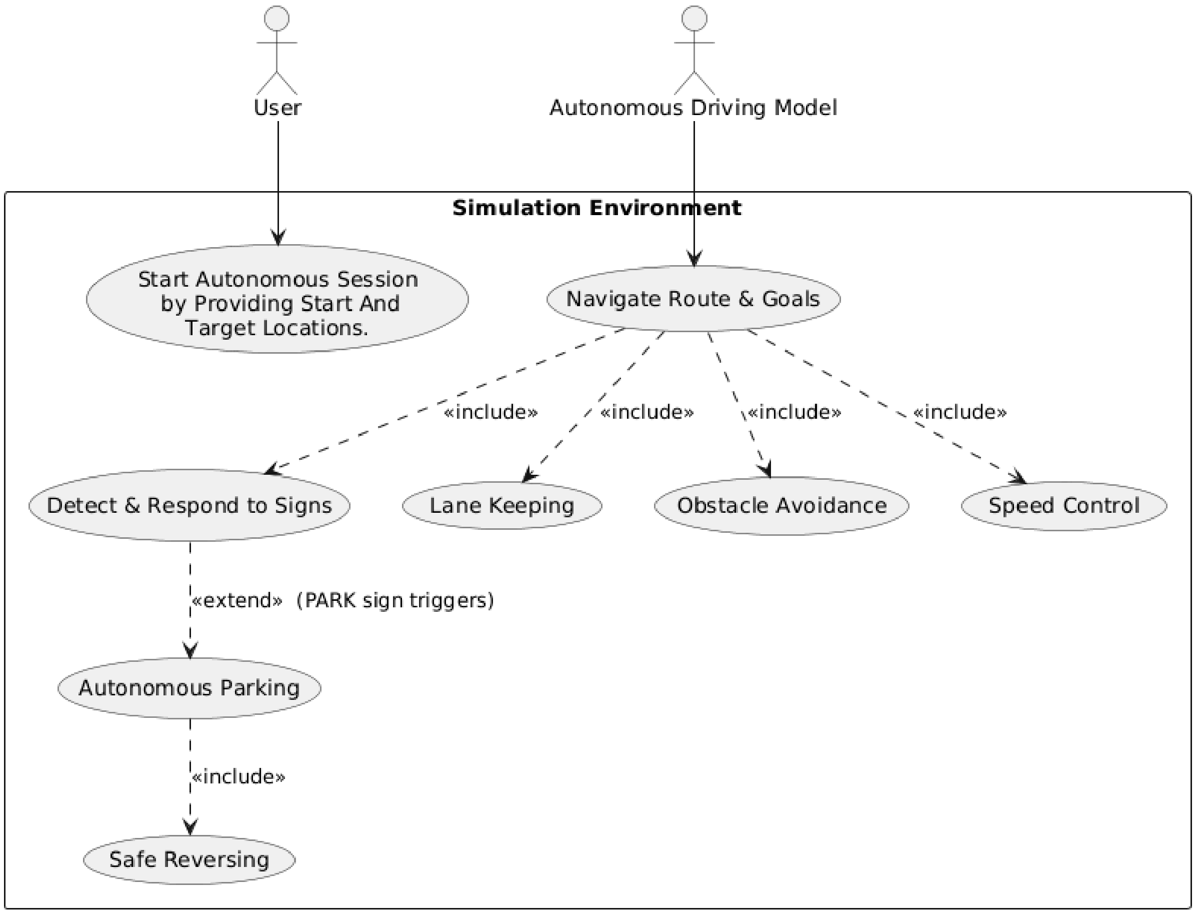
Actors: Autonomous Vehicle Model
Precondition: Vehicle near designated parking spot, parking spot is empty perpendicular space
Normal Flow:

1. Model recognizes parking bay
2. Model checks if the parking bay is available
3. Model navigates to parking area
4. Model identifies parking spot boundaries
5. Model plans the parking maneuver
6. Model aligns vehicle to the spot
7. Model steers into spot
8. Model stops when centered in spot
9. Model verifies position accuracy
10. Model reports parking complete

Postcondition: Vehicle parked within tolerance (≤0.15m lateral, ≤20° heading)
Exceptions:

- 2a: Spot is not available -> Vehicle does not park and passes the parking bay
- 4a: Spot boundaries unclear -> System requests manual spot specification
- 9a: Position error exceeds tolerance -> System adjusts position

