Politecnico di Torino

POLITECNICO DI TORINO

Master Degree in Computer Engineering

Automation and Intelligent Cyber-Physical Systems

Master Degree Thesis

Lane keeping in roadwork conditions for a scaled autonomous car

Supervisors
prof. Stefano Malan
prof. Michele Pagone

Candidate
Alessia Sedda

July 2025

Abstract

Driving automation technologies have been gradually penetrating the automotive market. The road infrastructure may be temporarily modified by construction sites, requiring suitable adjustments for the vehicle driving task. This thesis aims to develop the lane keeping task in a simulated diversion scenario induced by roadworks, exploiting vision-based lane detection techniques. The vehicle engaged for the simulation is a 1/10 scaled self-driving car, provided by Bosch for the Bosch Future Mobility Challenge (BFMC). Two specific scenarios have been reproduced: lane change and lane narrowing. Color segmentation, performed by using OpenCV C++ libraries, has been crucial for prioritizing yellow lane markings over white standard ones, and for detecting predefined obstacles. Analysis have been conducted by evaluating the vehicle speed, steering angle and position with respect to the lateral lane line. The diversion has been correctly taken in most of the experiments for both lane narrowing and lane change. The presence of a traffic sign indicating the diversion path, and of red barriers blocking the closed lane, improved the lane change behavior. Partial success has been obtained for obstacle detection when the red barrier is positioned across the lane, forcing the vehicle to stop. Further refinement through additional testing should enhance driving performance, while sensor fusion could improve object detection. Future research should comprise the integration of additional traffic signs and traffic cones to make the roadwork environment more realistic.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Driving automation

Automation refers to the use of electronic or mechanical devices and computer systems to perform tasks that were traditionally carried out by human labor. Driving automation is the execution by hardware and software systems of part or all of the Dynamic Driving Task (DDT), which includes steering, braking, acceleration, lane keeping and monitoring the environment. The Society of Automotive Engineers (SAE), a professional organization with activities in automotive engineering, transport industries, and commercial vehicles including self-driving cars, proposed six levels of driving automation [2] in the context of motor vehicles and their operation on roadways. The level of driving automation is determined by the functional capabilities of the driving automation system, based on the distinction of roles for the DDT between the system and the human user, and the evaluation of DDT fallback performance.

- **Level 0** - no driving automation: the human performs all DDT, even when enhanced by active safety systems. Example features of driver support are automatic emergency braking, blind spot warning, lane departure warning.
- **Level 1** - driver assistance: the driving automation system manages to handle either steering or speed control (but not simultaneously) while the human driver performs the remainder of the DDT, e.g. lane centering or adaptive cruise control.
- **Level 2** - partial driving automation: the human driver supervises the execution of the driving automation system control of both steering and speed.
- **Level 3** - conditional driving automation: environmental detection allows the system to perform most driving tasks but the human user is receptive to feature requests and ready to intervene. A traffic jam feature performs the complete

DDT within dense traffic and requires a human driver to operate when traffic clears.

- **Level 4** - high driving automation: the automated driving features can drive the vehicle not requiring over driving. Pedals and steering wheel may or may not be installed.
- **Level 5** - full driving automation: the features can drive everywhere in all conditions.

Driving automation system features of levels from 0 to 4 are restricted to a context called Operational Design Domain (ODD) defined by a set of conditions which may be environmental, geographical or temporal, e.g. on highways in clear weather or at certain speeds.
Nowadays, current vehicles have some level of driver assistance consisting on, at most, level 2 driving automation features. Goldman Sachs Research sees signs that partial automation and assisted driving are becoming more widespread, even if it has lowered its expectation for Autonomous Vehicles penetration: by 2030, up to 10% (previously 12%) of global new car sales could be Level 3 vehicles [3]. Currently available level 4 vehicles are designed for ride sharing applications, operating within a limited area. Considering the fact that fully autonomous vehicles are far from being achieved, Plebe et al. gave a general overview of imitation learning approaches in the context of autonomous driving, by investigating paradigms inspired by the only existing agents capable of driving: the human beings [4]. Their analysis focused on brain mechanisms including attention, memory organization, emotions, consciousness and many others. Although cognitive- and brain-inspired studies are still marginal, they believe them to be critical to the future of fully autonomous vehicles. Swaayatt Robots, a Bhopal-based startup, centers the research on autonomous driving on the development of level 5 technology via reinforcement learning across adversarial and unstructured environments [5]. They have demonstrated the capability to rely on Global Positioning System (GPS) maps for seamless end-to-end navigation with a series of trials on Indian roads within intricate traffic scenarios. Future of transportation relies on driving automation systems. The growing expectation that full driving automation will become a reality is related to the advancements in artificial intelligence.

1.2 Autonomous driving perception

The environmental perception system relies on the collection of data from multiple sensors. Data processing enables the system to understand its surroundings [6], and make driving decisions to determine the DDT. Visual perception, similarly to human visual perception, provides high resolution images used for lane detection, object recognition [7] and therefore obstacle detection, traffic signs recognition and

traffic light detection. GPS provides global position information allowing localization on maps. Light Detection and Ranging (LiDAR) sensors use laser pulses to provide precise distance measurements by creating a 3D map of the environment. Ultrasonic sensors use sound waves for very short-range detection, commonly used for parking assistance. Data are elaborated by means of a Convolutional Neural Network (CNN), or a Deep Neural Network (DNN), to extract significant information about the surroundings of the autonomous vehicle. It is important that the data elaboration and the decision-making process are completed in few fractions of second to obtain a real-time behavior. Reinforcement learning is a framework that is utilized by an agent for decision making. It can be embodied in hardware, such as an autonomous vehicle, and in this way interact with the real world [8]. The perception-action-learning loop describing reinforcement learning seems to be promising in the context of autonomous driving [5, 9].
Advancements in communication and sensing technologies have paved the way for a connected vehicular environment. A Connected and Autonomous Vehicle (CAV) should perform autonomous driving and continuously communicate with the surrounding vehicles and infrastructure [10, 11].

1.3 Dilemma situations

Autonomous driving may provide benefits considering environmental and safety aspects. It has the potential to optimize traffic flow, while efficient driving reduces fuel consumptions and emissions. Moreover, activities as stop-and-go traffic or long stretches on highway could intensify mental and physical fatigue levels during driving. Driver's ability may be temporarily reduced because of the influence of medication and lack of concentration. Full driving automation gives the opportunity to improve safety, by reducing human error which causes the majority of accidents, and to increase individual mobility, for instance elderly and disabled people could gain independence.

1.3.1 Legal aspects

The authors of [12] analyze the autonomous driving developments from an interdisciplinary perspective, focusing on legal implications and societal impact of self-driving vehicles. Traditional liability and current traffic laws, designed for human drivers, are incompatible with full driving automation. Determining legal responsibility in case of accident, becomes a complex challenge necessitating to account for the roles of manufacturers, software developers and system operators. For compliance with legal requirements, essential incident-related data must be collected with clear and informed user consent and stored for legal clarification purposes.

1.3.2 Ethical considerations

The reaction of the society to the impact of autonomous driving on our roads depends on how autonomous vehicles replicate the human decision-making process. P. Lin [12] discusses the role of ethics in innovation policy: "it can pave the way for a better future while enabling beneficial technologies".
The algorithm of the driving system must be equipped with ethical decision-making capabilities to effectively manage the full range of possible behavior of the autonomous vehicle. This is a difficult task, especially in situations such as an unavoidable crash that could harm people inside and outside the vehicle and a choice between the two groups must be made. The Moral Machine website [13] has been designed to explore the ethics of self-driving cars and understand how humans perceive machine intelligence making such choices. It presents moral dilemmas in which a self-driving car is required to choose the lesser of two evils, by collecting answers from people of different countries. While human drivers make such decisions instinctively in a split second, autonomous vehicles manufacturers have the luxury and responsibility of moral deliberation [14, 15]. Achieving trust and social acceptance of autonomous driving technologies requires their system decision-making process to be transparent, explainable and accountable. A research on acceptance factors from an end-user's perspective of Nastjuk et al. [16] revealed that trust is positively related to usage intentions, and perceived ease of use positively affects attitude towards using autonomous driving. Nonetheless, despite the tremendous advantages autonomous driving offers to individual and society, several user-related factors still need to be addressed.

1.3.3 Cybersecurity and privacy

Potential vulnerabilities in the deployment of autonomous vehicles are cybersecurity and privacy. Autonomous vehicles strongly rely on data coming from multiple onboard sensors and cameras that collect relevant information about the environment. The wireless communication network, that allows different vehicle components to exchange information, may be vulnerable to adversarial attacks, which could alter the behavior of the DDT of the system by corrupting the exchanged information. Any compromise in system integrity could have direct safety consequences.
"Cybersecurity" means the condition in which road vehicles and their functions are protected from cyber threats to electrical or electronic components. Log data are required to support the detection of cyber-attacks and provide data forensic capabilities [17].
In addition to environmental data, autonomous systems also collect private data, personally identifiable information, that may be associated with drivers, passengers, car owners or even passerby who are sensed. Other potentially significant data may include driving habits, user preferences, trip frequency and GPS location history. The improper use of such data and unauthorized access by third parties represent

a privacy risk. Information security may be guaranteed by limiting access rights and encryption, or by defining that only groups of entities can jointly access certain data. Additional surveillance measures, introduced to address potential adversarial attacks, may lead to additional privacy concerns. A trade-off between security and privacy is important to find a balanced approach that ensures technical safety.

1.4 Roadwork zones

Current research on driving automation concentrates on standard highway conditions. The road infrastructure is dynamic and may be temporarily modified by construction sites. There are not many evident data on how autonomous vehicles respond to such situation. It is important for Europe to provide equipped infrastructure to support clean and smart mobility. The novel European project iEXODDUS [18] aims to extend existing Highway-ODDs [17], addressing challenges in roadwork zones, tunnels, urban canyons, and road incident sites, to ensure safe, secure, and efficient automated driving operations. Researchers [19] are requiring that each short- or long-term closure on a public road will respect new protocols and standards for infrastructure construction practice, and it is also needed to develop a standard protocol for autonomous vehicles, which require special pavement marking to move smoothly in the work zone. The presence of roadworks is often indicated by additional traffic signs, cones, red and white barriers, and modified lane markings, which may be absent or, if present, differ in color or quantity compared to the standard configurations. Roadwork scene recognition [20] is essential to quickly detect roadworks and offer manual control to the human driver, or to an alternative navigation algorithm. The environment in roadwork zones can vary significantly, requiring higher levels of driver attention, and negatively impact road users by causing increased journey times, traffic congestion, and reduced road safety. A study using a VISSIM-based traffic micro-simulation, calibrated and validated with real data, compared the performance of CAVs in two roadwork scenarios [10]: lane closure, and narrow lanes. Narrow lanes achieved better safety and traffic efficiency benefits than lane closure. The average delay was lower for narrow lanes, not requiring a mandatory lane change.
In a simulated environment developed to integrate CAVs and roadwork areas featuring lane closure [11], CAVs started lane changing early, demonstrating a high success rate. Current driving automation technologies behavior in these circumstances is still under investigation. Level 2 and level 3 vehicles require human intervention in case of disengagement, which can be caused by technical malfunctions or activated for safety considerations. Entering a work zone can cause a disengagement, require a takeover maneuver and lead to new issues depending on the takeover performance. The driver's takeover behavior can impact on the vehicle and on the traffic flow. Factors which influence the takeover behavior include driver's individual characteristics, environmental conditions, e.g. weather and traffic flow, and

driving system properties. Xu et al.[21] analyzed the impact of level 2/3 driving automation on road safety in high-speed highway work zone environments. With the employment of microscopic traffic simulation and statistical analysis methods, they found that most disengagements in work zones are instigated by small vehicles. So, the increase in the proportion of large vehicles can lead to a reduction in the total number of conflicts. Moreover, increasing work zone speed limits contributes to reduce both single-vehicle and multi-vehicle conflicts. They concluded that current level 2/3 automated driving technology does not contribute to enhanced safety in work zone; under unfavorable conditions it may even adversely impact. In the context of roadwork zone safety, the research group of Dr. V. Saillaja provided a novel strategy to perform construction sites real-time monitoring, by means of the combination of Internet of Things (IoT) with object recognition [22]. The developed system includes IoT-embedded traffic cones, equipped with GPS, light and proximity sensors and cameras to capture real-time data and provide crucial information about the construction zone surroundings. The CNN models, such as YOLOv3 and YOLOv4, perform the object detection to identify items in the work zone and send real-time warnings to the on-site workers, traffic management authorities and site managers, helping in this way to prevent accidents. Autonomous vehicles passing through roadwork zones may be able to leverage information from IoT devices, transmitted instantaneously to the driving system through its wireless communication network. Roadwork information will be integrated into the autonomous driving perception module outlined in Section 1.2. Traffic conflicts and delays at roadworks decrease when CAVs receive additional information in advance via a smart traffic cone respect to relying only on data from in-vehicle sensors [23]. The framework proposed by N. Andrade et al. utilized YOLOv3-tiny for detecting roadworks signs in the traveled path, Deep Deterministic Policy Gradient (DDPG) for controlling the behavior of the vehicle when overtaking the working zones, and CoppeliaSim as simulation environment [9]. DDPG is model-free and uses DNNs to improve the actions of the vehicle, by only relying on experience and trial-and-error. Findings on the results have shown that YOLOv3-tiny is a good choice for real situations, and DDPG achieves the intended goal more than 50% of the trials. Application of the studied methods to real-world scenarios requires simulations to reach very high performance.

