POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Characterization of drivers' visual perception through driving simulator tests

Tesi di Laurea Magistrale in
Mechanical Engineering - Ingegneria Meccanica

Author: **Alessandro Testori**

Student ID: 10538311
Advisor: Prof. Stefano Melzi
Co-advisors: Ing. Antonio Cioffi
Academic Year: 2021-2022

Abstract

Autonomous vehicles are becoming much diffused in our life and at the same time some issues rise with it, since we are empowering a computer to take decisions for us. For this reason recently it's increasing the interest towards the development of models of driver "human-like", which is a key element for its success. It means to create a software of control that decides what to do similarly to what a real driver would do. There are two reasons for this: first of all, drivers may accept more easily an autonomous driver if they know that it will have a driving style similar to theirs. Secondly, a model of driver "human-like" helps in the development of the control strategy of autonomous vehicles. In fact, in the first phase, autonomous vehicles will interact with traditional vehicles and if we have a model of the driving style of real drivers, we can create virtual environments in which autonomous vehicles interact with vehicles that behave in a partially unpredictable way, that doesn't completely respect the traffic laws.

This thesis starts from these considerations, and it was possible to realize it thanks to the possibilities offered by the driving simulator of Politecnico di Milano to study the behaviour of drivers. In this thesis are reported all the steps taken to characterize the visual perception of drivers, which is the first step towards the complete description of normal drivers' style. In this thesis are explained the creation of virtual scenarios, the tests, the data collected, the calculated parameters and the creation of the model of perception.

Keywords: driving simulator, VI-Worldsim, RoadRunner, eye-tracker, model of perception

Abstract in lingua italiana

I veicoli a guida autonoma si stanno diffondendo sempre di più e allo stesso tempo crescono alcune preoccupazioni riguardo a questo, dato che si dà la possibilità ad un computer di prendere decisioni alla guida al posto delle persone. Per questa ragione di recente sta crescendo l'interesse per creare modelli di guidatore "umano", che è un punto fondamentale per il successo dei veicoli autonomi. Questo significa creare un programma di controllo che decida che azione compiere in modo simile a come farebbe un conducente reale. Ci sono due ragioni per perseguire quest'obiettivo: in primo luogo, i guidatori possono accettare con maggiore facilità un sistema di guida autonoma se sanno che quest'ultimo si comporterà in modo simile a loro. In secondo luogo, un modello di guida "umano" aiuta a sviluppare la strategia di controllo dei veicoli autonomi. Infatti, almeno nella prima fase, i veicoli autonomi dovranno interagire con veicoli tradizionali e se si dispone di un modello che simula il pilota reale, si possono creare ambienti virtuali in cui i veicoli autonomi interagiscono con veicoli che si comportano in maniera parzialmente imprevedibile e che non rispettano perfettamente le regole stradali.

Questa tesi ha come punto di partenza queste considerazioni, ed è stato possibile realizzarla grazie alle possibilità offerte dal simulatore di guida del Politecnico di Milano di studiare il comportamento dei conducenti di veicoli. In questa tesi sono riportati tutti i passaggi necessari per caratterizzare la percezione visiva dei guidatori, ovvero il primo passo verso la creazione di un modello completo di un conducente reale. In questa tesi sarà spiegata la creazione degli scenari virtuali, i test, i dati raccolti, i parametri calcolati e la creazione del modello percettivo.

Parole chiave: simulatore di guida, VI-Worldsim, RoadRunner, eye-tracker, modello di percezione

Contents

Introduction

Autonomous vehicles could drastically improve the safety on the road, but they might be useless if people doesn't accept them. When an ADAS takes control on the vehicle, it works autonomously from the driver, and this could be a trust issue, since drivers are used to have complete control on the car. For this reason it is essential to develop a model of driver "human-like" for 2 reasons:

1. Drivers may accept more easily an autonomous driver if it is expected to behave similarly to what they would also do during driving.
2. A model "human-like" could be useful to create environments of simulation in which autonomous vehicles interact with vehicles that behave in a similar way to real vehicles present on the road, driven by real drivers.

Since it's very useful to develop this model of driver, it's of fundamental importance to characterise the driving style of drivers with numbers and parameters, so that an algorithm has the basics to reproduce it. To do so, first it's essential to develop a model of perception of drivers. This is the starting point for the development of this thesis. To obtain this result, it is essential to employ a driving simulator.

A driving simulator is a particular device which allows to put the human element inside a virtual ambient of simulation. Humans are unpredictable and are hardly to reproduce the exact same behaviour many times, so this why the simulator is so helpful. Since it creates a virtual ambient, it is capable to give every time the exact same conditions. This is important because in science repeatability is fundamental to give validation to the experimentation phase. After experimentation some data may be extracted, and it is desirable to be influenced by just the human factor, because generally it is required to study just the influence of one factor at a time. If the experimentation phase is conducted in real life, more problems come, since more effort must be applied to guarantee the same external conditions and the same parameters of the equipment that will interact with people.

With the simulator installed at Politecnico di Milano (and generally also for all professional driving simulators) it is possible to see exactly how a driver interacts with the vehicle, onboard controls and the other characters on the road such as traffic cars, pedestrians and so on. This is possible thanks to the instrumentation mounted in the simulator; for example there are:

- sensors on the pedals to keep trace of the braking history, the intensity and the duration
- sensors on the steering wheel to keep record of the steering history and the torque applied by the driver
- electronic eyeglasses called 'eye tracker' which give possibility to know where and what the driver is looking while driving
- other sensors to measure the skin conductivity

By putting all these information together, it is possible to study deeply what the driver gives to a vehicle as an input and perform an objective evaluation of its performances through numbers and charts.

A driving simulator, thanks to its characteristics, it's particularly suitable to study autonomous vehicles (ADAS), because in first analysis they can be tested in a controlled environment and in complete safety. Autonomous vehicles can provide more safety compared to normal cars, in particular the active protection, since they work to prevent accidents. They gain information about the environment around the car with cameras, infrared, lidar, lane sensors and so on. All these sensors give information to the ECU, that can take decisions and intervene. For this reason, a driving simulator is quite useful in this sense, since the control strategy at the core of the ADAS (Advanced Driver Assistance Systems) can be applied in a virtual environment, to study how it will perform, what decisions will take etc. The driving simulator offers repeatability to each test, and so consistency in the initial conditions and external factors, which is fundamental in experimental research.

The first part of this thesis is dedicated to describe how the virtual scenarios of the simulator are created and what are the reasons behind each choice. Then it is presented the active part of the thesis which are the tests: how they are conducted and what are the data collected. Following there is the data elaboration, the parameters required to describe the field of view of a driver and final interpretation of the results.

In the second part is presented another test conducted to get the same target. The focus this time is on the capacity to estimate the speed of moving cars and determine where drivers usually look when driving in situations that they may encounter every day. Also this part will present: the scenarios created, why they were composed in a certain way, unfolding of the tests at the simulator, data collected, data elaboration and results.

At the end of the thesis the information collected during the tests are elaborated to create a model of perception of cars that has the target to describe how generally drivers perceive other vehicles in their field of view.

0.1. Summary

1st chapter "Features of DriSMi" gives an overview of the history of driving simulators. Then it is presented the layout of the driving simulator of Politecnico di Milano, softwares used, co-simulation of different softwares, additional sensors available.

2nd chapter "Test at the simulator" presents how the virtual scenarios were created, what software were used and what are their purpose. Then it is explained how to make the traffic cars move as desired. The last part explains what was done in the two tests, virtual scenarios used, procedures and rules given to drivers.

3rd chapter "Data processing" contains a description of the data collected from the tests, how they were analysed and what values were obtained. This procedure is done for the two tests, for each of the virtual scenarios.