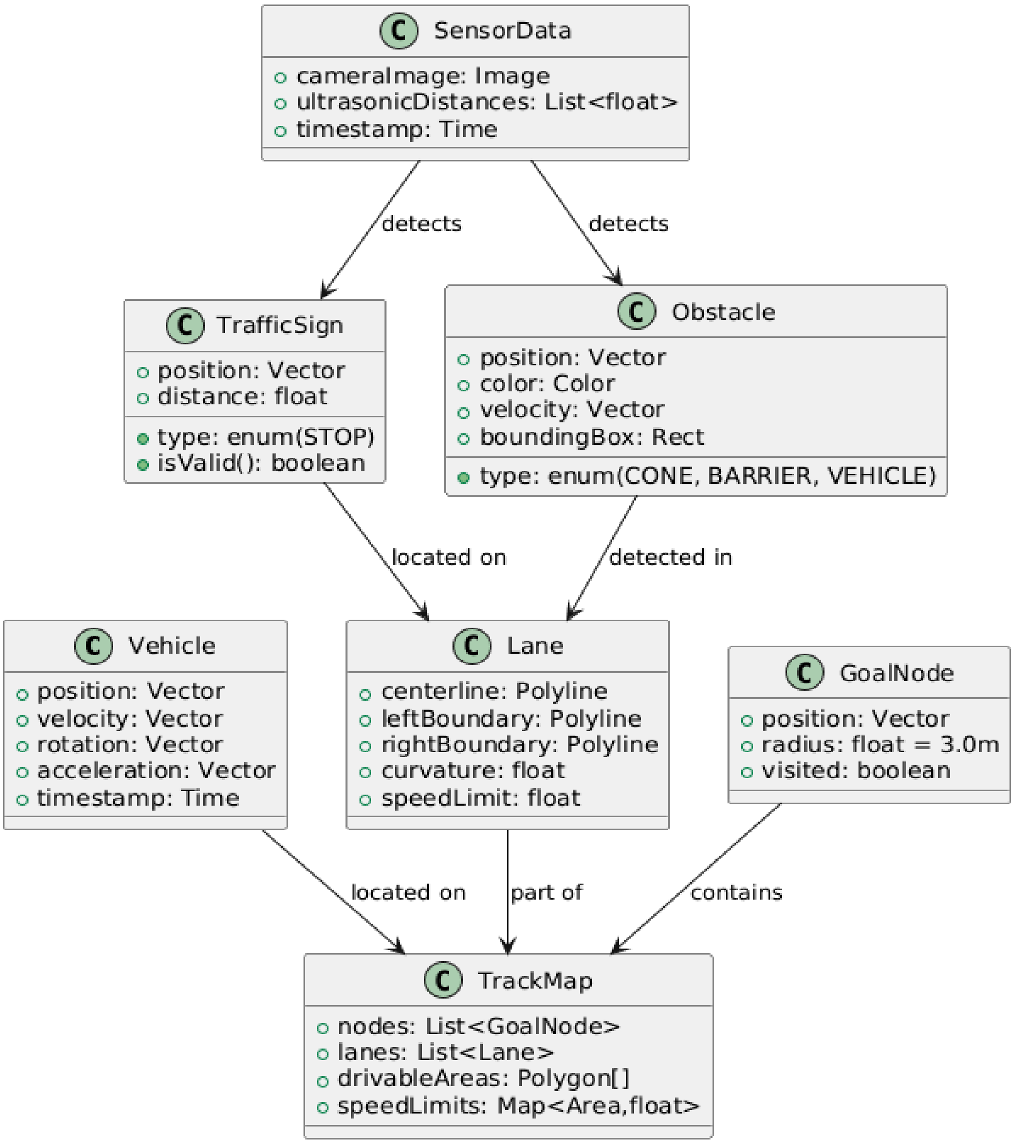
3.5.2 Use Case Model



Use Case Diagram

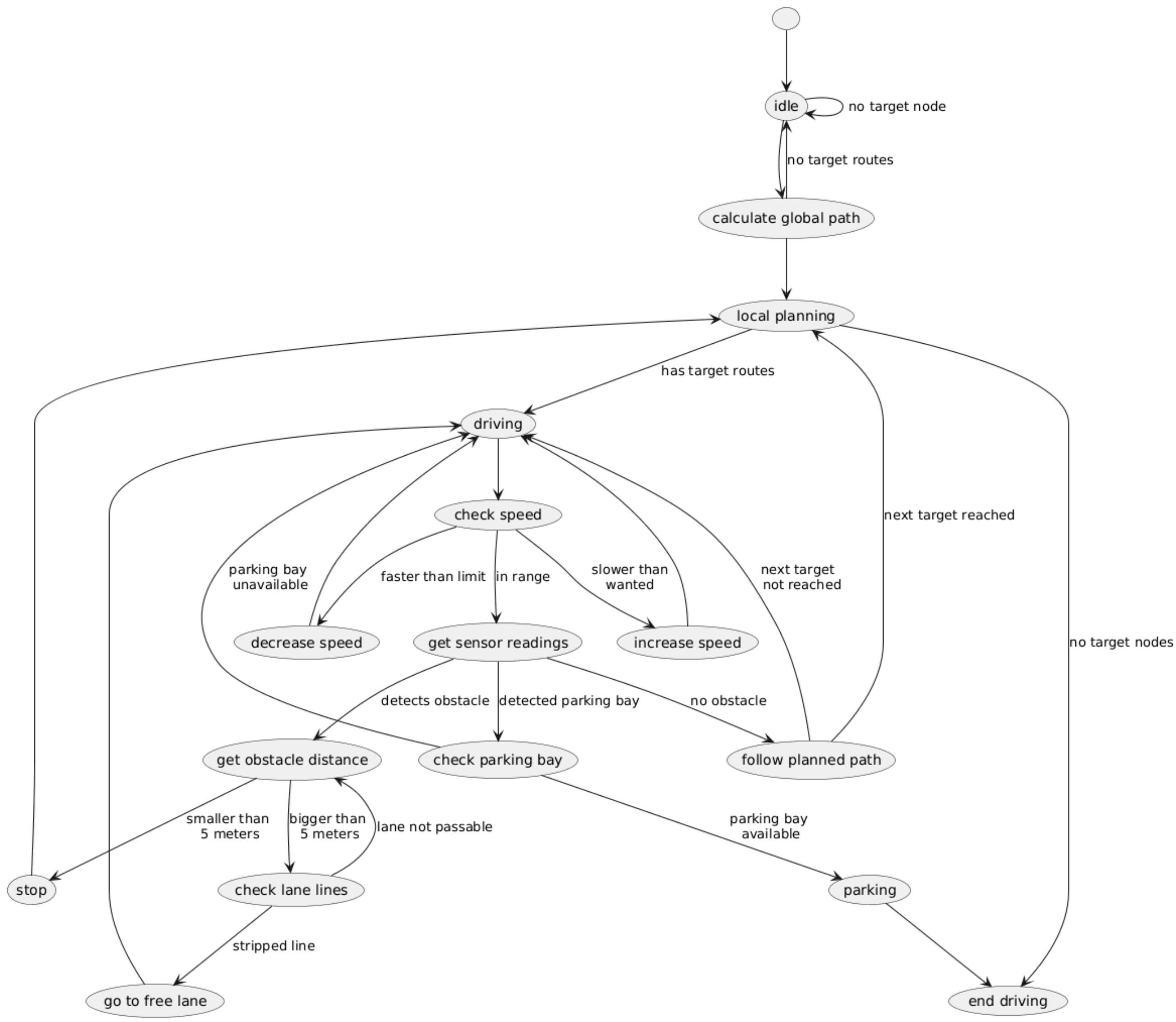
The diagram shows the existing actors and their respective actions within the simulation environment. The user initialises the simulation environment by importing the starting point and the list of target locations. Then the Autonomous driving model calculates the shortest path to pass every target in the shortest time. During the run, if the vehicle encounters any stop sign it comes to a halt for 3 seconds and resumes the run after that. And if a parking sign is detected then the model will park the vehicle in the designated spot for parking with correct maneuvers.

3.5.3 Object and Class Model



Object and Class Diagram

3.5.4 Dynamic Model



Dynamic Model

4. Glossary

Autonomous Driving Algorithm - A system designed to make real time decisions of the vehicle without human intervention, including perception, planning, and control of the vehicle.

Energy Efficiency - The optimization of power consumption relative to distance traveled or task completion, measured in joules per meter or similar metrics; critical for extending vehicle operational range. In our case, it means minimizing unnecessary actions.

Virtual Environment - A simulated representation of the real world for testing the autonomous system safely without physical vehicles or infrastructure.

Path Planning - Using an algorithm to determine optimal route for the vehicle navigation while taking obstacles, blocked routes and efficiency objectives into consideration.

Control Subsystem - The component responsible for the movement of the vehicle such as, brake, gas, steer, park.

Perception System - The computer vision algorithm that detects the objects environment to detect obstacles and signs.

Real Time Performance - Having the system respond to changes and environment almost instantaneously.

5. References

1. N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in * Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 2149–2154

2. S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, May 2022

3. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788

4. A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035

5. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005