1.5 Thesis objective

The goal of the thesis is to examine lane keeping algorithms [24, 25] on a scaled self-driving car and adapt them to a simulation environment of a roadworks scenario in the following circumstances:

- lane change in the presence of yellow lane markings;
- lane change in the presence of yellow lane markings and red barriers;

- lane change in the presence of yellow lane markings and a predefined traffic barrier;
- lane narrowing highlighted by the presence of yellow lane markings;
- stop after obstacle recognition.

Chapter 2

The self-driving car

The 1/10 scale vehicle in Figure 2.1 has been provided by Bosch for the edition 2021-2022 of the international technical competition Bosch Future Mobility Challenge (BFMC), initiated by Bosch Engineering Center Cluj in 2017. The competition invites student teams every year to develop autonomous driving and connectivity algorithms to navigate in a designated environment simulating a miniature smart city [26]. The vehicle is equipped with the required electronic and mechanical components to support driving tasks by leveraging the lane keeping feature. It can be considered a prototype of SAE level 5 driving automation car.

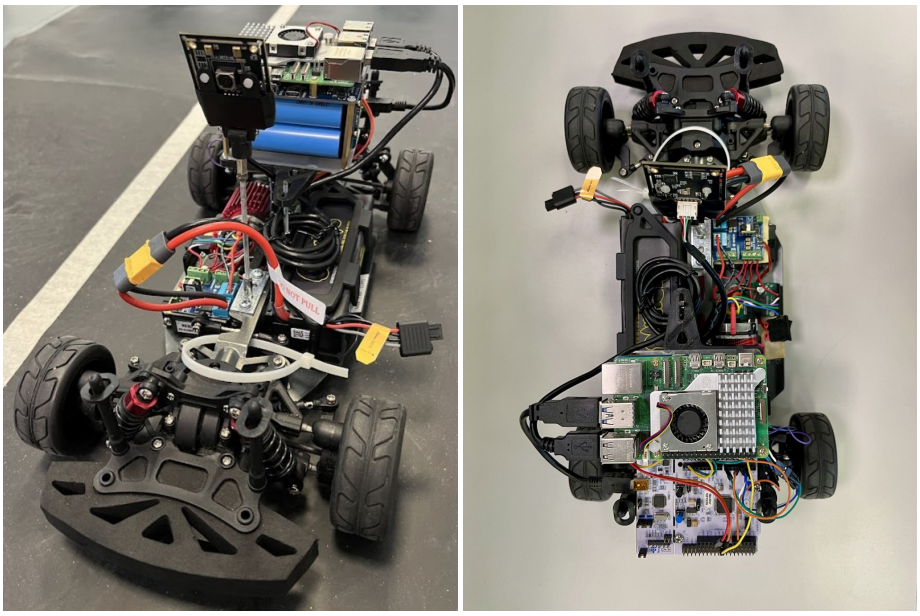


Figure 2.1: The self-driving car

2.1 Hardware

The self-driving car assembled components that have been utilized are represented in the diagram in Figure 2.2, that shows how they are physically connected. Part of the original setup has been modified, as outlined in Section 2.1.1.

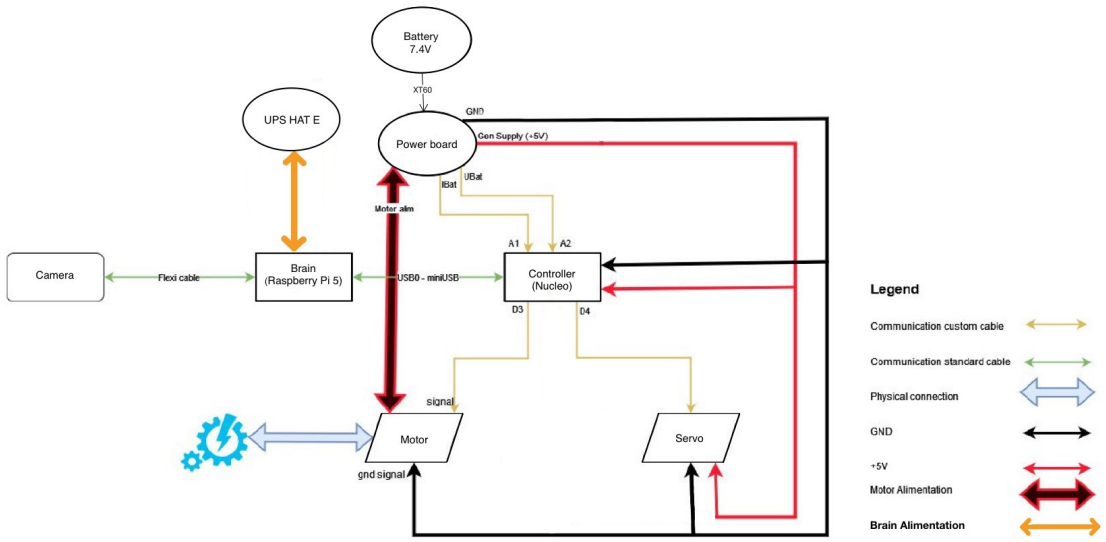


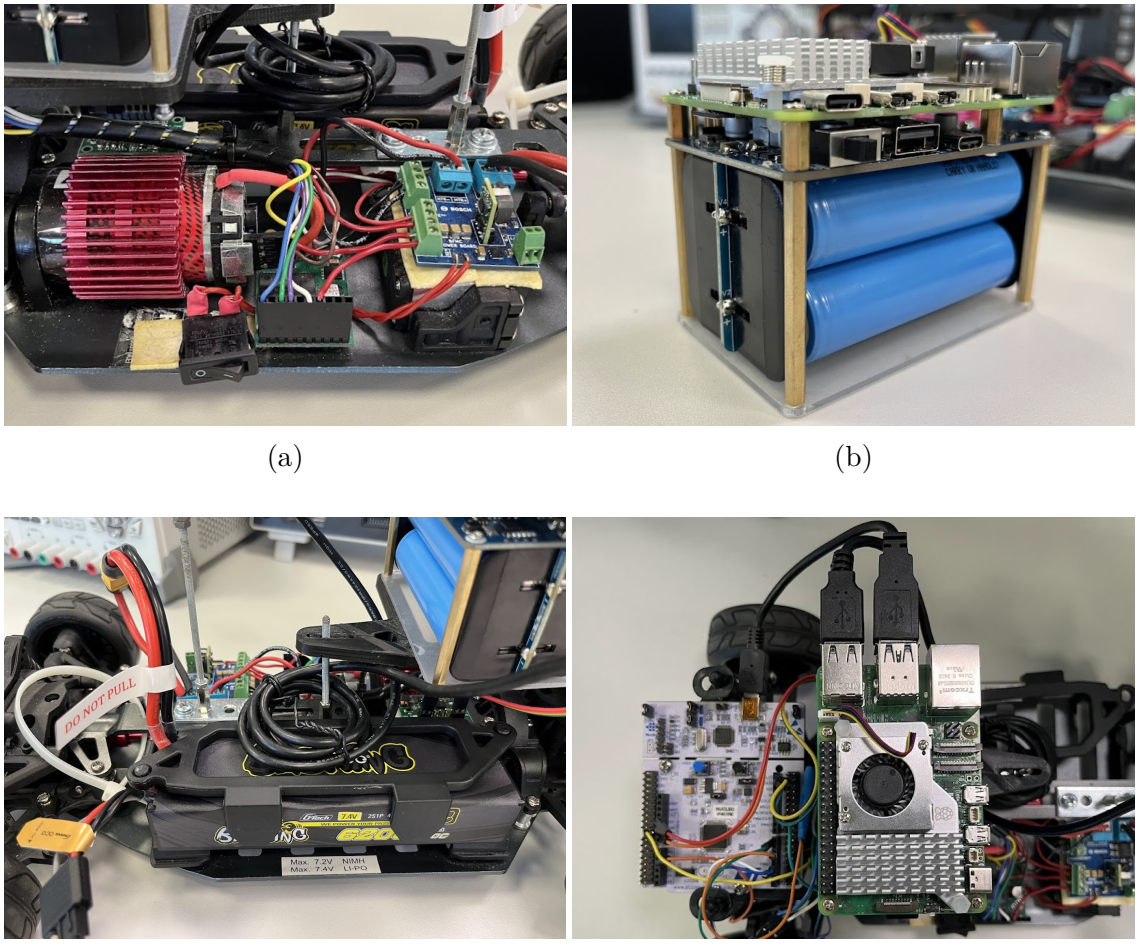
Figure 2.2: Connection diagram

- **Chassis - Reely TC-04:** the completely assembled base structure of the car.
- **Servomotor - Reely RS-610WP:** actuator for the steering feature.
- **Motor - Reely 531009:** on the left side in Figure 2.3a, equipped with modular incremental encoder and H-bridge motor driver.
- **LiPo Battery** (2s, 7.4 V, 6200 mA h, 100 C): used to power up the motors, showed in Figure 2.3c.
- **Power board:** BFMC custom made power distribution board (Figure 2.3a, right side), it delivers the power coming from the connection of the LiPo battery to the motors and returns feedback regarding battery voltage and the instant consumption. To increase the autonomy, an additional battery may be connected.
- **Controller - Nucleo F401RE:** used to control the motors and read data from the power board.

- **Brain - Raspberry Pi 5:** single-board computer that controls the automated part of the vehicle, performing best with its active cooler. It communicates with the controller with a USB0-miniUSB cable (Figure 2.3d).

- **Camera - DFRobot FIT0729:** The main sensor in the vehicle, connected to the brain through USB port.

- **Brain alimentation - UPS HAT (E):** connected to the Raspberry Pi 5 through pogo pins and equipped with four 21700 (5000 mA h, 3.7 V) lithium-ion batteries (see Figure 2.3b).



(a) (b)

(c) (d)

Figure 2.3: In detail (a) motor equipment and power board, (b) Raspberry Pi 5 on UPS HAT, (c) LiPo battery, and (d) connection between brain and controller.

2.1.1 Raspberry Pi 5 power requirement

The original brain for the BFMC, Raspberry Pi 4 Model b, was alimented by the power distribution board. The LiPo battery was indeed used for powering up the entire vehicle. The transition to Raspberry Pi 5, a choice made to improve performance and reliability [24, 25], rendered the battery power supply insufficient. In fact, the Raspberry Pi 5 has a faster quad-core CPU than the previous model and includes a new chip for managing I/O, both increasing power consumption. A voltage of 5 V and a minimum input current of 3 A are required. Using less than 5 A could lead to peripheral instability, which is not acceptable in this application, because the connections with the controller and the camera are made through USB port. The solution that has been selected to power up the brain is the Uninterruptible Power Supply (UPS) HAT type E expansion board, designed for the Raspberry Pi series, with four lithium-ion batteries. The board includes multi battery protection circuits, a battery meter chip, a fast-charging chip and a high-power buck chip, to provide high-power input and output.

2.1.2 Lithium-ion batteries

A battery is a device capable of storing chemicals which can react under different conditions to convert chemical energy into electrical energy [27]. The chemical reaction occurs inside the battery cell, cations flow between the anode and cathode through the electrolyte, generating an electrical current. The electrochemical reactions in a lithium-ion battery are the oxidation of ions, causing the electrons to be ejected from the anode, and the reduction of ions, allowing the electrons to be received in the cathode [28]. In a lithium-ion cell, the cathode is made of a substrate of a thin film of aluminum, the anode is usually made of a thin copper substrate coated with active material, most often graphite. The transfer of lithium-ions from cathode to anode represents the lithium-ion battery charging. The discharging event is represented by the energy flow generated by lithium-ions moving from anode into cathode. Special attention to battery operating conditions is needed to monitor the charge-discharge protocol such that any physical damage, aging and thermal runaways are avoided [29]. Moreover, the battery must be protected from overcharging and over discharging. Lithium-ion batteries are rechargeable batteries which have been widely used for electric vehicle application thanks to their attractive features such as high efficiency, long cycle life, low discharge rate, high voltage, lightweight and better power and energy density compared to other energy storage devices. There are different types of lithium-ion cells, distinguished by their form factors:

- ***Cylindrical cell***: the can, made of steel, nickel-coated steel or aluminum, offers high strength packaging requiring a lot of energy to damage it. However, a high initial impedance causes a high rate of heat generation, easy manageable

due to the small dimensions.

- ***Prismatic cell***: the rectangular shape of a hard steel, plastic or aluminum can is gaining attention from major auto manufacturers. Its major benefit is the low quantity of cells required to achieve the voltage and energy needed.
- ***Pouch cell*** (also called polymer or laminate - LiPo): this type of cell has become the standard in many portable power applications, thanks to its flexible pouch-like design. It uses a soft polymer laminate casing. The term polymer originally referenced to employing a polymer-based instead of a liquid-based electrolyte. The integration of safety features such as cell venting is more challenging for this type of lithium-ion cell.

Four cylindrical cell batteries (21700 format - 21 mm x 70 mm) and a LiPo battery have been used in this application, respectively shown in figures 2.3b and 2.3c.

2.2 The lane keeping algorithm

The research of this thesis leverages the work conducted by C. Gentile [24] and L. Di Biase [25] in 2024 on the development of a lane keeping algorithm based on lane detection and speed and steer control. The C++ algorithm running on the brain acquires frames from the camera, elaborates them and computes commands which are sent to the controller. Lane detection is achieved by means of feature-based techniques aimed to detect lane markings. Multi-threaded programming includes one thread acquiring frames that are collected into a queue, the main thread responsible for frame elaboration and commands computation, and the thread continuously communicating with the controller. The commands sent to the controller are the values of speed and steer, respectively limited in the intervals $[-50,50]\text{cm}\,\text{s}^{-1}$ and $[-25,25]^\circ$.