4th chapter "Model of perception" explains how the model of perception is created starting from the data obtained in the two tests. Finally the model of perception is applied in a virtual scenario and the results are reported.

1 | Features of DriSMi

DriSMi (Driving Simulator of Politecnico di Milano) is an innovative cable-driven dynamic simulator. Nowadays, driving simulators are getting popular as tools for designing road vehicles and related control systems [1] [2] [3]. Starting from rather simple devices developed in last decades of the past century, engineers have been developing new driving simulators exploiting the progresses in computer graphics and control systems [4]. New architectures have been proposed trying to improve the realism of driving experience while dealing with constraints related with costs and available workspace [5].

1.1. Driving simulators history

Companies started building motion simulators in the first part of the 20th century, with flying simulators. One of the first ones was the French company called 'Antoinette', operative between 1903 and 1912. They were pioneers who built lightweight engines and aeroplanes before WWI. They also had a flight school, which led to the creation of one of the first flight simulator. It started operating around 1909, and prepared pilots to control their plane 'Antoinette VII'. It was a rudimental simulator, composed of a half barrel mounted over a universal joint, with flight controls, pulleys, and stub-wings (poles) to allow the pilot to maintain balance while instructors applied external forces.

Figure 1.1: Antoinette flight simulator [6]

In 1949 a major step forward was taken thanks to the engineer V. Eric Gough, who invented the hexapod to test tires [7]. It is a kinematic structure composed of 2 platforms and 6 cylinders. The base platform is fixed, while the 6 cylinders and the upper platform are free to move. All cylinders are independent and give the possibility to the upper frame to move independently. In this way it can perform 3 motions and 3 rotations in space. This structure will be used almost in all future motion simulators. For example in 1965 D. Stewart used a variant of this structure to create a flight simulator.

Figure 1.2: Hexapod [8]

In 1985 Mazda built its driving simulator to test future driver-vehicle systems and simulate vehicle motions in emergencies or under critical situations. It consists of a mini super computer, a moving base with a large amplitude motion system and a high speed visual system and a movable cabin.

Another big step forward was taken in 2002 when National Advanced Driving Simulator presented NADS-1, figure 1.3. It is a base platform that can move on a bay of about 19 m x 19 m. On top of it there's an hexapod and a dome, where the drivers enter. The dome is a sort of closed hemisphere that can rotate around the Z axis of 330° thanks to a turntable. Inside real size vehicles can be placed with actuators reproducing vibrations coming from the road and a 360° screen to give complete immersivity.

Figure 1.3: NADS-1 [9]

In 2007 Toyota exceeded the NADS-1 and created an even bigger driving simulator, figure 1.4. It has a very similar construction philosophy, except for the fact that the full size car rotates inside the dome, while the NADS-1 had all the dome rotate together with the car. This incredible simulator is the biggest in the world, with a working space in X, Y direction of 35 m x 20 m and to produce a longitudinal/lateral acceleration up tp 0,5 G and a speed of 6,1 m/s. The dome rests on a hexapod with a max leaning angle of 25°. It shares with the NADS-1 the max yaw angle of 330° and 4 shakers mounted on each wheel of the car to reproduce road irregularities.

Figure 1.4: Toyota driving simulator [10]

There are also simpler architectures of driving simulators, in which 3 actuators move the cockpit inside the platform. An example is the DiM 150 built by VI-grade, figure 1.5.

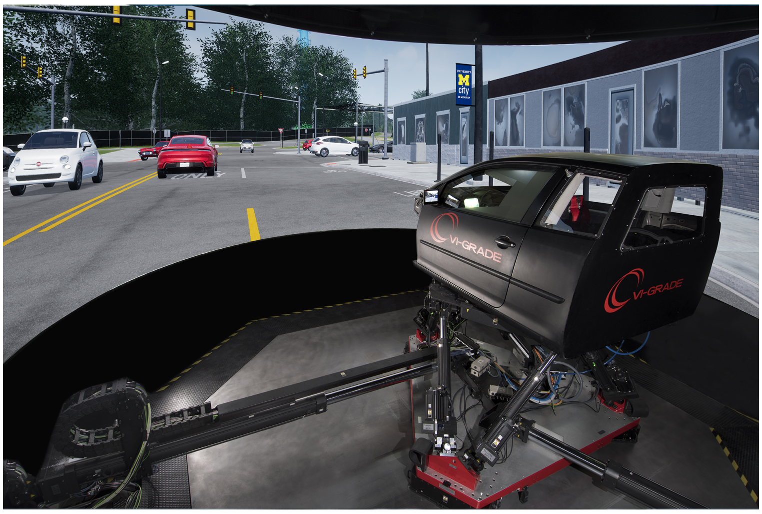


Figure 1.5: VI-grade DiM 150 [11]

1.2. Driving simulator of Politecnico di Milano

Some information about the driving simulator are reported to define the working equipment used for the development of the thesis. The new driving simulator “DiM 400” was installed in the DRISMI Lab in March 3rd 2021 by the company VI-Grade, figure 1.6.

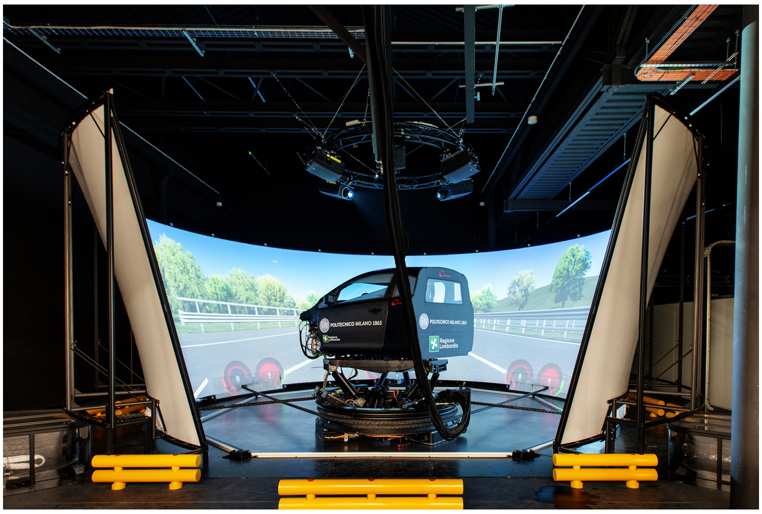


Figure 1.6: VI-grade DiM 400 [12]

The Dim 400 differs from the simpler versions DiM 150 (Figure 6) and DiM 250, because it adopts a cable driven system for the motion of the lower stage to enable larger displacements and for a longer time, instead of the 3 electric actuators configuration of the 2 smaller simulators. The lower stage (diskframe) is mounted on 3 airpads that create a cushion of air between the baseframe and diskframe. The lower stage is moved by 4 cables pulled by 4 rotary motors positioned at each corner of the baseframe. The electric motors (TK850.200.50.A model manufactured by Phase Motion Control) are triphase synchronous servo motor with a rated torque of 8147 Nm, maximum speed of 150 rpm and a power output of 113 kW. The diskframe is free to move on a 6 x 6 m octagonal platform and can rotate around the vertical axis, to reproduce yaw dynamics, for a maximum of 58° with an estimated bandwidth of about 3 Hz. The combined work of the 4 motors generates the x and y axis motion to replicate the chassis and road dynamics. This configuration gives some benefits compared to the simpler versions:

- Bigger displacement of the diskframe, which means higher accelerations: up to 1,5G in longitudinal and lateral direction and up to 2.5G in vertical direction
- Bigger heave motion which gives better perceptions of the vertical dynamics
- Some maneuvers like double lane change can be reproduced on a 1:1 scale
- The bandwidth of the cable drive is 3 Hz

Inside the lower part, there is a system called "ICS" (Inertial Compensation System), used to compensate the high frequency inertia in x,y and yaw motion. This system compensates the imbalance of the forces of the Hexalift: since it is mounted on top, the reaction forces are transmitted to the base; to ensure that these forces do not disturb the work of the cables this system generates opposite inertia forces to counterbalance them. This system generates inertia forces by moving 3 masses. The system includes 3 actuators, and a single actuator is composed by:

- A mass
- A linear module
- A reducer
- A brushless motor

On top of the lower stage, we can find the upper stage, a 6 DOF structure called “Hexapod”. This structure has a bandwidth of 30 Hz and is composed of:

- An upper platform
- 6 identical joints, each one made of: 2 cardanic joints, linear link with passive inertial joint and implemented linear joint
- Inclined linear guides

The 6 linkages are made of 2 rails on which rotates a worm gear, which is moved by an electric motor, as shown in figure 1.7.