2.2.1 Image processing

Each acquired frame is restored from the queue and gets elaborated. The captured frame contains information compliant with the Red Green Blue (RGB) color space, working on three different channels. A conversion to the uni-dimensional grayscale color space is performed to reduce the image size, and achieve a better edge recognition. Focusing on white lane markings, a mask is applied to the frame to extract only white pixels. To remove noise from the image a Gaussian blur low-pass filter is used. Bright areas of the image are expanded by applying gap-filling operations, and isolated comparing each pixel intensity with a threshold. Once the frame has been cleaned, the image is transformed from the frontal view into a bird's eye view to eliminate the perspective. The lane lines appearing in the frame are now parallel. Lines are extracted from the image by means of a Hough Transform (HT)

implementation, and classified as belonging to the left or right lane, or to the stop line, in accordance to their slope value. A sliding window approach is used to follow the course of the lane lines and, with the histogram peak detection, their center points are recognized and used to construct a central line, which will be considered the reference line of the trajectory.

2.2.2 Mathematical model

The self-driving car has been modeled as simply as possible with the kinematic single-track model (Figure 2.4). Without accounting for dynamic forces, this model gives good results for non-critical driving situations [30]. The wheels on each axis are considered as a single unit.

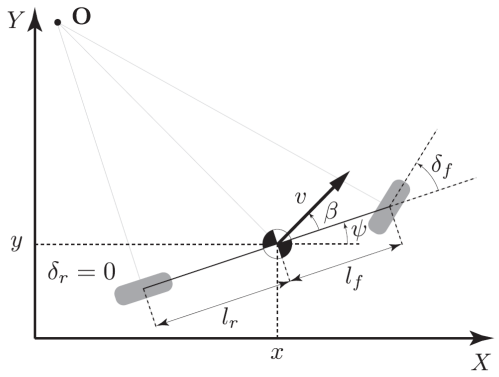


Figure 2.4: Kinematic single-track model

The nonlinear equations describing this model include the definitions of the linear and angular velocities affecting the vehicle longitudinal and lateral motion, with respect to an inertial reference frame (X, Y):

$$\dot{x} = v \cdot cos(\psi + \beta) \tag{2.1}$$

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

$$\dot{\psi} = \frac{v}{l_r} \cdot sin(\beta) \tag{2.3}$$

$$\dot{v} = a \tag{2.4}$$

$$\beta = arctan\left(\frac{l_r}{l_r + l_f} \cdot tan(\delta_f)\right) \tag{2.5}$$

where (x, y) are the coordinates of the center of mass, ψ is the inertial heading (yaw angle, describing the rotation around the vertical axis of the car), v is the speed, β is the angle between the velocity vector and the longitudinal axis of the car, and δ_f is the steering angle.

2.2.3 Control design

The states describing the vehicle behavior in time are the yaw angle ψ of the car and the lateral offset, defined as the distance between the position of the car and the lateral lane line. A gain scheduling control based on ψ has been developed by computing the slope of the reference line, the final result of image processing. The equations of the reference line and the expression of its slope m, which is computed exploiting linear regression, are given by

$$y = mx + q \tag{2.6}$$

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \tag{2.7}$$

where n is the number of points and x and y are their coordinates. The steering value δ_f is defined according to the value of m. In general, if $|m|$ is large, then $|\delta_f|$ is small because the vehicle is already moving along the correct direction. On the contrary, if $|m|$ is small, a large $|\delta_f|$ will bring the car along the prescribed trajectory. The sign of δ_f is negative for left turns and positive for right turns, which is consistent with the sign of m.
The computed δ_f may be updated by taking into account the error due to the lateral offset. The horizontal position on the frame of the reference line, specifically the x-coordinate of the third point belonging to it, is subtracted to half of the frame width, which represents the car position. If $|\delta_f| < 10$, the subtraction result is compared to specific thresholds to compute the modification for the steering value, with a greater correction in case the error is more significant.
The speed value v is set proportionally to $refSpeed$, an initial reference value, with the proportionality factor depending on $|\delta_f|$. The general trend is to allow higher velocity values for small $|\delta_f|$, and to reduce velocity as $|\delta_f|$ increases. For large steering values, the speed is set to the minimum value required to move. If a horizontal line is detected, e.g. a stop line, $v = 0\,\mathrm{cm\,s^{-1}}$ and the car stops. In presence of straight lane lines, the velocity tends to reach the reference speed, while it decreases when approaching a curve.

2.2.4 Time sampling

To achieve real-time performance, the original program has been modified to ensure that frame acquisition, image processing and control are executed within a constant sampling time. Since frame elaboration typically takes 30 ms on average, but may exceed this under certain conditions, a fixed sampling time of $T_s = 50\,\mathrm{ms}$ has been chosen to provide sufficient time for all tasks. During each interval, a frame is acquired and processed, speed and steering values are computed accordingly with image processing results, and the corresponding commands are sent to the controller. Figure 2.5 provides an overview of the implemented logic flow. To improve

performance, frame storage in a queue has been removed, indeed each frame is processed in real time within T_s, making buffering unnecessary. The communication thread responsible for transmitting speed and steering commands to the controller has been removed, since the commands values depend on the processing of the acquired frame. Commands are now transmitted once every T_s. The lane keeping feature has been enhanced to better handle curves, by continuing the reference line generation even in cases where only one lane marking (either left or right) is detected.

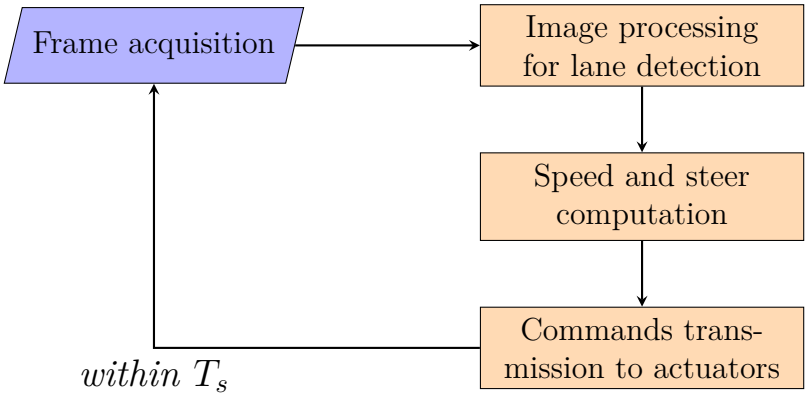


Figure 2.5: Lane keeping algorithm

Chapter 3

Lane keeping in a roadworks scenario

The design of the roadworks environment and the related considerations are based on the Italian road infrastructure and regulatory standards, which may differ from those adopted in other countries. In normal conditions, pavement markings in Italy are white, whereas in other countries the standard color for lane markings may be either white or yellow. Roadworks conditions involve the introduction of temporary yellow lane markings which overlap standard white markings, in certain cases causing uncertainty in determining the correct lane trajectory. Residual markings could induce confusion, even among attentive and experienced drivers. The correct interpretation of lane trajectories may be improved through the combined use of lane markings and physical traffic barriers.
The lane keeping algorithm described in Section 2.2 demonstrates degraded performance when multiple lane colors are present on the track. Specifically, the algorithm is designed to detect two lane boundaries, left and right, for constructing the reference line. The introduction of additional lines, such as overlapping or residual markings, may lead to incorrect boundary identification and, consequently, errors in lane trajectory estimation. For instance, the algorithm may erroneously classify a yellow left-curved lane marking as the right boundary and a residual white marking as the left boundary. The misinterpretation leads to the construction of an incorrect reference line, resulting in inappropriate steering and velocity commands. The car may continue on a straight path rather than initiating the required left turn. The self-driving vehicle must accurately identify the correct pair of lane markings, which delineate the intended driving trajectory. To replicate the behavior of a human driver in roadwork conditions, the algorithm is expected to assign precedence to the yellow lane when present, while discarding leftover white markings.

A clear distinction between white and yellow lanes must be established. It is essential to compute two separate masks according to this distinction, enabling to selectively combine them when appropriate, or to apply a specific one depending on the driving scenario. The selected mask, either resulting from the combination of individual masks or chosen singularly, will be applied to the frame before executing the perspective transformation step.
This thesis investigates a simulated diversion scenario induced by roadworks, concentrating on prioritizing yellow lane markings within the lane keeping system. Two specific scenarios are examined: lane changing and lane narrowing. The former is analyzed under varying conditions, including the presence of multiple overlapping lane markings, the integration of a traffic sign to indicate a leftward deviation, and the addition of red traffic barriers serving the same purpose as the sign in guiding vehicle trajectory. Lane narrowing is represented exclusively through the use of yellow lane markings.

3.1 Image processing for color segmentation

Camera sensors are assuming an increasingly critical role in modern autonomous driving technologies. They provide high-resolution color images that are essential for detecting lane markings, road signs, vehicles, obstacles and other objects present in the driving environment. Image processing supports a wide range of tasks, including lane and object detection, by leveraging software-based methods, and it is fundamental for understanding the semantic features of the surrounding environment. Color image segmentation consists on partitioning an image into several regions that are homogeneous with respect to one or more characteristics [31]. Color is a very significant low-level feature that can be used to extract homogeneous regions related to an object or part of it. The identification of such regions is typically achieved by comparing the color characteristics of each pixel in an image with those of its neighboring pixels.
In this work, from the perspective of the self-driving car, frames captured by the onboard camera are processed to perform color segmentation. This technique is employed to extract information regarding the color of lane markings, and to detect predefined obstacles and traffic signs. OpenCV is utilized as an efficient tool for implementing color segmentation due to its speed and reliability, which are essential for real-time applications.

3.1.1 Color spaces

Frames acquired from the onboard camera are processed using the OpenCV libraries [32]. In OpenCV, the default color representation format is the RGB color space, which maps the RGB color model to human perceivable color. The trichromatic color vision is based on three primary additive colors - red, green, blue - and their

combinations on the eye cone photo receptors produce the full spectrum of colors perceived by humans. Each pixel in a frame is composed of three numerical values corresponding to the intensity levels of the red, green and blue color channels. A RGB image uses 8 bits per channel, therefore R, G, and B range from 0 to 255 in value. The conversion of a RGB image to a grayscale image is the transformation in terms of the standard matrix notion given by

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

The grayscale image is obtained by keeping its luminance channel Y, containing most of the signal energy, and discarding IQ channel chrominance, which carries information with much less energy.

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{3.2}$$

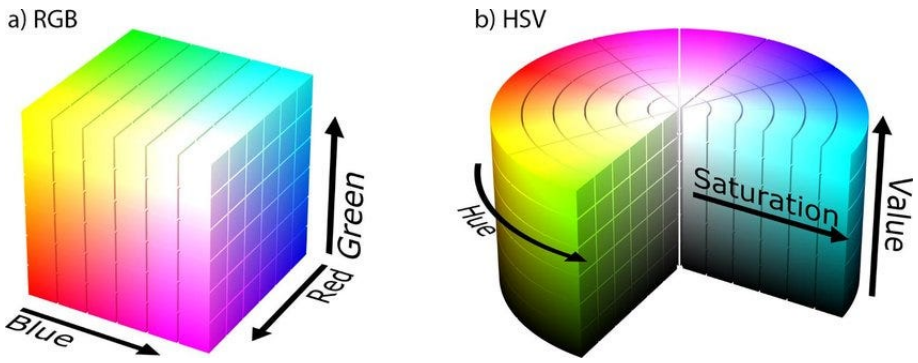


Figure 3.1: RGB vs HSV [1]

In the Hue Saturation Value (HSV) color space, points of a RGB color model are represented by cylindrical coordinates, where the hue is the rotation angle with respect to the central vertical axis, the saturation is the distance from the vertical axis and the value, or brightness, is the distance along the height of the cylinder. Conversion from RGB to HSV is given by

$$V \leftarrow \max(R, G, B) \tag{3.3}$$

$$S \leftarrow \begin{cases} \frac{V - min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$$H \leftarrow \begin{cases} 60\frac{G-B}{V-min(R,G,B)} & \text{if } V = R \\ 120 + 60\frac{B-R}{V-min(R,G,B)} & \text{if } V = G \\ 240 + 60\frac{R-G}{V-min(R,G,B)} & \text{if } V = B \\ 0 & \text{if } R = G = B \end{cases} \tag{3.5}$$

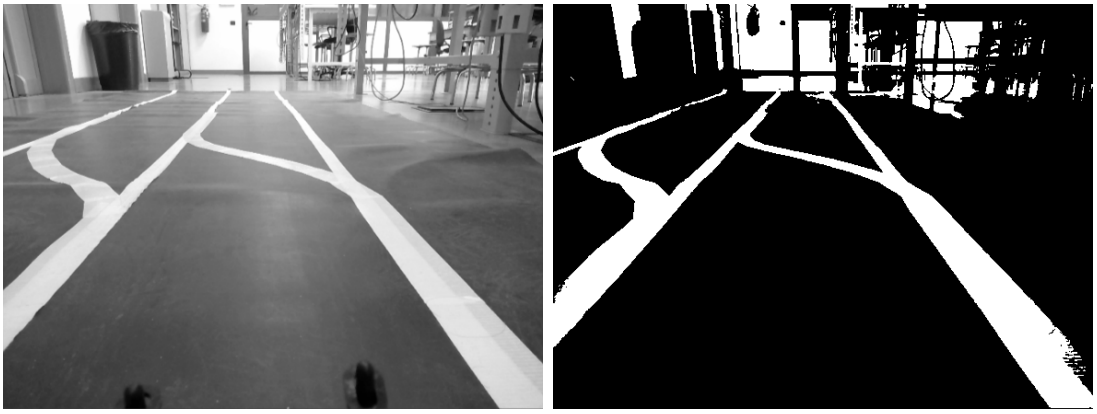
For 8-bit images, the values are then converted to the destination data type to fit in the interval [0,255].

$$V \leftarrow 255V, S \leftarrow 255S, H \leftarrow H/2 \tag{3.6}$$

A comparison between RGB and HSV color spaces is shown in Figure 3.1. While the RGB color space is useful when pure colors are present in the image, the HSV color space exploits the purity of color (saturation) in images presenting different shades of colors.

3.1.2 Mask computation for lane markings

In the grayscale color space, differentiating between yellow and white through mask creation is challenging due to their very similar luminance values. Moreover, variations in lighting conditions can further affect these values, reducing the reliability of grayscale-based color discrimination. As a result, the white mask generated by the grayscale frame tends to include also yellow markings (Fig. 3.2).



(a) Grayscale converted frame (b) White mask from grayscale image

Figure 3.2: Grayscale mask extraction

The HSV color space has been chosen to differentiate among yellow and white markings, since it separates color information from intensity, providing greater robustness to variations in lighting conditions. It is clear from Figure 3.3 that conversion from RGB to HSV allows the yellow and white lines to be easily distinguished. In particular, it has been possible to generate a yellow mask that exclusively isolates yellow pixels, together with a white mask that effectively excludes them (Figure 3.4). The original frame has been converted from RGB to HSV format. Within the HSV color space, two binary masks, each matching the dimensions of the original image, have been derived to highlight yellow and white areas. The threshold values

for the different colors have been defined as follows:

$$maskRange \leftarrow \begin{cases} H = [0; 242], S = [0; 12], V = [200; 255] & \text{for white} \\ H = [20; 30], S = [25; 255], V = [200; 255] & \text{for yellow} \end{cases} \tag{3.7}$$

Pixels of the computed masks set to 1 indicate the region of interest in the image. The role of these masks will be detailed in Section 3.1.4.

(a) Original captured frame (b) HSV converted frame

Figure 3.3: RGB to HSV conversion

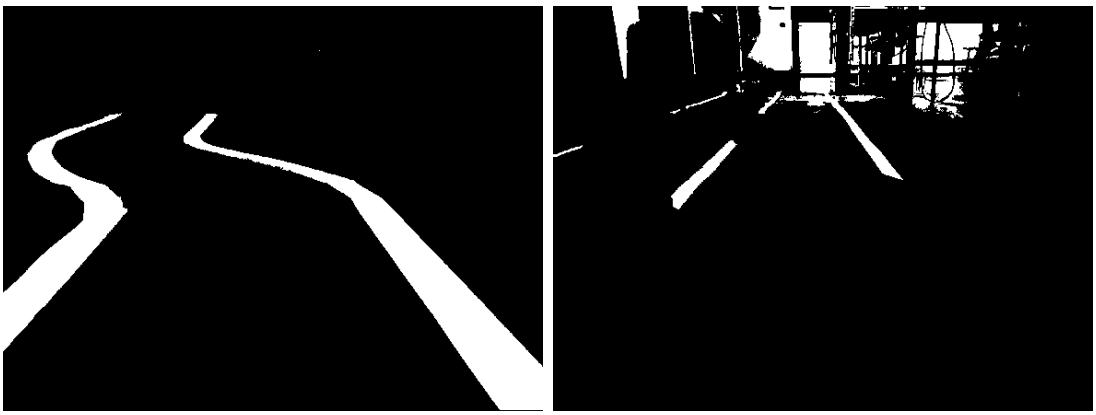
3.1.3 Color calibration

Indoor illumination can be affected by several factors that change lighting conditions. The amount of daylight entering a space fluctuates throughout the day and depends on weather conditions. Artificial light is influenced by its type, intensity and positioning. Additional factors include environmental changes such as movement of people, opening or closing doors and windows. Given these considerations, the selected threshold ranges may not be optimal in varying environmental or illumination conditions. A *maskCalibration* function has been developed to modify the HSV default threshold bounds established in Section 3.1.2. The void function receives as argument an integer number n, which specifies the target color (with 1 corresponding to white and 2 to yellow). Based on the value of n, the function sets the reference HSV threshold bounds according to equation (3.7). The function displays three separate windows to enable real-time visualization of the threshold adjustments effects: the first shows the original frame captured by the camera, the second presents the binary mask highlighting the pixels that fall within the current bounds, and the third window provides interactive sliders for modifying the H, S, and V thresholds. The function is called and executed before initiating the self-driving mode to guarantee accurate recognition of lane markings by verifying that

yellow and white markings are correctly distinguished. High V thresholds are effective in filtering out light reflections on the track surface, which constitute additional noise components that are rarely attenuated by the denoising methods presented in Section 2.2.1. However, this function needs some sort of automatic improvement to enable online calibration (while driving), which could be an appropriate choice for real autonomous vehicle applications. The static approach adopted here is not appropriate, because requiring preliminary computation of the correct bounds.

3.1.4 Lane selection

The number of yellow pixels has been determined by counting the non-zero elements in the yellow mask in Figure 3.4a. The same procedure has been applied for the white mask in Figure 3.4b. After the frame has been converted to grayscale to enable the required transformations for lane detection, a mask has been applied to enhance the visibility of lane lines. The selection of the appropriate mask to be applied is based on the count of detected yellow and white pixels. In the absence of a sufficient number of yellow pixels, suggesting white lane markings are still visible on the track, both yellow and white masks are merged to support lane detection. The predominance of yellow lane markings indicates the beginning of a roadwork zone, requiring the exclusive application of the yellow mask to the grayscale image to prioritize the temporary roadwork markings over any remaining white lines. The selection process is depicted in Figure 3.5.



(a) Yellow mask (b) White mask

Figure 3.4: Final masks obtained exploiting the HSV conversion of the original frame

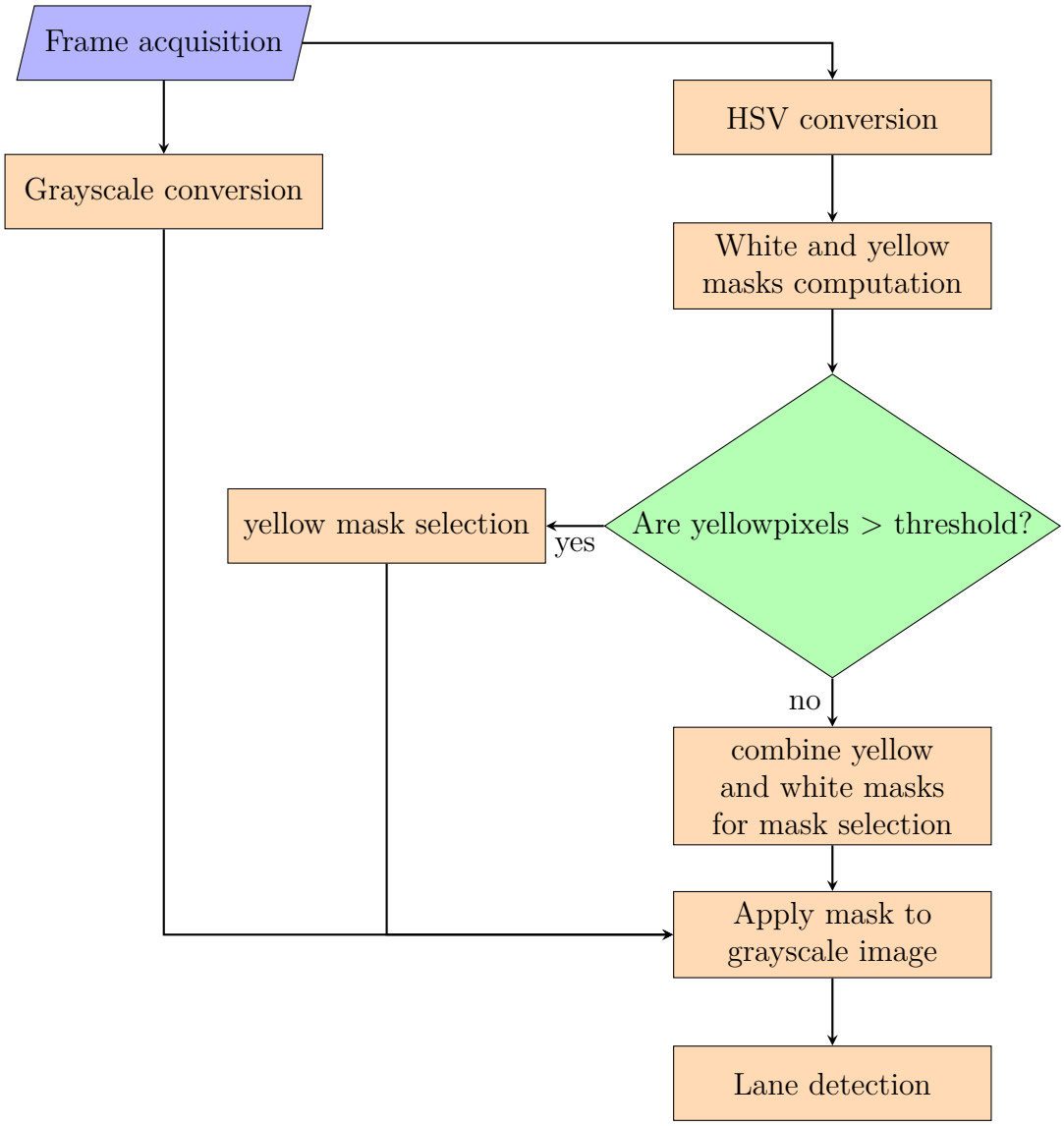


Figure 3.5: Flowchart summarizing mask choice

3.1.5 Object detection

The complex vision task of object detection is approached using a contour-based method, with image segmentation serving as a preliminary step. A distinct binary mask is computed for each of the two target objects, which are a rectangular yellow warning sign displaying red directional arrows and a red barrier. The method involves searching for contours within the binary mask image. For each detected contour, the area is calculated and compared against predefined thresholds to estimate the distance from the obstacle. The contour area is inversely proportional to

the distance, and the thresholds are experimentally calibrated to ensure accurate distance estimation.

Traffic Red-Yellow Barrier (TRYB)

To determine the pixel range, required to extract the region related to the TRYB, threshold-based segmentation is employed. The initial approach involved combining red and yellow color masks, by considering a different yellow range from the one used for lane markings. While this combination was partially effective in isolating the TRYB, the resulting mask contained blank regions that compromised the accuracy of contour extraction and subsequent analysis. A manual calibration phase was then performed using the *maskCalibration* function, initialized with default HSV values corresponding to the red color. The HSV parameters were iteratively refined by adjusting interactive sliders until the target sign was distinctly and reliably segmented. Final default thresholds for the TRYB are given by

$$rangeTRYB \leftarrow \Big\{H = [0; 23], S = [74; 158], V = [147; 255] \tag{3.8}$$

The subtraction of the yellow mask corresponding to lane markings ensures that these features are excluded from the TRYB mask, thereby preventing interference during the segmentation process.
The gain scheduling control strategy has been modified to amplify the computed steering command by 20% when the vehicle is sufficiently close to the detected TRYB. Additionally, the vehicle speed is progressively reduced as it approaches the TRYB. Figure 3.6 shows detection of TRYB in a frame captured from the onboard camera.

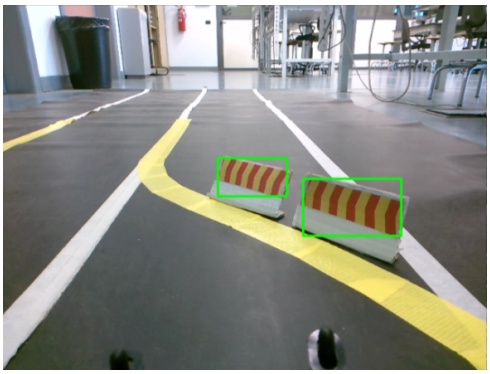


Figure 3.6: Detection of roadworks traffic signs

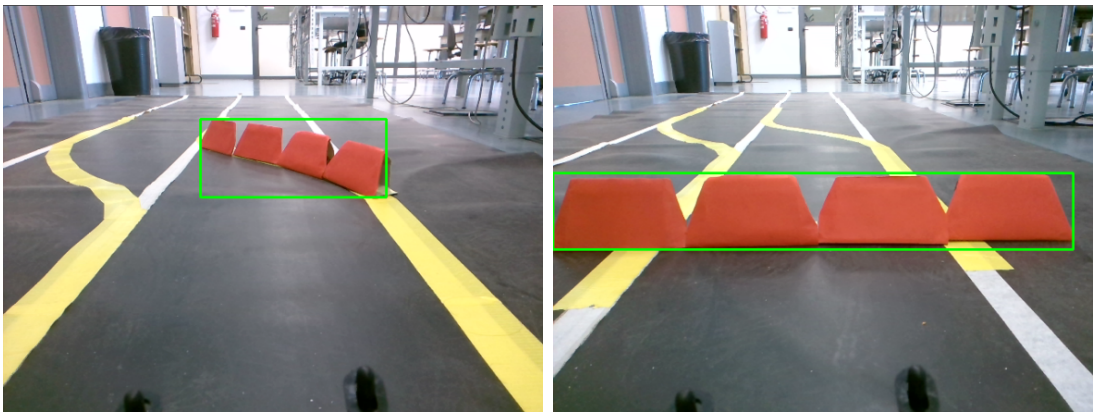
Red barrier

The red obstacle mask is easier to obtain compared to the mask for the TRYB, due to the homogeneous color distribution of the object. In fact, a single red mask

is sufficient for segmentation, by considering the following threshold bounds as reference

$$rangeBarrier \leftarrow \Big\{ H = [0; 10], S = [59; 255], V = [142; 255] \tag{3.9}$$

The *maskCalibration* function is modified to support threshold adjustments for the target objects, with $n = 3$ corresponding to the traffic sign and $n = 4$ to the red barrier. The function assigns the corresponding reference HSV threshold bounds as defined in equations (3.8) and (3.9), respectively.
The gain scheduling control strategy has been adapted to handle three specific situations, each managed by dedicated threshold settings, considering as obstacle a group of four red traffic barriers. If the obstacle is aligned with the curvature of the yellow markings (shown in the following chapter in Figure 4.10a), its detection enhances the lane-change behavior in accordance with the diversion path by means of an amplification of the computed steering command by 20%. In the second scenario (Figure 3.7a), the four barriers are positioned over the yellow curvature markings. The absence of both yellow and white lane markings, replaced by traffic barriers, may pose challenges to the lane keeping algorithm that rely on their presence. In fact, the steering value is set inversely proportional to the detected barrier area regardless of the slope of the reference line. In both first and second situations, the vehicle speed decreases as the perceived obstacle area increases. In the final scenario, when the obstacles are positioned across the lane and obstruct the vehicle trajectory (Figure 3.7b), the speed is explicitly set to zero as the vehicle approaches closely. Area threshold required to stop is 35000, corresponding to a distance of 33.5 cm. For all detected areas above this value, a speed command value of 0 is sent to the controller. When the obstacles are removed, the car resumes motion, following the correct lane trajectory.