Figure 1.7: Hexapod under the cockpit of the simulator[12]

The 6 linkages generate the x,y and z axis motion, pitch, roll and yaw rotation of the upper frame (the cockpit)to replicate the high frequency body motion.

On top of the hexapod we find the final and most important structure, the cockpit, figure 1.8. It is equipped with 8 shakers, with a bandwidth of around 200 Hz, to reproduce vibration coming from the engine or road irregularity. It is also equipped with more features to increase realism and immersivity:

- Active belts/active seats that mimic the effect of extended longitudinal/lateral accelerations
- Active steering system to give a realistic feedback to the driver and simulate active steering control systems
- Active braking system to reproduce the proper pedal feeling and mimic the effect of active braking systems like the ABS
- 5 speakers to reproduce the noise from the inside and outside of the vehicle
- Virtual side and central mirrors to have a better understanding of the surrounding area
- Virtual dashboard to give basic information to the driver

Figure 1.8: Cockpit of the simulator[13]

The visualization system is composed of 5 projectors and a conical screen that covers 270° of the field of view of the driver.

The main applications of this simulator are:

- Vehicle dynamics
- Ride and comfort
- ADAS and AV
- Motorsport sector like race set-up and driver's training

1.3. DriSMi laboratory

DriSMi is the laboratory at Politecnico di Milano that hosts the simulator environment and all its hardware (figure 1.9).

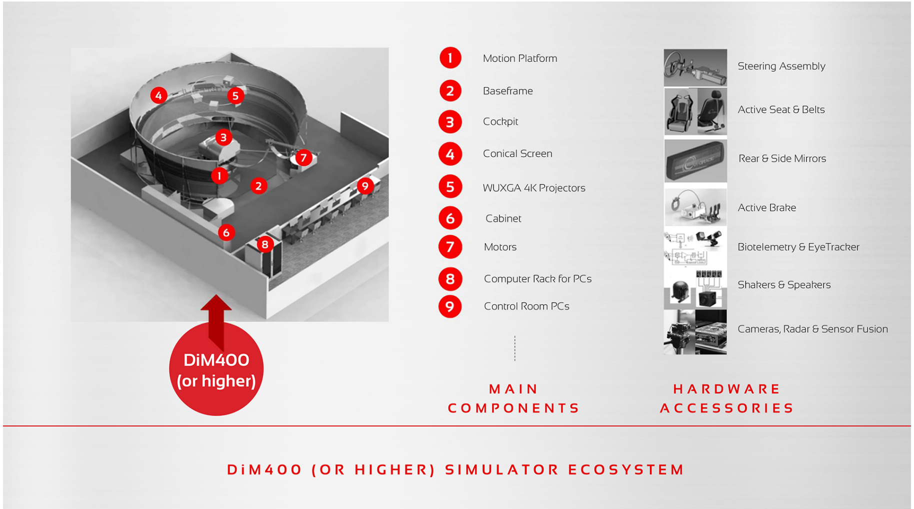


Figure 1.9: Layout of the simulator[14]

Number 3 is the cockpit of the simulator. Around it, there is a 270° screen illuminated by 5 projectors hanging from the ceiling (number 4,5). The platform (number 2) is the space in which the cockpit moves, with a distance between 2 parallel segments of 6 m. Four motors (number 7) move 4 cables to control the motion of the cockpit. Beside the platform, there is the cabinet (number 6): it's the motion control power subsystem. Everything is controlled from the control room (number 9) where technicians keep under control the simulator and start all the tests. Beside it, there is the room where are stored rack pcs that perform all the calculations. The simulator runs on the Concurrent Real-Time Machine, a high performance, Linux based computer platform for time-critical simulations data acquisition and process control applications named iHawk [15]. Thanks to this hardware, it is possible to guarantee a latency between the input of the driver and the response of the simulator of just 20 ms [16]. The main specifications of the Concurrent Real-Time Machine are:

- Operative System: RedHawk Linux 7.3 , 64 bit
- CPU: Intel® Xeon(R) Gold 6144 @ 3.50 Ghz x 16
- RAM: 45.7 GB
- Graphics: Quadro P400/PCIe/SSE2

1.4. Softwares used by the simulator

The simulator utilizes mainly 2 softwares: VI-CarRealTime [17] and VI-Worldsim [18], both released by VI-Grade, the same company that produces the simulator. VI-CarRealTime is a real-time vehicle simulation environment that adopts a single simplified vehicle model. It can perform simulations on vehicles and get information on all the subsystems present in it. It is also able to perform co-simulations in real time with other software. In fact, it works together with VI-Worldsim at the simulator. This software is a fully integrated graphic environment that can work on driving simulators. It is based on UNREAL graphic engine. It gives possibility to incorporate on the simulator various environments, traffic simulation, pedestrians, animals, lightning, weather and more. Cars, pedestrians and animals are AI-driven actors who follow the normal routes or can be controlled by defining specific behaviors, giving the possibility to fully customize the simulation [18]. Another important software required to exploit at its best the simulator, is Road Runner [19]. This software is released by MathWorks and gives possibility to create virtual scenarios for simulations. This tool is essential since it gives possibility to create every kind of road scenario and satisfy whatever requirement is needed to conduct a test. It is focused on the architecture of the scenario and also on its appearance. This programme is mainly focused on the creation of roads for cars. In fact, there are a lot of commands dedicated to the creation of a road: it is possible for example to impose the shape of a road, it's width, number of lanes, markings on the ground, vertical signs and so on. It is also possible to add external objects to increase the resemblance to the real world like guard rails, trees, lawn, traffic light etc. Roads created with this programme aren't just aesthetic tools, in fact they are paths that communicate information with the cars and for this reason cars that are driven by the PC, stay inside the road, respect the speed limit and stop at a red light.

As a demonstration of the big potentiality of this programme, here are reported some screenshots of a scenario created with it. This scenario reproduces an actual intersection present in Milan (Italy) between the highway A4 and A50 near Settimo Milanese (figure 1.10 and figure 1.11).

Figure 1.10: Highway intersection (1)

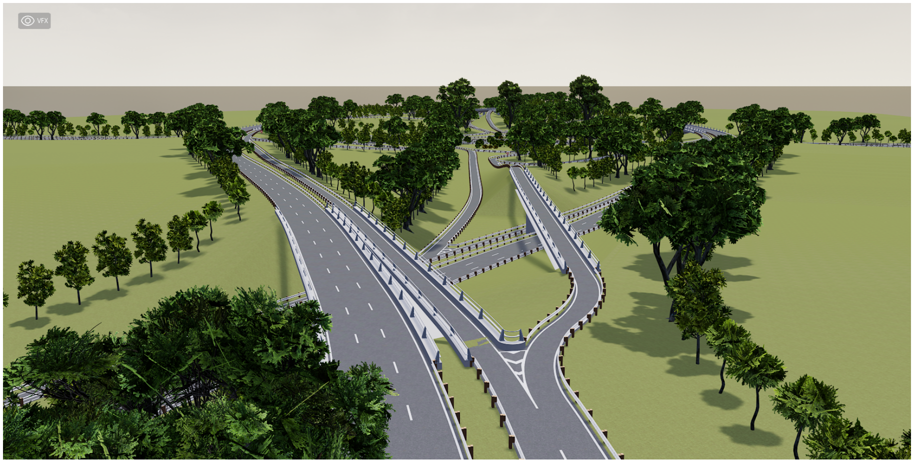


Figure 1.11: Highway intersection (2)

1.5. Software environment

Once the scenario is satisfying requirements, it can be exported. The file has extension 'rrscene' . Export means to create a file, starting from the scenario file, which can be used by other softwares like Worldsim and CarRealTime.