(a) (b)

Figure 3.7: Detection of red barriers

Chapter 4

Results

The self-driving car runs on a 6 m long black track. The environment designed for the roadworks simulation consists on the integration of yellow markings in combination with white markings to follow the pattern of a diversion. Part of the tests involve the presence of predefined obstacles, TRYB and red barrier, which have the function of signaling lane closure. Results will be analyzed by evaluating speed and steering commands, lateral offset and slope of the reference line over time. In general, acceptable intervals for the variables of interest are given by

- **Slope** $m \in \mathbb{R}$
  For vertical lines m is not defined. As $|m|$ increases, the line becomes steeper and may visually approximate a vertical orientation. Ideally, $|m|$ is large for straight roads if the longitudinal axis of the car is positioned parallel to the lane lines for the start.

- **Offset** $[-320,320]$ is the maximum interval, which distance (640) corresponds to the width of the image captured by the onboard camera in pixels. The optimal solution for lane keeping is offset equal to 0. An acceptable interval could be $[-100, 100]$, meaning the offset is under the 31.25%. More reasonably, offset varying in $[-25,25]$ establishes good performance.

- **Steer** Physical constraints of the servomotor limit the steering angle in the interval $\delta_f = [-25,25]^\circ$. Higher $|\delta_f|$ commands are not effectively obtained.

- **Speed** It remains nearly constant across the majority of experiments, with a slight decrease in response to increased curvature. It ranges in the interval $[0.0,0.1]$, while its precise conversion to physical units ($\mathrm{cm\,s^{-1}}$) depends on track conditions, such as surface smoothness and the presence of dust or minor dirt that introduce additional friction.

Time sampling results

A fixed sampling time of 50.00 ms has been defined as the system target to ensure timely execution of image processing, control computation, and command transmission to actuators. The experimental results showed an average execution time of 50.06 ms, with the first iteration excluded due to its consistently higher duration of approximately 200.00 ms. The obtained results indicate a close adherence to the intended timing constraint.

4.1 Straight road

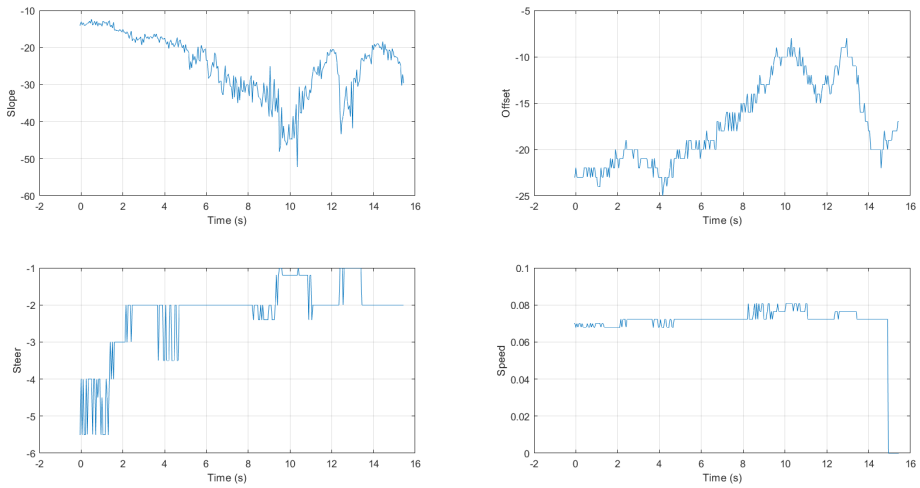


Figure 4.1: Test on straight road

First analysis have been conducted for the evaluation of the lane keeping behavior in a standard straight road exclusively characterized by white lane markings. For optimal lane keeping achievement, minimal offset and high slope (in absolute value) are required. Figure 4.1 illustrates driving behavior observed during a test in which the car follows a straight trajectory, starting with an initial heading angle $\psi > 0$. In particular, the bottom left plot shows how the steering value δ_f is changing depending on the reference line slope m (top left graph) and lateral offset (top right graph). In general, small δ_f corrections have been adopted to increase $|m|$ and decrease the offset. Speed value is almost constant and reaches 0 in occurrence of a stop line, enabling the car to reach the end of the track in 15 s. A comparison of different successful tests, starting from different positions with respect to the lane, is given in Figure 4.2. In detail, the green and blue lines refer to tests in which the vehicle initial position required a larger initial steering correction due to smaller

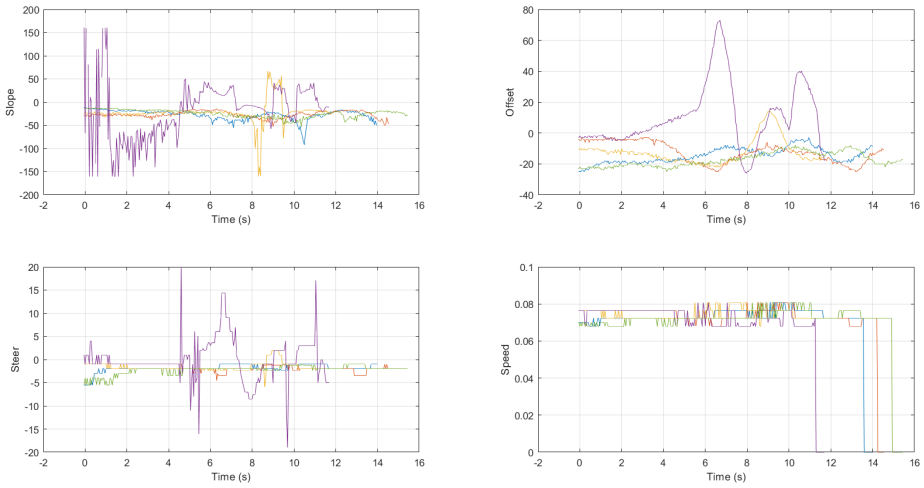
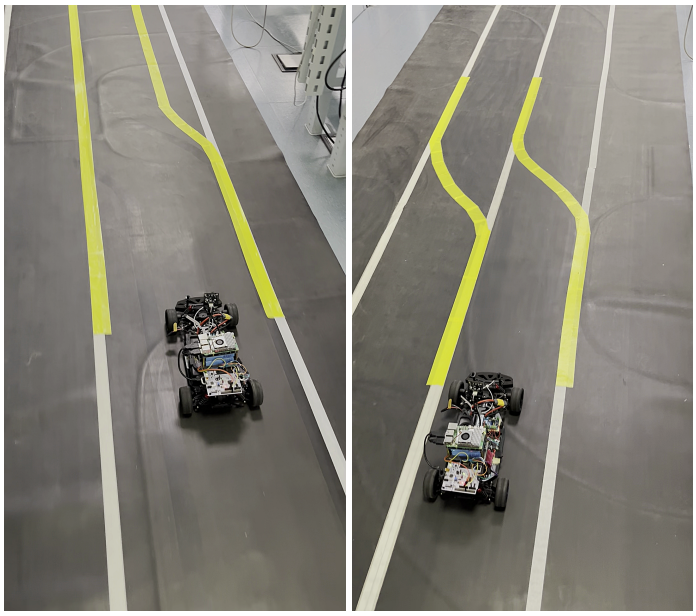


Figure 4.2: Comparison of tests on straight road

slope values and higher offset values compared to the other tests. Conversely, the yellow and red tests exhibit smaller offset values and larger slope values, resulting in reduced steering effort to maintain the correct trajectory, as indicated by the variations in steering commands. The purple plots indicate an apparently optimal initial position, characterized by very high slope magnitudes (ranging from 100 to 150), with oscillations resulting from the vehicle close alignment with a vertical line direction ($m \to \pm\infty$). Due to the improved initial positioning, the vehicle maintains a higher constant speed, which decreases its controllability and requires rapid corrective steering to remain within the lane boundaries. The resulting trajectory exhibits a zigzag pattern. From the comparison, it can be observed that the car reaches the stop line at the end of the straight path more quickly when it starts with a small yaw angle, indicating that its longitudinal axis is already parallel to the lane and reference lines, and centered between the two lane boundaries. In contrast, misalignment at the start leads to increased time to reach the end of the track.

4.2 Lane narrowing

Testing environment for the lane narrowing scenario has been prepared to simulate the narrowing of a 50 cm wide lane into a lane of width 35 cm (Figure 4.3a), respectively demarcated by white and yellow lines. Plots in Figures 4.4 and 4.5 describe the driving behavior of the self-driving car during tests. In the former, two main steering actions are sufficient to achieve the successful result: turning left when the lane width has been decreasing, then right when the final width has been



(a) Lane narrowing (b) Lane change

Figure 4.3: Temporary lane markings for roadworks

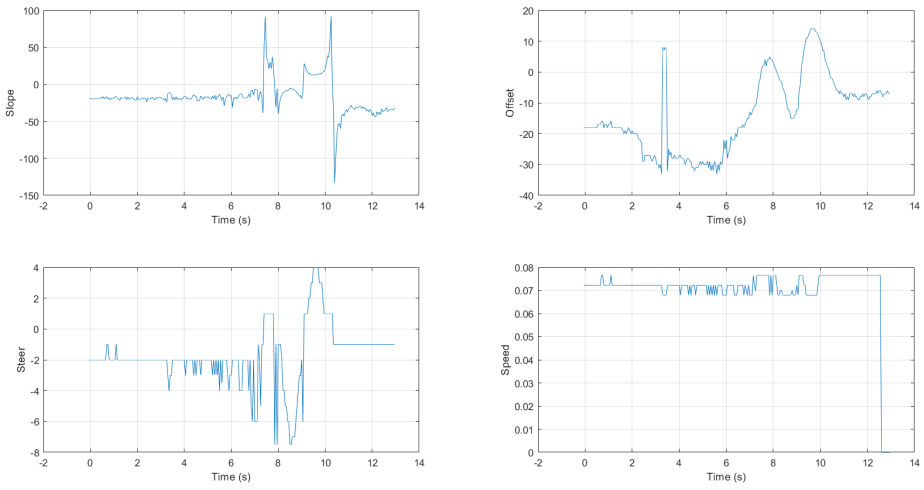


Figure 4.4: Test 1 for lane narrowing

reached. The car stops at the stop line requiring almost 13 s, and the offset always remains within the interval $[-40,20]$, corresponding to keep it under the 20% of its maximum range. In the latter experiment, lane keeping achievement is obtained through a zigzag initial movement, since the car is starting to run from a position closer to the right lane line. Higher steering values, with respect to the previous

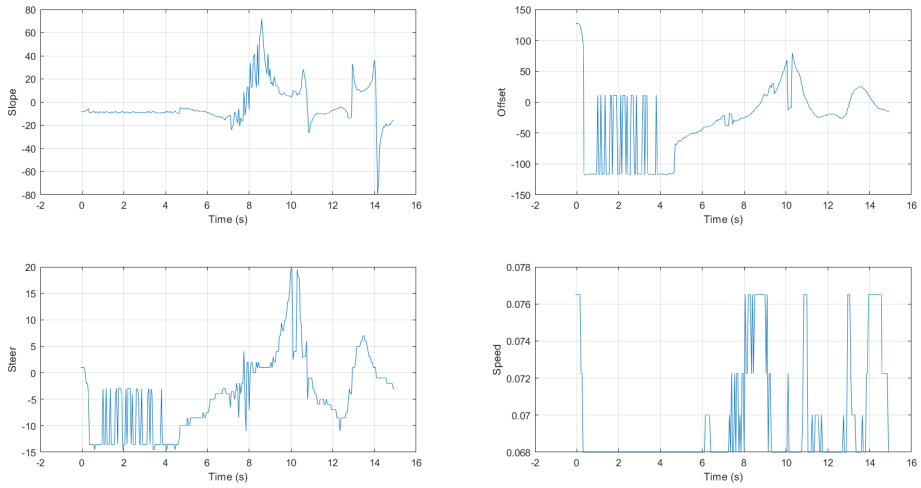


Figure 4.5: Test 2 for lane narrowing

test, are needed to correctly follow the reference line trajectory.

4.3 Lane change

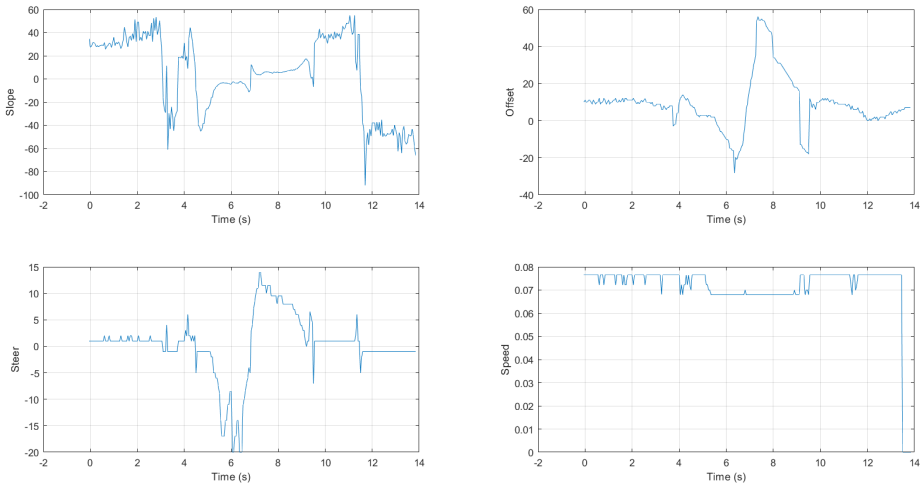


Figure 4.6: Test 1 for lane change

The scenario consists of a one-way highway road represented by two parallel white lanes, each 35 cm wide. A diversion is simulated using yellow markings that initially

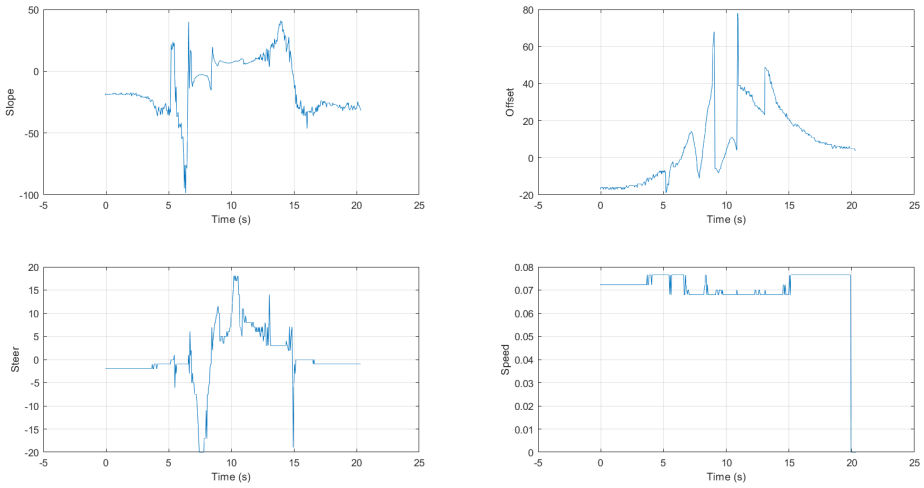


Figure 4.7: Test 2 for lane change

overlap the white lines of one lane, continue through a leftward curve that transitions the trajectory onto the second lane, and relinquish precedence back to the white markings (shown in Figure 4.3b). The lane lines of the curvature maintain a distance of 35 cm.
Figure 4.6 illustrates the results of a test in which the vehicle completed the diversion trajectory in approximately 14 s. The steering angle chart (bottom left) reveals an initial left turn followed by a right turn of comparable magnitude, reflecting the geometry of the diversion path. Figure 4.7 presents the outcome of a slower test run along the same diversion trajectory, during which the car traversed over the yellow lane markings, leading to a higher rolling resistance due to the altered surface characteristics of the yellow markings. In lane changing tests, offset variations are more significant compared to the straight road scenario, due to the tight curvature that challenges the stability of the lane keeping algorithm.

4.3.1 Traffic Red-Yellow Barrier for lane closure

The integration of the TRYB as an indicator of a sharp curve has been proposed to facilitate lane change (as in Figures 3.6 and 4.3b). Figures 4.8 describes the typical lane changing behavior: a left turn starting at $t = 5.0\,\text{s}$ followed by a right turn, around $t = 8.0\,\text{s}$. δ_f is included in the interval $[-20, +20]$ for all time instants and the offset is maintained in $[-50, +50]$. The peaks observed at $t = 9.7\,\text{s}$ in the slope (999), offset (-320), and speed (0) profiles indicate a momentary loss of lane detection, presumably due to insufficient illumination. The subsequent rapid recovery ensured that no visible effects have been observed during the test. The second proposed experiment, with results shown in Figure 4.9, exhibits behavior

similar to that observed in the first experiment: the time required to complete the diversion trajectory is approximately 13 s; in this case the initial left turn, followed by a right turn that concludes with a straight segment, is more pronounced; the lateral offset range is wider, spanning in $[-150,50]$. The integration of TRYB visually enhances the driving performance, achieving the desired result.

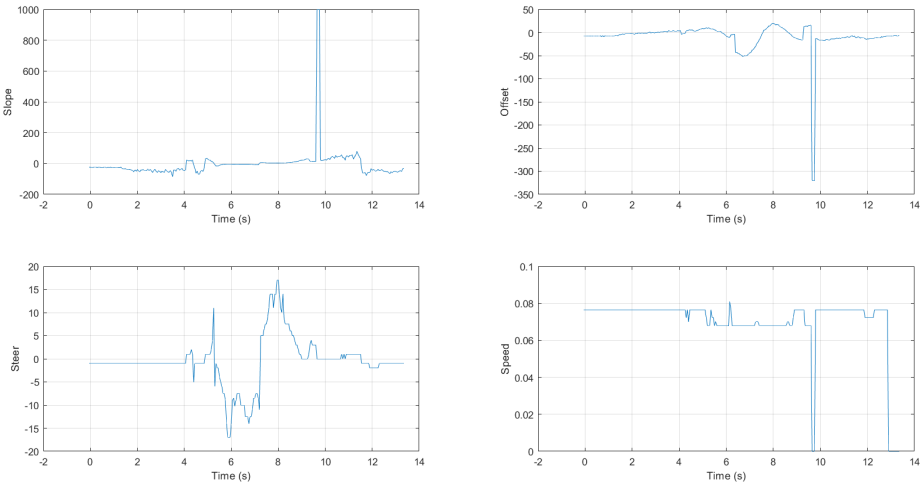


Figure 4.8: Test 1 for lane change with TRYB

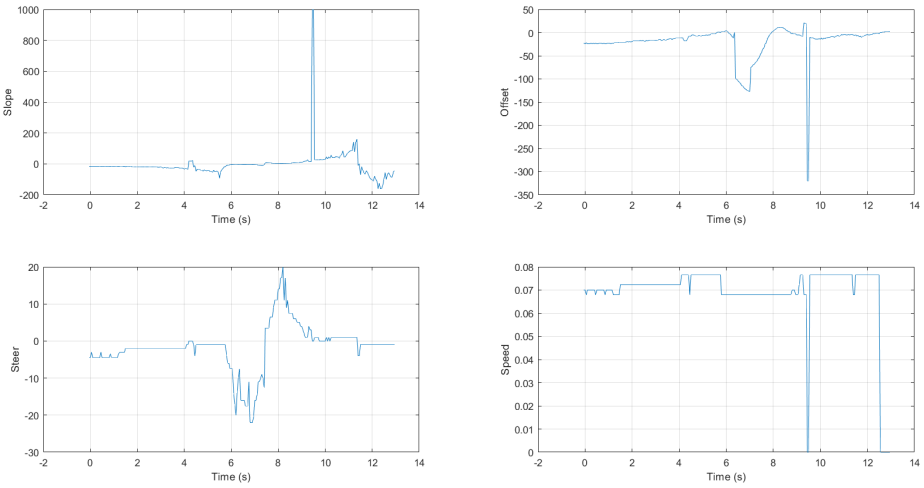
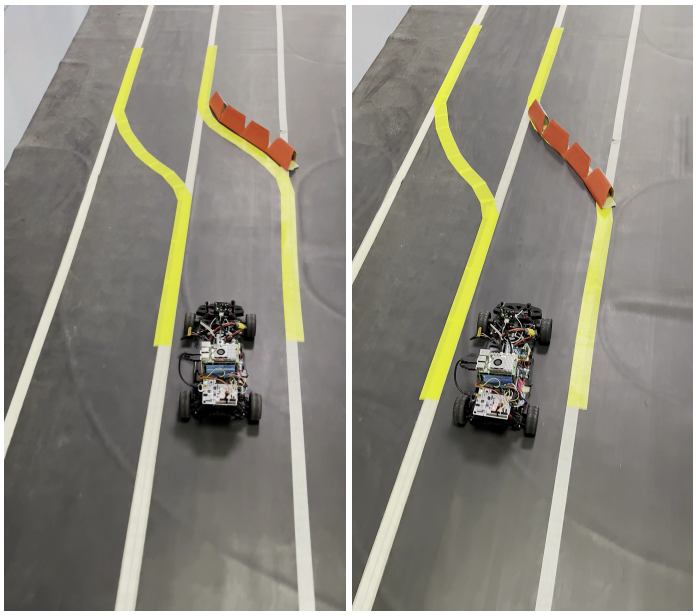


Figure 4.9: Test 2 for lane change with TRYB

4.3.2 Red barrier for lane closure

To indicate the correct diversion path and more clearly signal that the lane is closed, red barriers were also used as indicators in two different scenarios: positioned behind the yellow lane markings (1) as shown in Figure 4.10a, and placed directly over the markings (2), blocking the straight path and hiding the roadwork markings (Figure 4.10b).



(a) Case 1 (b) Case 2

Figure 4.10: Lane closure due to roadworks

1. For the first scenario, the initial experiment (shown in Figure 4.11) demonstrates appropriate lane-changing behavior, despite the steering command exceeding the maximum allowable value around $t = 6\,\text{s}$. This results in a significant offset variation, which remains within acceptable limits. Further refinements of the algorithm are necessary to limit the extreme values of the steering range. Nevertheless, this experiment is visually the most successful one. Figure 4.12 instead shows the results of a second test conducted within the first scenario. During this test, the vehicle encountered difficulty moving over the yellow markings, leading to increased rolling resistance. This is evident from the repeated speed and steering commands observed in the time interval $t = [15,20]\text{s}$. Further reducing vehicle speed when approaching roadworks, indicated by the presence of yellow lane markings, should improve performance and help preventing this issue.

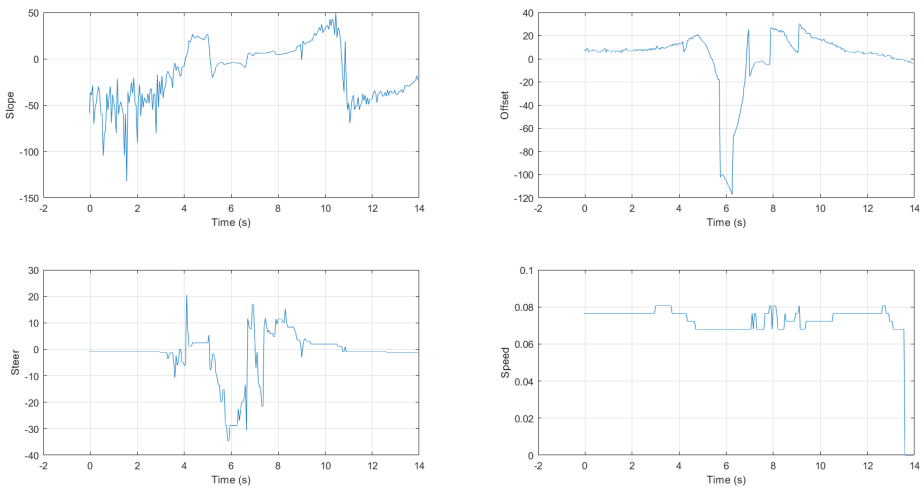


Figure 4.11: Test 1 for lane change with barriers

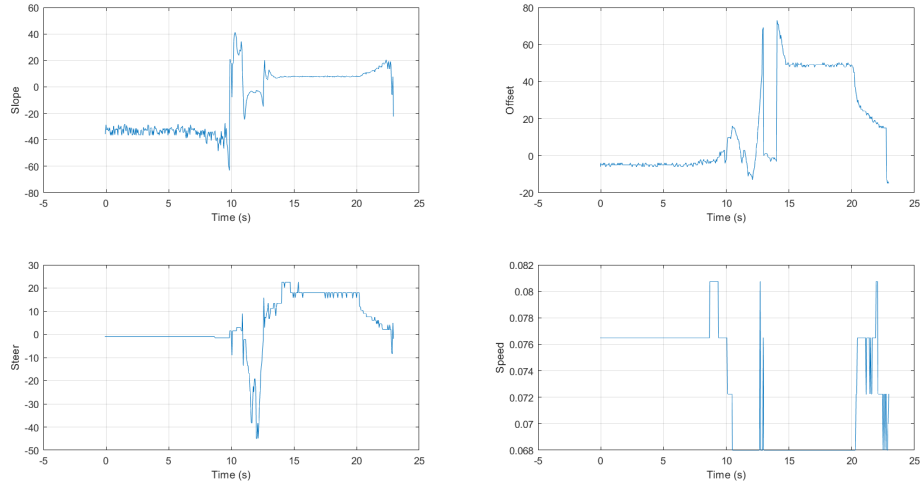


Figure 4.12: Test 2 for lane change with barriers

2. The second scenario presents the challenge of operating without lane detection, as the lane markings are not visible. The results shown in Figure 4.13 indicate that, in the absence of lane recognition, the steering value is computed independently of the reference line slope during the time interval $t = [4,5]$s. As discussed in the context of the first scenario, further refinements to the algorithm should aim to ensure that the physical limits of the servomotor are not exceeded. The self-driving car correctly followed the trajectory defined by the roadwork constraints. At $t = 7.9\,\mathrm{s}$, lane detection was temporarily lost

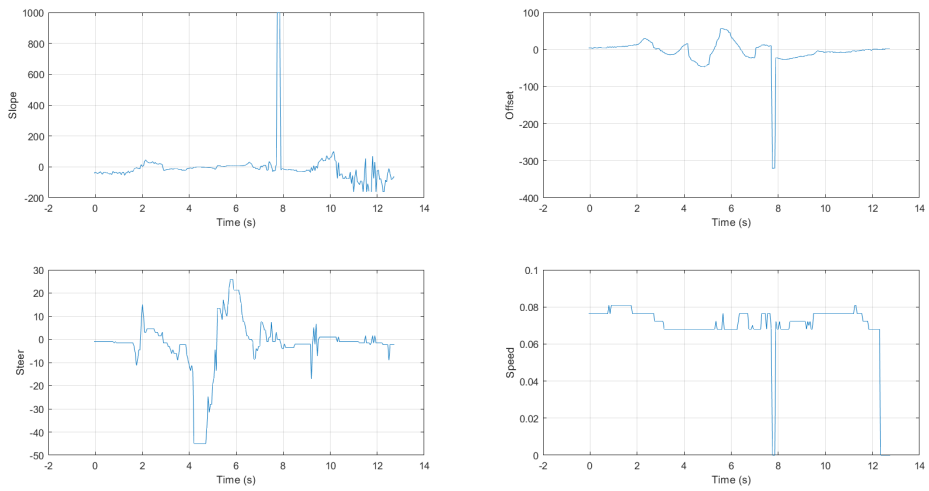


Figure 4.13: Test 3 for lane change with barriers (over markings)

after navigating the curve, an issue also observed in other tests, but it had no visible impact on the vehicle behavior.