The first step is to export the file directly from RoadRunner environment: this operation generates 5 files with different extensions for each scenario that are 'fbm','3D object','geojson','rrdata' and 'xodr'. Then these files are opened with 'Worldsim Modkit' that uses 'Unreal Engine', which is a 3D creation tool for photoreal and immersive experiences. Modkit is a programme that can perform exports of scenarios, in a format exploitable by VI-grade programmes. This software performs an elaboration of the file with extension and in about half an hour (for the complexity of the intersection scenario) it creates the RRM zipped folder that contains different files. The final step consists in extracting the zip folder and put the different files in specific Worldsim folders. This is required because in this way Worldsim can find the location of the file of a specific scenario and open it. Among the different files, the one that contains the info of a scenario has extension 'json'. When Worldsim opens a scenario, it reads these files. Now that the scenario is finally opened, it can communicate with CarRealTime. When a driving session starts at the simulator, CarRealTime performs calculations in real time of the dynamics of the vehicle, starting from the input given by the driver (steering wheel angle, gear shifter, throttle pedal or brake) and sends data to Worldsim. The latter performs the graphic elaborations that allow the driver and the technicians to visualise the resulting motion of the vehicle.

1.6. Additional sensors

An instrumented steering wheel, designed at Politecnico di Milano [20], provides real-time measurement of the forces, moments and grip forces at each hand (figure 1.12).

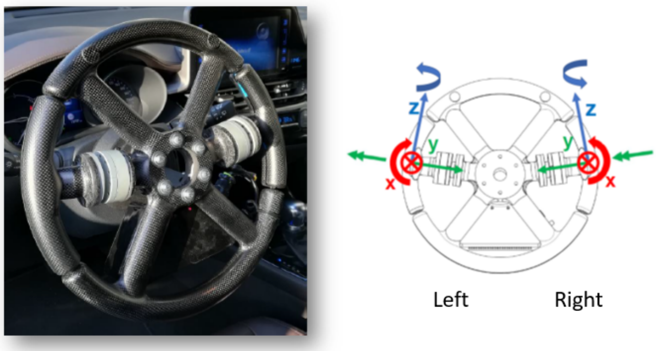


Figure 1.12: Instrumented steering wheel

Another important sensor is the eye tracker 'Pro Glasses 3' provided by the company Tobii [21]. It is a pair of electronic glasses that gives possibility to keep under control where the driver looks and for how long, figure 1.13.

Figure 1.13: Eye tracker

It has at the centre a full HD resolution camera with 106° field of view to capture the scene. Inside the lenses there are 4 cameras to precisely get the motion of the eye and 16 IR illuminators. The glasses are paired with hardware that can transmit acquisitions in real-time via wi-fi and via LAN connection. The laboratory is equipped with software provided by the same company, called 'Pro Lab'. It is a complete solution for behaviour research. It provides a visual user interface and features that support the researcher on all phases of an eye tracker experiment: from test design and recording to analysis.

2 | Test at the simulator

In this chapter are presented the scenarios in which the tests will take place. Participants will drive inside virtual scenarios, so it is important to understand the context in which tests will take place. The second topic of this chapter is related to the motion of the cars that will interact with participants. So, it will be explained the techniques used, programmes and solutions to optimize the management of all the environment of the simulation. The third topic is the explanation of the tests procedure, so there are reported instructions given to drivers, what they see and what they must do.

2.1. First scenario

The first scenario is a proving ground called ‘Piazzale1’ (figure 2.1).This scenario was created to study how the speed of a moving car can influence its perception by the driver. In each sub-scenario, there will be 1 ‘ego’ car and 1 ‘traffic car’. The ego is the car driven by the driver, while the traffic car moves autonomously. To study the influence of the speed on perception, the traffic car will pass in front of the driver following a straight trajectory so that it can be placed at different longitudinal distances. The lateral position of the ‘traffic car’ is calculated so that it would enter the visual field of the driver after 15 s from the start (assuming a visual field of the driver of 45° from the longitudinal direction) with a MATLAB script developed specifically for this test called ‘Calcolo_distanze’. In this way the driver has some time to get used to the scenario.

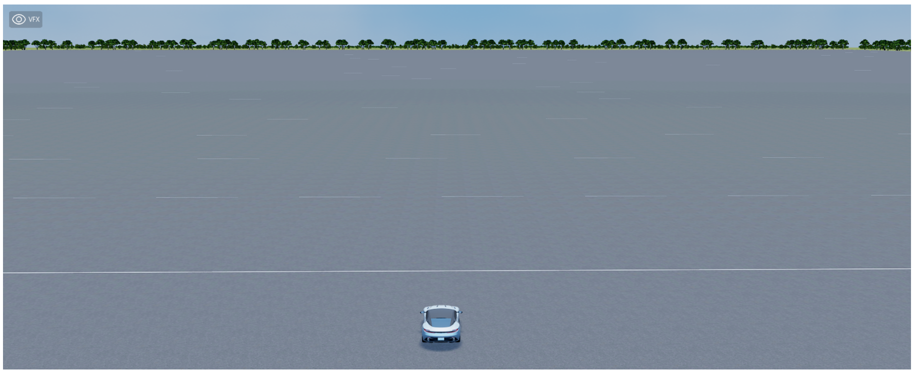


Figure 2.1: This is how the first scenario appears at the start of the test

This scenario has a squared shape with 1km long sides. This is bounded laterally by guard rails and trees to avoid that the driver gets distracted by looking outside of the concrete square. After the creation, cars were added with the software ‘Worldsim’. The car which will be driven in the simulator during the experiment is called ‘ego’ and it’s placed in the scenario, at coordinates x = 37,5 m y = 0 m z = 0 m. It’s important to keep in mind that the origin of each car is placed at the centre of their rear axle, so all distances between cars are measured with respect to this point. Then there is a ‘traffic car’ at 3 different longitudinal distances from the ego car, for 3 different speeds, hence creating 9 different sub-scenarios. The 3 distances are:

- 25 m
- 75 m
- 150 m

While the cruising speeds are:

- 30 km/h
- 70 km/h
- 110 km/h

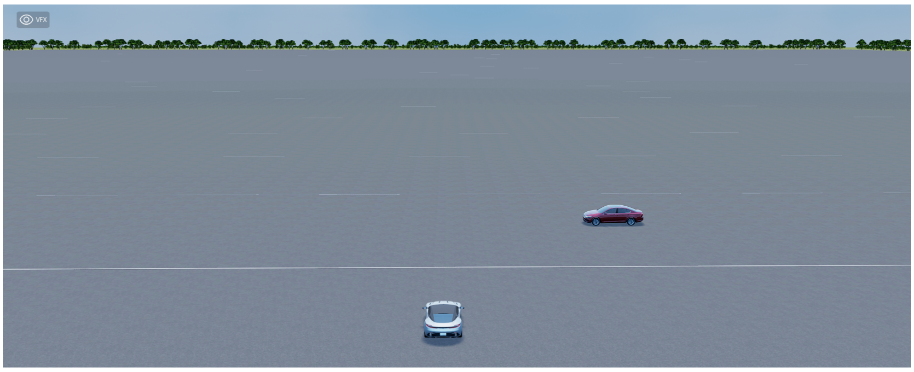


Figure 2.2: This is the first test when the car at 25m passes in front of the driver