4.4 Stop after obstacle detection

Red barriers have been placed across the entire width of the lane, perpendicular to its direction, at a distance of 2 m from the starting point of the track, in order to simulate a complete lane closure without the possibility of changing lanes (positioned on the track of Figure 4.3b as in Figure 3.7b). This setup has been designed to test the vehicle emergency stop behavior. In general, the testing outcomes have not been satisfactory, as the vehicle successfully stops in only 50% of the case studies. The results of the tests in which the vehicle successfully stops at the predefined distance in approximately 4.5 s (Figures 4.14 and 4.15) reveal an initial loss of lane detection, evidenced by a slope value of 999 and an offset of -320, followed by an uncertain steering response, intended to correct the travel trajectory. In some cases, the vehicle stops in time to avoid a collision but does so at a very short distance from the barriers, as shown in Figure 4.16. In this particular experiment, the vehicle required 6.2 s to come to a complete stop, which is longer than expected and compared to the stopping times observed in the successful tests. A failed test is illustrated in the plots of Figure 4.17. In this case, the vehicle required 6.6 s to stop, and this has not been sufficient to avoid impact with the barriers.
Performance improvements could be achieved by decoupling image processing for object detection from lane recognition, executing it concurrently in a separate

thread or on a dedicated parallel processor devoted solely to object detection. Furthermore, alternative communication protocols, such as a dedicated serial interface, could be considered to replace the current USB0–miniUSB cable connection, with the aim of minimizing latency between command processing and actuation.

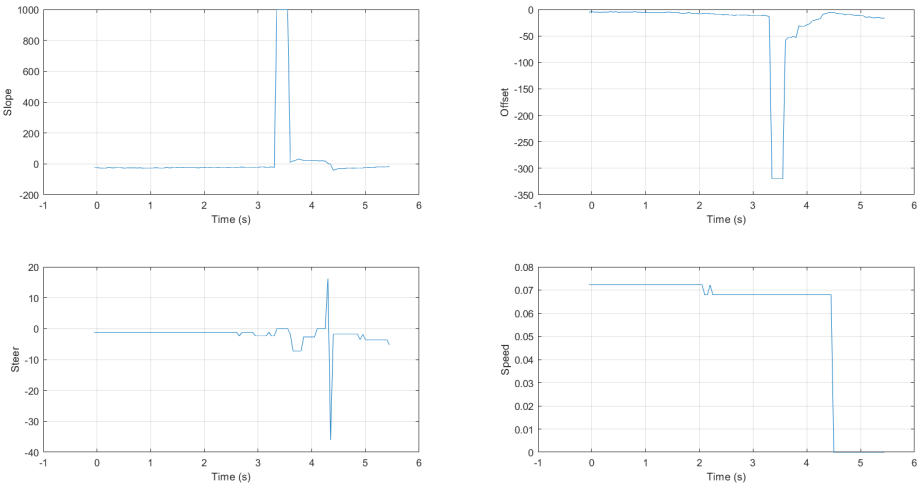


Figure 4.14: Test 1 for emergency stop

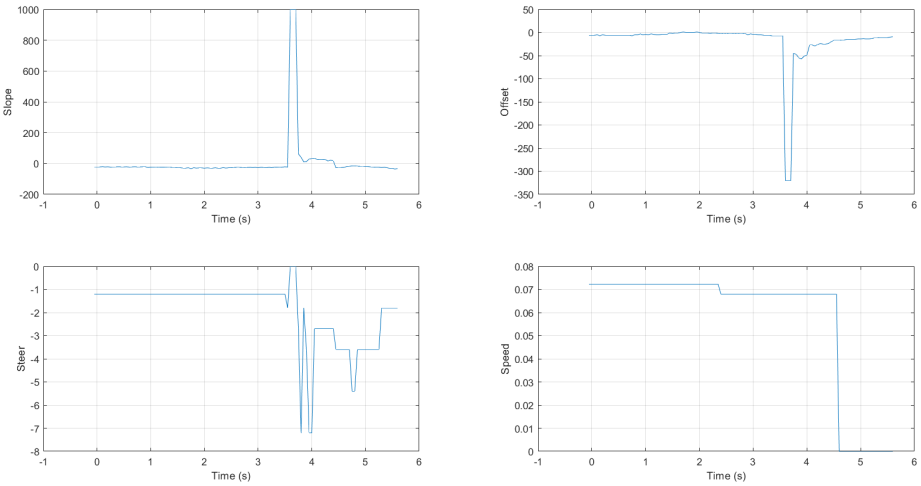


Figure 4.15: Test 2 for emergency stop

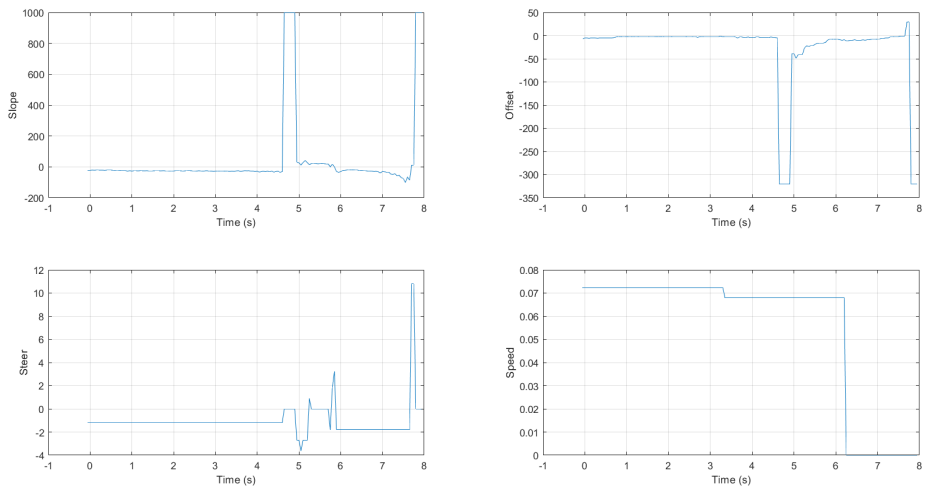


Figure 4.16: Test 3 for emergency stop

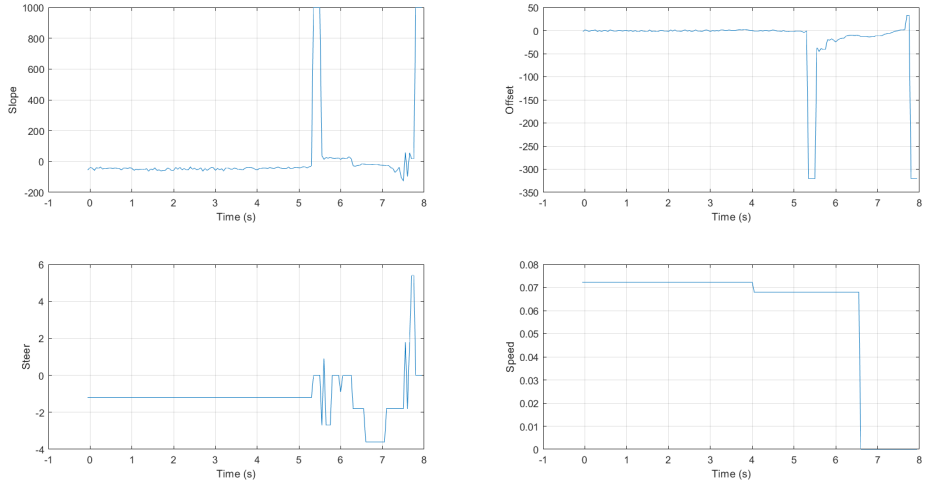


Figure 4.17: Test 4 for emergency stop

4.5 Results from other tests

This section presents additional test results from the same categories previously analyzed, summarized in tabular form. The table reports the test number, the range of variation of the metrics defined at the beginning of the chapter, and the time required to complete the prescribed trajectory. The time metric is the only one for which the measurement unit is indicated, since the other metrics adhere to the considerations already outlined.

Test	m	offset	δ_f	v	t
1	[-132, 92]	[-33, 14]	[-7.5, 4.0]	[0.000, 0.077]	12.95 s
2	[-80, 72]	[-118, 128]	[-14.5, 19.5]	[0.068, 0.077]	14.95 s
3	[-133, 114]	[-160, 38]	[-25.5, 9.0]	[0.000, 0.077]	10.50 s
4	[-160, 160]	[-25, 10]	[-20.0, 13.0]	[0.000, 0.077]	11.45 s
5	[-160, 107]	[-24, 7]	[-5.0, 9.0]	[0.068, 0.077]	10.05 s
6	[-73, 102]	[-28, 29]	[-7.0, 20.0]	[0.068, 0.077]	11.95 s
7	[-160, 111]	[-35, 32]	[-11.0, 8.0]	[0.068, 0.081]	9.95 s
8	[-60, 999]	[-320, 30]	[-8.5, 20.0]	[0.000, 0.077]	11.95 s

Table 4.1: Lane narrowing tests

Test	m	offset	δ_f	v	t
1	[-91, 78]	[-15, 34]	[-20.0, 20.0]	[0.000, 0.077]	18.05 s
2	[-52, 999]	[-320, 73]	[-31.5, 22.5]	[0.000, 0.081]	15.95 s
3	[-160, 999]	[-320, 40]	[-45.0, 31.5]	[0.000, 0.081]	22.95 s
4	[-99, 41]	[-19, 78]	[-20.0, 18.0]	[0.000, 0.077]	20.35 s
5	[-100, 999]	[-320, 82]	[-20.0, 17.0]	[0.000, 0.077]	14.95 s
6	[-92, 55]	[-28, 56]	[-20.0, 14.0]	[0.000, 0.077]	13.85 s
7	[-74, 78]	[-320, 51]	[-20.0, 20.0]	[0.000, 0.090]	16.50 s
8	[-91, 67]	[-131, 25]	[-25.5, 30.1]	[0.000, 0.090]	11.35 s

Table 4.2: Lane change tests

In particular, Table 4.1 presents the results obtained from the lane narrowing tests. On average, the self-driving car required 11.72 s to complete the trajectory defined by the narrowed lane. Steering values remain within the physical limits, in some cases the variation range is more significant, due to less favorable initial car positioning. In the final test the reference line trajectory is temporarily lost, as indicated by the maximum value of the slope m and the minimum offset value, though this does not result in any visible issues. Slope values reach high magnitudes, and the speed remains nearly constant within the same interval, except when the minimum is zero, signifying that the car has halted at a stop line.
Table 4.2 presents the results of lane change tests, based uniquely on the combination of yellow and white lane markings. The average time to complete the lane change is 16.74 s, which is higher than the times recorded for the narrow lane diversion. Loss of the reference line trajectory occurs in a greater number of tests. In some cases, the steering commands exceed the physical limits, requiring algorithmic correction as discussed in Section 4.3.2.
Tables 4.3 and 4.4 report the outcomes of the integration between lane keeping and object detection of traffic barriers indicating a lane closure. The mean time of

15.93 s indicates an improvement compared to the diversion scenario solely marked by yellow lines. However, this value is still higher than the mean time observed in the lane narrowing scenarios, highlighting the greater complexity of lane change maneuvers. Although brief disruptions in the reference trajectory occur more often, the offset values remain within acceptable ranges.