2.2. Second scenario

The second scenario is called ‘Piazzale 2’. This scenario was created in such a way to study the influence of the distance and speed of the moving car on the perception of the driver. With this scenario, it is possible to verify also if the distance has more influence on the perception of the car with respect to the speed or viceversa. The structure is the same of the previous scenario ‘Piazzale 1’, in fact the starting point is the same roadrunner file. The difference is that instead of 1 ‘traffic car’, there are 6 of them. These cars are placed as follows:

- 2 cars at 25 m
- 2 cars at 75 m
- 2 cars at 150 m

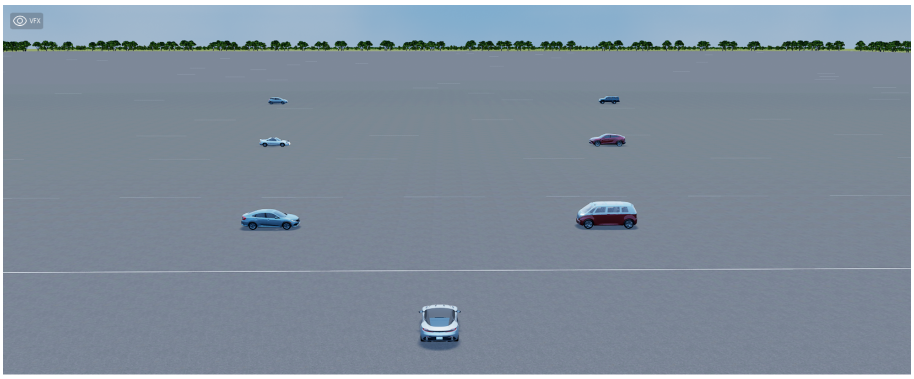


Figure 2.3: This is the second scenario when all 6 cars enter the field of view of the driver

For each pair of cars, one is positioned to the left and one to the right of the ego car. In all the different sub-scenarios, the 6 cars are always present; they may differ one from the other by their cruising speed selected randomly from the 3 previously adopted:

- 30 km/h
- 70 km/h
- 110 km/h

To get the lateral distance of the traffic cars it was used again the MATLAB script ‘Calcolo_distanze’. Here as before, the cars appear in the field of view of the driver 15 seconds after the start of the test, regardless of their speed.

2.3. Third and fourth scenarios

Figure 2.4: Third scenario at the start of the test

Figure 2.5: Fourth scenario at the start of the test

The third and fourth scenarios are created from a new scene designed in roadrunner. Basically, the only change between the two consists in the portion of the scenario considered for the test. These tests were created to study where the drivers look when driving along a straight road or inside an intersection. This test is carried out with the help of the eye tracker. The aim is to study what are the possibilities that a driver may look in front of the car, on the dashboard, on the side mirror or elsewhere. Thanks to the eye tracker software it is possible to calculate for how long a driver looks in each section of its field of view and understand better when there is a bigger risk of distraction. The road is shaped like a cross: one longitudinal straight and a transversal straight. The road style is a simple one, with 2 lanes, 1 per direction of travel and a central dashed line to allow overtaking. At the intersection of the 2 roads, there are vertical stop signs for all the 4 incoming straights and markings on the ground. The third scenario is called 'Straightroad' and exploits the first part of the main straight. It consists of 6 sub-cases:

1. Only the ego car is present inside its lane.
2. There is the ego car and a car parked in the opposite.
3. There is the ego car and a car parked in the same lane.
4. There is the ego car and a car that moves at 50 km/h on a straight trajectory in the opposite lane, so of course it moves in opposite direction with respect to the ego car.
5. There is the ego car, the parked car in the opposite lane, and a car that moves in the opposite lane that overtakes the parked car.
6. There is the ego car, the car parked in the same lane, and a car that moves in the opposite lane.

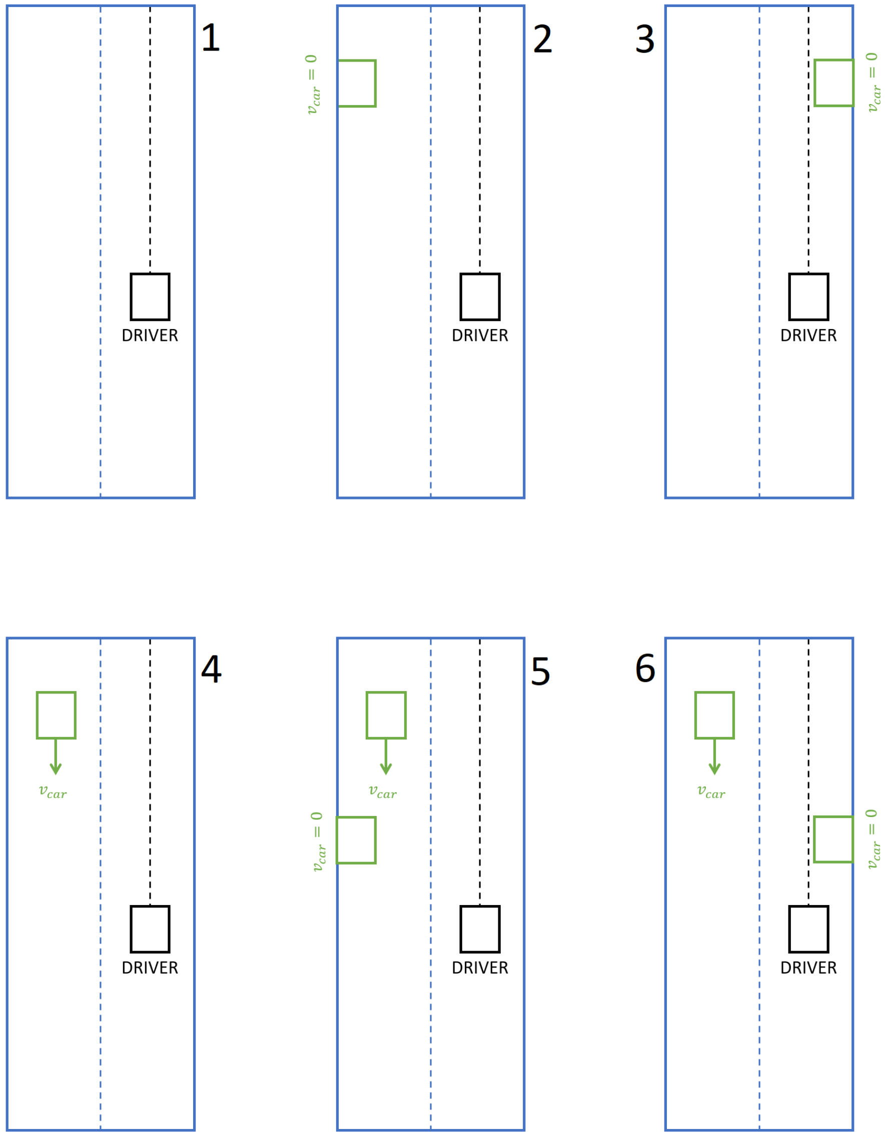


Figure 2.6: Scheme with all the 6 sub-cases for the straightroad scenario

Figure 2.7: 6th sub-case of Straightroad

The fourth scenario is called ‘Crossroad’ and takes into consideration the intersection between the 2 main straights. The purpose of this scenario is to study where a driver looks and for how long when encountering an intersection. Also in this case it is essential to use the eye tracker to get all the useful data. Similarly to the previous scenario, it is possible to study the main areas of interest for the driver, and get what may be possible sources of distraction. It is also made up of 4 sub-cases:

1. There is only the ego car, positioned 50m before the stop sign.
2. There is the ego car, and a moving car which comes from the left side of the driver of the ego car, that approaches the intersection at a constant speed of 50 Km/h.
3. It is equal to the previous scenario, except that the moving car comes from the right side of the driver of the ego car.
4. There is the ego car, the car coming from the left side and the car coming from the right side.

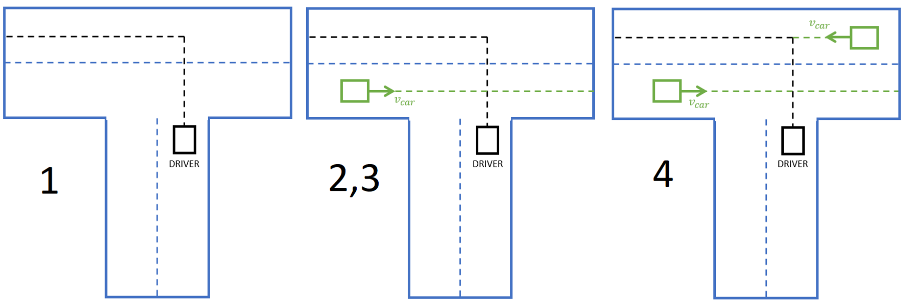


Figure 2.8: Scheme with all the 4 sub-cases for the Crossroad scenario

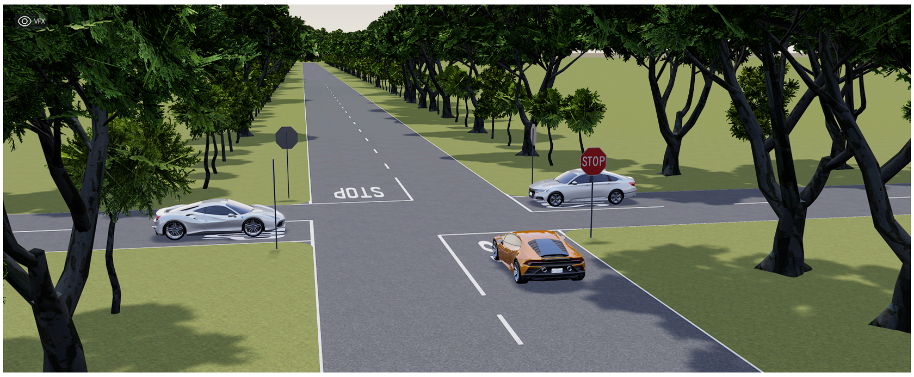


Figure 2.9: 4th sub-case of Crossroad scenario when the driver reaches the stop sign and cars arrive from both sides

Both moving cars have a lateral distance of 100 m from the ego, meaning that they encounter the visual field of view of the driver at the same instant of time.

2.4. Scenarios with disappearing car

There are two scenarios equal to 'Piazzale 1' as far as it concerns road geometry. The first scenario is created to evaluate the ability of drivers to estimate the position of a car. For this reason parked cars will disappear after some seconds. The second scenario is created with the aim of studying the ability of drivers to predict the velocity of moving cars. Also in this case the car will disappear and reappear to evaluate this prediction ability.

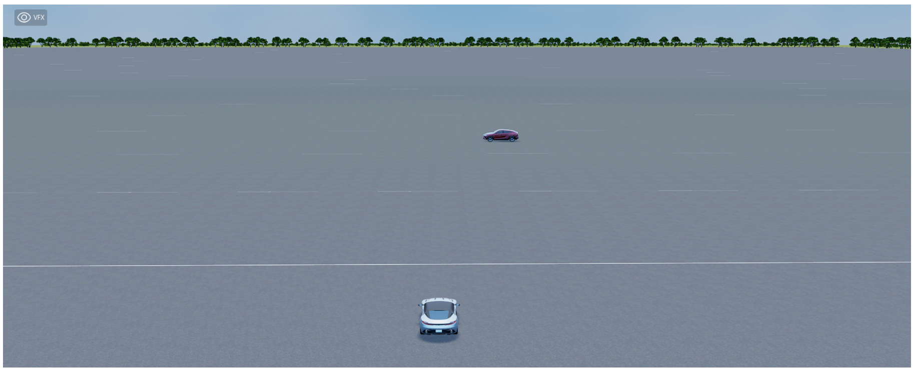


Figure 2.10: This is the disappearing car scenario with the parked car at random coordinates

1. In the first sub-scenario there is the ego car positioned as in the first scenario, and there's another car standing still. After some seconds from start, this car disappears. The procedure to position the car and make it disappear will be discussed in a

dedicated paragraph. Once the car disappears, the driver has to drive where it is believed that the car was placed.

2. In the second scenario there is the ego car located at the same starting point, and another car that moves in front of the ego car along the x axis. It moves at constant speed and after some seconds it disappears. Then after a second interval of time, the car appears again. This test is carried out with the eye tracker, because the driver has to look where it is believed that the car will appear again, and the eye tracker allows to know exactly where the driver looks.

2.5. How to manage the motion of the cars

2.5.1. First method

During the tests, the cars move in front of the driver in different conditions. It is important to keep under control the position and speed of these cars since it is desired to study the influence of these parameters on the perception of the cars by the drivers. As said previously, cars are positioned inside the scenario using the programme by VI-Grade called 'Worldsim', which gives the possibility to insert the x,y,z and yaw angle of the vehicle and its behaviour. The options are:

- Wander, in which the car moves randomly inside the road
- Trajectory, in which the car follows a straight trajectory by defining the destination point
- User controlled, in which the vehicle will be controlled by the driver inside the simulator

Of course the ego car, the one controlled by the driver, has a user controlled behaviour. Regarding the moving cars, the most appropriate option is the 'trajectory' one. The problem of this option is that it isn't possible to impose a constant travelling speed, since it tries to reach the target point in the fastest possible way. As a consequence, this option can't be applied. To solve this problem, it was necessary to contact the support from VI-Grade, to get a solution. It consists of a MATLAB code and a Simulink model to conduct a co-simulation between Worldsim and MATLAB. Basically, the MATLAB code called 'Launcher.m' is capable to launch Worldsim and open the scenario selected inside the code. Then it is necessary to open the Simulink model. Here, there are some blocks from the VI-Grade library, accessible adding a special command in MATLAB 'addpath_vicrt_20'. The blocks required are:

- 'File .res reader' block, which takes as input the simulation of a vehicle conducted in VI-CarRealTime and reads it (.res is the extension of the simulation files)
- 'File sender' block, which takes the information of the file .res and overwrites it onto the car to control inside the scenario

Now it's important to clarify how to get the '.res' files and why they are useful. These files are generated using the software VI-CarRealTime from VI-Grade. This programme is capable to conduct simulations of motions on many different vehicles and extrapolate all kind of information for each time step, for example the tire slippage, tire vertical load, the speed of the vehicle, engine RPM, roll, pitch and yaw angle etc.

For this test, it was just necessary to perform a simulation of a simple manoeuvre. There isn't a straight line manoeuvre, but this issue can be simply overcome by selecting a step steer manoeuvre, and imposing the steer angle equal to zero, so that the car keeps travelling along a straight line, with an imposed speed. The coordinates and the yaw angle of the starting point were selected to match the coordinates of the scenario; in this way the car will move exactly where it was decided. Once everything is prepared, the simulation can start in files-only mode and many different files will be produced, including the .res. Now this file can be used to make the Simulink model run correctly.