Test	**m**	**offset**	δ_f	**v**	**t**
1	[-50, 120]	[-14, 18]	[-10.0, 2.0]	[0.000, 0.077]	10.35 s
2	[-84, 999]	[-320, 19]	[-17.0, 17.0]	[0.000, 0.081]	13.35 s
3	[-160, 999]	[-320, 21]	[-22.0, 20.0]	[0.000, 0.077]	12.95 s
4	[-160, 999]	[-320, 30]	[-20.0, 20.0]	[0.000, 0.077]	21.95 s
5	[-160, 999]	[-320, 62]	[-36.0, 25.2]	[0.000, 0.081]	18.25 s
6	[-110, 999]	[-320, 80]	[-36.0, 32.4]	[0.000, 0.081]	16.45 s
7	[-160, 999]	[-320, 79]	[-36.0, 20.7]	[0.000, 0.081]	14.95 s
8	[-132, 49]	[-117, 30]	[-34.6, 20.4]	[0.000, 0.081]	13.95 s
9	[-77, 999]	[-320, 73]	[-36.0, 24.0]	[0.000, 0.081]	16.25 s
10	[-46, 999]	[-320, 76]	[-36.0, 36.0]	[0.000, 0.081]	15.95 s

Table 4.3: Lane change tests with traffic barriers (TRYB or red) and visible lane markings

Test	**m**	**offset**	δ_f	**v**	**t**
1	[-160, 999]	[-320, 57]	[-45.0, 25.9]	[0.000, 0.081]	12.75 s
2	[-160, 160]	[-48, 74]	[-45.0, 45.0]	[0.000, 0.081]	20.95 s
3	[-160, 999]	[-320, 38]	[-45.0, 30.4]	[0.000, 0.081]	19.95 s
4	[-60, 39]	[-59, 83]	[-45.0, 90.0]	[0.000, 0.081]	14.95 s

Table 4.4: Lane change tests with red barriers and not visible lane markings

Table 4.5 contain performance data related to emergency stop tests, a particular case of the combination of lane keeping and object detection. This table includes an additional column to indicate the stop outcome for each test instance. The reference line trajectory is lost in every test, as expected, due to the barriers being physically placed over the lane markings, resulting in occlusion and preventing their detection. Successful tests require a stopping time of less than 6 s to halt at the correct distance. However, exceeding this threshold does not necessarily imply test failure. Test outcomes reveal that the vehicle was unable to stop within the required distance in nearly half of the trials, as previously outlined in Section 4.4.

Test	**m**	**offset**	δ_f	**v**	**t**	***stopped?***
1	[-80, 999]	[-320, 10]	[-36.0, 10.8]	[0.000, 0.072]	7.95 s	NO
2	[-126, 999]	[-320, 33]	[-3.6, 5.4]	[0.000, 0.072]	7.95 s	NO
3	[-102, 999]	[-320, 30]	[-3.6, 10.8]	[0.000, 0.072]	7.95 s	YES
4	[-42, 999]	[-320, -4]	[-36.0, 16.2]	[0.000, 0.072]	5.45 s	YES
5	[-85, 999]	[-320, 22]	[-48.6, 12.6]	[0.000, 0.072]	7.95 s	NO
6	[-37, 999]	[-320, 1]	[-7.2, 0.0]	[0.000, 0.072]	5.60 s	YES
7	[-55, 999]	[-320, -3]	[-13.5, 13.5]	[0.000, 0.072]	6.25 s	YES
8	[-160, 999]	[-320, 4]	[-7.2, 4.8]	[0.000, 0.072]	5.95 s	YES
9	[-88, 999]	[-320, 29]	[-20.4, 20.4]	[0.000, 0.072]	7.35 s	YES
10	[-99, 999]	[-320, 4]	[-13.5, 16.2]	[0.000, 0.068]	5.20 s	YES
11	[-160, 999]	[-320, 3]	[-10.1, 10.1]	[0.000, 0.072]	6.60 s	NO
12	[-52, 999]	[-320, 20]	[-13.5, 25.5]	[0.000, 0.072]	5.35 s	YES
13	[-43, 999]	[-320, 28]	[-14.9, 25.7]	[0.000, 0.072]	6.60 s	YES
14	[-52, 999]	[-320, 2]	[-22.5, 2.3]	[0.000, 0.077]	6.35 s	NO
15	[-80, 999]	[-320, 17]	[-42.8, 45.0]	[0.000, 0.068]	7.55 s	NO
16	[-108, 999]	[-320, 9]	[-13.5, 2.3]	[0.000, 0.072]	6.10 s	NO

Table 4.5: Emergency stop tests

Chapter 5

Conclusion and future works

Roadwork conditions present significant challenges for driving automation technologies and require additional considerations. The dynamic and often unpredictable situations created by temporary construction sites must be managed accurately and in a timely manner.
In this thesis, the presence of roadwork-induced diversions necessitated the extension of a lane-keeping algorithm to enable the recognition and differentiation of yellow lane markings from white ones. Color segmentation techniques have been crucial for implementing this enhancement.
In combination with the newly introduced roadwork features, the development of a *maskCalibration* function has improved the robustness of the lane detection, due to the capability of handling varying lighting conditions and of neglecting light reflections on the track. However, since the calibration cannot currently be performed online, a one-time calibration must be executed prior to driving mode. Future improvements could focus on adapting the function for online use, enabling real-time responsiveness to changing environmental conditions.
Color segmentation has also proven effective for object detection, enabling the identification of specific roadwork elements and enhancing the system ability to recognize the beginning of the roadwork zone.
The testing environment for the BFMC self-driving car has been designed in order to support lane narrowing and lane change scenarios. The integration of two types of barriers has been investigated to facilitate and evaluate the lane-changing behavior. An emergency stop scenario has additionally been included to analyze the vehicle response to complete lane obstruction. Experimental results demonstrated that, in the majority of tests, the vehicle correctly performed the diversion in response to both lane narrowing and lane change conditions. However, it has been observed that the vehicle often approaches the diversion at high speed, beginning to decelerate only after completing the curve. Further refinement through additional testing should enhance driving performance. For instance, the vehicle could be programmed to start running at a higher speed and initiate deceleration as soon

as roadworks are detected, specifically when the yellow lane markings begin. Furthermore, implementing this strategy may help prevent the wheels from running over the lane markings, an issue encountered in some tests, thereby improving the vehicle ability to remain entirely within the lane boundaries. Experimental results also revealed that lane changes generally take longer to complete than lane narrowing maneuvers. This finding suggests that, in roadwork zones, adopting lane narrowing strategies may be more efficient than implementing full lane closures, as they are less likely to cause significant delays in travel time.
The presence of traffic barriers indicating the diversion path, and blocking the closed lane, visibly enhances the lane-changing performance, based on observations of the vehicle behavior. Partial success has been obtained for obstacle detection when the red barrier was positioned across the lane, forcing the vehicle to stop. However, significant improvements are still required. Introducing a dedicated parallel processing unit exclusively for object detection could be beneficial.
Sensor fusion involving the camera and a LiDAR sensor could enhance object detection performance. Additionally, integrating hardware components such as headlights may support consistent color calibration by leveraging their predefined luminance conditions, thereby reducing the impact of changing ambient light.
Future research should focus on incorporating a wider range of traffic signs and traffic cones representative of real-world roadwork scenarios, in order to enrich the realism of the testing environment. The development of a communication network facilitating data exchange with smart devices could enhance the perception and interpretation of the road environment.
The Raspberry Pi 5 offers significantly improved performance compared to previous models; however, it remains limited in computational resources when compared to a desktop PC or platforms such as the NVIDIA Jetson, which are commonly used for AI and advanced robotics. These limitations in processing power may render the Raspberry Pi 5 insufficient for executing DNNs or supporting a full ROS deployment, despite the benefits these technologies provide in enhancing image processing and overall system functionality.
Alternative control algorithms supported by neural networks will be explored in future work and evaluated against the current vehicle performance.
Research focused on adapting autonomous driving technologies to the complexities of roadwork environments demonstrates significant potential for enabling robust and reliable control systems applicable to real-world conditions.
To enable the integration of fully autonomous driving technologies into real-world environments, a standardized framework for managing roadwork conditions should be properly defined. Targeted changes to road signage and infrastructure in roadwork zones may include the use of reflective materials for pavement markings in bright and highly distinguishable colors, such as red, orange, or fluorescent green, that are clearly differentiated from standard white lines. Additionally, traffic elements with uniform and high-visibility coloring can improve detectability for both

human drivers and autonomous vehicle sensors. This would aim to reduce the high degree of variability that makes autonomous driving particularly challenging. Moreover, traffic elements should support the integration of smart devices capable of communicating their position and characteristics to approaching vehicles.
The establishment of international standards, along with the optimization of traffic signals and road markings for machine perception, could significantly enhance the reliability and safety of self-driving vehicles in real-world environments, and support further advancements for autonomous driving research.

Acronyms

ODD Operational Design Domain

SAE Society of Automotive Engineers

DDT Dynamic Driving Task

LiDAR Light Detection and Ranging

GPS Global Positioning System

IoT Internet of Things

CNN Convolutional Neural Network

DNN Deep Neural Network

CAV Connected and Autonomous Vehicle

BFMC Bosch Future Mobility Challenge

UPS Uninterruptible Power Supply

RGB Red Green Blue

HSV Hue Saturation Value

TRYB Traffic Red-Yellow Barrier

Bibliography

[1] https://medium.com/neurosapiens/segmentation-and-classification-with-hsv-8f2406c62b39.

[2] On-Road Automated Driving (ORAD) Committee, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Apr. 2021. [Online]. Available: https://doi.org/10.4271/J3016_202104

[3] https://www.goldmansachs.com/insights/articles/partially-autonomous-cars-forecast-to-comprise-10-percent-of-new-vehicle-sales-by-2030, 2024.

[4] A. Plebe, H. Svensson, S. Mahmoud, and M. Da Lio, "Human-inspired autonomous driving: A survey," *Cognitive systems research*, vol. 83, pp. 101 169–, 2024.

[5] https://www.swaayattrobots.com/.

[6] L. Fantauzzo, E. Fani, D. Caldarola, A. Tavera, F. Cermelli, M. Ciccone, and B. Caputo, "Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 504–11 511.

[7] B. Bovcon, J. Muhovic, D. Vranac, D. Mozetic, J. Pers, and M. Kristan, "Mods-a usv-oriented object detection and obstacle segmentation benchmark," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 8, pp. 13 403–13 418, 2022.

[8] R. Atienza, *Advanced deep learning with TensorFlow 2 and Keras : apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more*, 2nd ed. Birmingham, UK: Packt Publishing, 2020.

[9] N. Andrade, T. Ribeiro, J. Coelho, G. Lopes, and A. Ribeiro, "Combining yolo and deep reinforcement learning for autonomous driving in public roadworks scenarios," 01 2022, pp. 793–800.

[10] M. K. Singh, N. Formosa, C. K. Man, C. Morton, C. B. Masera, and M. Quddus, "Assessing temporary traffic management measures on a motorway: Lane closures vs narrow lanes for connected and autonomous vehicles in roadworks," *IET intelligent transport systems*, vol. 18, no. 7, pp. 1210–1226, 2024.

[11] M. Hussain, S. Glaser, G. Larue, S. Dehkordi, and M. Masoud, "A cooperative lane-change behaviour evaluation for connected and autonomous vehicles in road work zones environments," *IEEE Transactions on Intelligent Vehicles*,

vol. PP, pp. 1–25, 01 2024.

[12] M. Maurer, J. C. Gerdes, B. Lenz, H. Winner, J. C. Gerdes, B. Lenz, H. Winner, and M. Maurer, *Autonomous Driving: Technical, Legal and Social Aspects*. Cham: Springer Nature, 2016.

[13] https://www.moralmachine.net/.

[14] A. Shariff, J.-F. Bonnefon, and I. Rahwan, "Psychological roadblocks to the adoption of self-driving vehicles," *Nature Human Behaviour*, vol. 1, 09 2017.

[15] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[16] I. Nastjuk, B. Herrenkind, M. Marrone, A. B. Brendel, and L. M. Kolbe, "What drives the acceptance of autonomous driving? an investigation of acceptance factors from an end-user's perspective," *Technological forecasting & social change*, vol. 161, pp. 120 319–, 2020.

[17] https://www.connectedautomateddriving.eu/.

[18] J. Reckenzaun and S. Solmaz, "Infrastructure for the extension of odds - applied in connected and automated driving and standardization procedures," in *2024 IEEE International Automated Vehicle Validation Conference (IAVVC)*. IEEE, 2024, pp. 1–6.

[19] P. Singh and M. A. Islam, "Movement of autonomous vehicles in work zone using new pavement marking: A new approach," *Journal of Transportation Technologies*, vol. 10, pp. 183–197, 01 2020.

[20] B. Mathibela, I. Posner, and P. Newman, "A roadwork scene signature based on the opponent colour model," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4394–4400.

[21] Z. Xu, Z. Song, Y. Dong, and P. Chen, "Impact of level 2/3 automated driving technology on road work zone safety," 2025.

[22] V. Saillaja, M. Rehaman Pasha, S. Krishnaveni, B. Ravinder, S. Srinivasan, and V. Nandagopal, "Iot-embedded traffic cones with cnn-based object detection to roadwork safety," in *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE, 2024, pp. 120–125.

[23] N. Formosa, M. Quddus, M. K. Singh, C. K. Man, C. Morton, and C. B. Masera, "An experiment and simulation study on developing algorithms for cavs to navigate through roadworks," *IEEE transactions on intelligent transportation systems*, vol. 25, no. 1, pp. 1–13, 2024.

[24] C. Gentile, "Lane keeping algorithms and intersection management on a scaled mock-up autonomous vehicle," Malan, Stefano Alberto, 2024-10-31.

[25] L. Di Biase, "Lane keeping algorithms and intersection management on a scaled mock-up autonomous vehicle," Malan, Stefano Alberto, 2024-10-31.

[26] https://boschfuturemobility.com/.

[27] J. T. Warner, *The Handbook of Lithium-Ion Battery Pack Design: Chemistry, Components, Types, and Terminology.*, 2nd ed. San Diego: Elsevier Science

& Technology, 2024.

[28] J. T. Warner, *Lithium-ion battery chemistries : a primer*. Amsterdam, Netherlands: Elsevier, 2019.

[29] M. Hannan, M. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable & sustainable energy reviews*, vol. 78, pp. 834–854, 2017.

[30] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1094–1099.

[31] K. Siang Tan and N. A. Mat Isa, "Color image segmentation using histogram thresholding – fuzzy c-means hybrid approach," *Pattern recognition*, vol. 44, no. 1, pp. 1–15, 2011.

[32] https://opencv.org/.