Note that each file '.res' is suitable to be used for just one sub-scenario, so it was necessary to produce many files to satisfy each position and speed constraint. As a consequence, depending on the particular sub-scenario selected, there is a switch in the Simulink model that automatically selects the correct file res reader block. As an example, if there is one single car moving at 3 different speeds, there will be 3 file res reader block (one for each speed simulation) entering the switch, and only one will be given in input to the file sender block.

This approach shows some drawbacks:

- First of all, it was necessary to produce many files, so a lot of memory is required
- Adding one file res reader block significantly increases the time required to start the Simulink simulation; as a comparison, in 'Piazzale 2' it was necessary to put 18 of these blocks, 3 for each car: it can take up to 1 minute just to start the simulation, which isn't acceptable
- If it is required to change a parameter like the position or speed of the cars, it is necessary to run a new simulation, which is time consuming

After some experimentation, it was found a faster and easier way to run the simulations, allowing to reduce the computational cost and increase the easiness to change parameters compared to this procedure.

2.5.2. Second method

The second method requires again the file sender block, necessary to communicate with the scenario opened in Worldsim. The difference is that with this method, it is possible to give as an input all information regarding the state of the vehicle by simply creating some variables in MATLAB and then sending them to Simulink. This procedure is much faster compared to the previous: in fact, even if one car has to move in different conditions, it is just necessary to modify the variables corresponding to the position and speed inside MATLAB, depending on the scenario. As an example, for 'Piazzale 2' it was required to put 18 of these blocks with the previous method, while with the new one, just 6 blocks. It also has the benefit of being very flexible in terms of modifications, since it is just needed to change the value of the corresponding variable, and the new simulation is ready to start.

In the following section the active part of this thesis is explained: the tests. A briefing was held with the technicians of the simulator to give instructions of what are the coefficients to change between each run of the test to speed up the process. In fact, between each run of the test, it is necessary to stop the simulation, update variables, run again the program and wait until the end of the upload to start a new run of the test. The MATLAB scripts were produced such that it was just necessary to change one variable from the value 1 to 3 to impose a different speed to the cars. It was adopted the same procedure in all the other scenarios: for example in the scenario with 6 cars, it was just necessary to assign a value between 1 and 3 to each car to make it move at the desired speed. This way of doing helped save time for each person and respect the time schedule. It also made easier for the technicians to update easy variables and avoid possible mistakes.

2.6. Test procedure

Some information of the participants are collected like age, gender, years of driving experience and average distance driven per year. Before entering the simulator, some basic operations are explained to the drivers. This is done so that they can get more familiar and used to the simulator, in order to make the experience as close as possible to the real world and have a more natural reaction to events that take place inside the simulator. All the tests are carried out in 2 sessions:

1. The first session (Test A) was held when the eye-tracker was still not available. In this session drivers experienced the first scenario 'Piazzale 1', the scenario with the disappearing car and the second scenario 'Piazzale 2' with the 6 cars.
2. The second session (Test B) comprehends 'Piazzale 1' with the moving car that disappears, and the 2 sub scenarios of 'Piazzale 3', 'Crossroad' and 'Straightroad'. These tests are carried out in a different moment because they require an additional tool called 'eye tracker'. This tool is an electronic pair of glasses specifically developed for driving simulators; it allows to see where the drivers look and have a better perceptions of their field of view.

2.7. Test A

The sample is composed of 29 people selected randomly, divided in 2 groups: the first with 22 people under the age of 30, and the second one with 7 people over the age of 30. The sample has an average age of 28,17 years and standard deviation of 8,86 years. On average, participants drove 7931 km with a standard deviation of 6871 km. This value is quite big, in fact there are some drivers with lot of experience, and some which just got their driving license. The most experienced driver obtained its driving license 45 years ago, while the least experienced obtained it some months before. On average, drivers obtained their driving license by 9,31 years, with a standard deviation of 9,16 years.

2.7.1. Test A1

When drivers come to the driving simulator, it is explained briefly what will happen in the 3 tests. It is also clarified where the stop button is and how to setup the seat to fit well and reach the pedals. Once they are ready inside the cockpit, it is moved to the center of the platform (to let people enter it, the cockpit is moved to the edge of the platform). When in place at the center, instructions are repeated to all the participants, through a microphone, regarding the first scenario:

1. Drivers must always look straight ahead.
2. A car will pass in front of them, and when they perceive it with their peripheral vision, they must press the stop button.

This test is composed of 9 runs, in which change the longitudinal position of the moving car (25, 75, 150m) and its speed (30, 70, 110 km/h).

This first test is done with the aim of studying the peripheral vision of a driver: the main goal is to identify the angle of the visual field of view with its deviation, and how the speed and distance of an approaching vehicle influences it. From the average value of all the drivers it is possible to assume an angle of inclination from the longitudinal direction at which it is highly probable that drivers start to perceive the moving vehicle and calculate its standard deviation. It was selected a delay between perception and pressing of the button from a paper that studied reaction times [22].

2.7.2. Test A2

When the first test is completed, drivers are kept inside the driving simulator and start the second test. In this test the scenario is equal to the previous one and there's a car parked in the proving ground with random coordinates. After some seconds it disappears. These are the instructions given to them:

1. Drivers must always look straight ahead.
2. A car will appear randomly in their visual field of view.
3. When the car disappears, they can engage the first gear with the steering wheel paddle and drive the ego car to reach the position where they think the car was parked.
4. Once they reach this position, they must press the stop button.

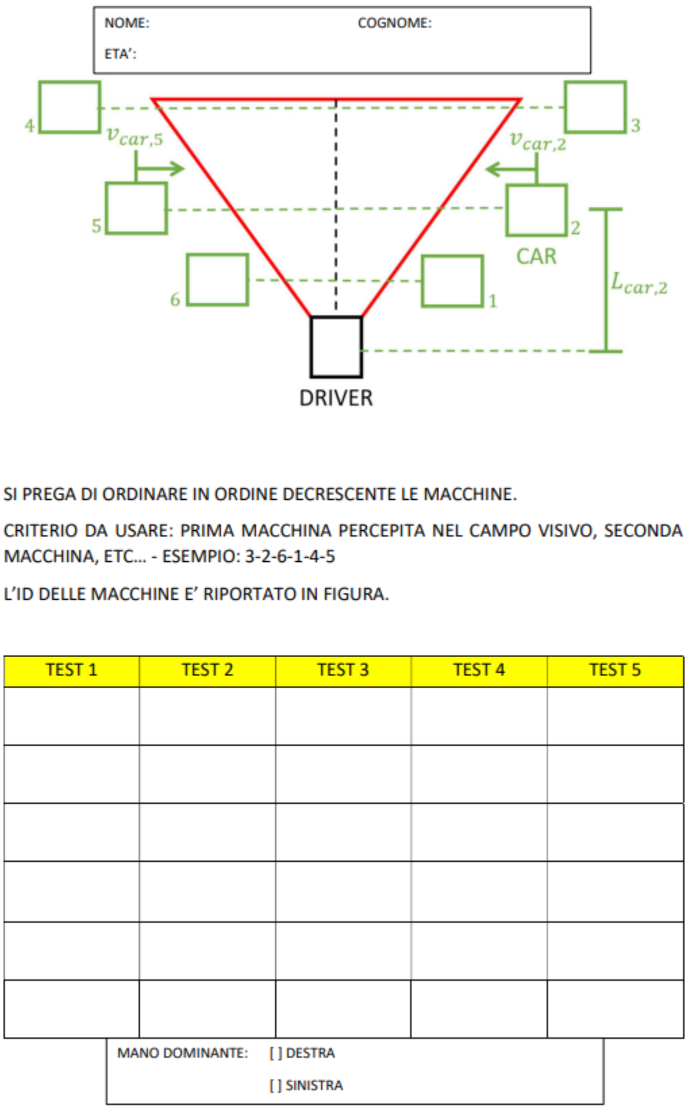
The parked car must have random coordinates, so it was created a matrix of random coordinates x and y to insert inside the running MATLAB code required to make the simulation work. After the drivers reach the estimated correct position, they must press the stop button. In this way it is possible to collect the data regarding the final position of the ego car (the driver) and calculate the difference with the actual position of the parked car. Drivers take 5 runs of this test.

The purpose of this test is to evaluate how good is the ability of the drivers to estimate the position of an object inside their field of view. The point is that if an object is in front of a person, the estimation of the position is quite precise, but if the object is more to the side, then probably this precision decreases.

2.7.3. Test A3

After this, drivers experience the last scenario called 'Piazzale 2', the one with 6 cars in motion. It was created a MATLAB code to assign the speeds to the cars randomly with normal distribution. Then these values were collected in a matrix given to the PC of the simulator. The benefit of this procedure is that between two runs of the test, the values for each car can be inserted with just one command easily and save time instead of writing manually each of them. These are the instructions given to them before the start:

1. They must always look straight ahead.
2. 6 cars will pass in front of them.
3. When the last car crosses their field of view, they can take the questionnaire and indicate the order in which cars have drown their attention, from the first to the last one.
4. In this test no buttons must be pressed and driving isn't required, so it is just sufficient to focus on cars.
5. In the questionnaire there is a graph indicating how cars are numbered, to help report them correctly.



TEST 1	TEST 2	TEST 3	TEST 4	TEST 5

Figure 2.11: This is the questionnaire for the 6 cars test

After the drivers prove each take of the scenario, some time is left to compile the questionnaire and give a subjective feedback of the scenario just experienced. Each driver had 5 takes of this test, so the data collected consists of 5 rankings for each person.

The aim of this scenario is to evaluate what are the main factors that influence perception of vechicles crossing the field of view. In this case the factors that influence the perception are the longitudinal distance and speed of the vehicle. Their effective influence will be investigated in the postprocessing of the data collected during the tests.

Tests are experienced by drivers in the order just reported. The total duration of all 3 tests is of 30 minutes. No driver felt sick during the unfolding of the tests.

2.8. Test B

The sample of this test is composed of 28 people randomly selected, 21 are under the age of 30, and 7 are over the age of 30. The average age is 30,32 with standard deviation 11,25. Participants have got the driving license on average for 12,32 years, with a standard deviation of 11,25. When participants arrive at the simulator, it is explained briefly what they will do and they get prepared to wear the electronic glasses ‘eye tracker’. First the participant wears it, then it is connected to the pc to start the software required to calibrate the glasses. To calibrate, the driver looks at a specific card with a target on it for the time required to conclude the calibration. This operation is repeated for all participants, to compensate for the different shape of each face and ensure constant quality of the data. Then drivers sit inside the simulator wearing the eye tracker. Then the cpu of the glasses is connected to the cockpit of the simulator by a LAN connection: this gives the possibility to control the start and stop of the registration directly from the control room of the simulator. Drivers won’t remove the glasses until the end of the complete test, to ensure that the calibration is valid through all the different scenarios.

2.8.1. Test B1

Just like for the first session of test, once the driver is ready, the cockpit is moved to the centre of the platform and instructions are repeated through a microphone, regarding the scenario of the moving disappear car:

1. A car will pass in front of the drivers.
2. They must keep looking to the car.
3. Suddenly, the car will disappear, and drivers must keep following the car as if it was still present.
4. After some seconds, the car will reappear. Drivers must keep looking at the car for about 5 seconds, then they can press the stop button.

This test is repeated 6 times for each participant, varying randomly the speed (30, 70 and 110 km/h) and the longitudinal distance. From this test the data from the eye tracker is collected: the coordinates at which drivers looked each instant of the simulation. It was necessary to do a calibration test to create a correspondence between the coordinates of gaze and the distance covered by the car (this procedure will be further discussed in the data collected for test B). The target of this test is to evaluate the ability of drivers to estimate the speed of a moving car. The perception is influenced by the speed itself and the longitudinal distance of the car.

2.8.2. Test B2

At the end of the first part, drivers stay inside the cockpit and test the second scenario 'straightroad'. In this case drivers will effectively drive in a road, so the simulator is switched from static to dynamic, to replicate sensations given by a real car in motion. Participants will keep wearing eye glasses also in this run. These are the instructions given:

1. Drivers will start on a long straightroad. They have to engage the first gear and start driving along this road.

2. Drivers will interact with some cars present on the scenario, simulating some situations that may happen in real life.

3. Drivers must drive according to the road laws, keeping the speed under the limit of 50 km/h.

4. When told, they can press the stop button.

In this scenario 6 different cases will be presented to drivers, like for example a car moving along the opposite lane, cars overtaking parked cars and so on. In this case the data collected is the point where the driver is looking. In post processing, some areas of interest (AOI) will be overlapped to the video of the scene, to get information about where drivers look.

The aim of this test is to obtain a model for each driver, describing the probability that this person is looking in a certain direction for each of the possible 6 cases. Then the models are merged together to obtain an average model for each of the 6 cases. These models will provide an estimation of how it is possible that a driver is looking in one direction rather than another.

2.8.3. Test B3

Finally, drivers experience the last scenario called 'crossroad'. Also in this case drivers will effectively drive just like in the previous scenario, so the simulator will be in 'dynamic' mode. Also for this test drivers will keep wearing the eye tracker. These are the instructions given:

1. Drivers will start near an intersection of 2 roads. They have to engage the first gear and drive to the intersection, where they will stop.

2. Drivers must look around to check that no car is arriving.
3. If the road is clear, they can take the left turn and drive along that road.
4. After ten seconds of driving in the new road, they can press the stop button.

Drivers will repeat this test 4 times, each time will simulate a different situation that may happen in real life: case with no cars, case with one car coming from the left, case with one car coming from the right and last the case with both cars approaching the intersection. In this case, like in the previous scenario, areas of interest will be overlapped on the video of the test to get the list of the areas where the driver looks during the maneuver.

The target is to calculate the time spent looking in each direction, to get the probability that the driver is looking in one direction rather than another. Just like for the previous scenario, a model is created for each driver, for each of the 4 sub cases. Then models will be merged to get 4 average models of probability.

3 | Data Processing

3.1. Data collected for test A

During the tests data were collected for each instant of time of the simulation: the speed of each car present in the scenario, linear speed, angular speed and so on. One important aspect is that for some tests it was useful just the data corresponding to the instant when the driver presses the stop button. When the test starts, the PC generates a CSV file containing all the info previously mentioned, so it is just required to look at the last row of the CSV file to extract the value needed. Each take of each driver has its separate CSV file, which is a lot of unused data, so during the analysis it is important to extract the values required and put them in matrixes to be used later in MATLAB. Another important aspect is that it is required to consider the reaction time that occurs between the perception of the car and press the stop button. It was selected a value tau=0,3 s from a paper that studied the reaction time between a stimulus and the pressing of a button [22].

So in the end, after a first selection of data, it was collected:

- For the first test, the coordinates X,Y of the moving car and the speed
- For the second test, the coordinates of the parked car and those of the ego car
- For the third test, the order in which cars are perceived, including also their speed
- For all tests, the age of the driver to investigate the performance of under 30 and over 30

3.2. Data analysis test A

Now it is time to start the elaboration of the data collected and show the results obtained so far. To reach this goal, it was used MATLAB to create 3 scripts, one for each part of the first test. With this program some calculations were performed on the values and plotted the results. It was also used Excel for the third part to plot some histograms, inserting values obtained from MATLAB.

3.3. First test: single car approaching from the side

First of all it was calculated the angle of the car when pressing the stop button for the 3 distances (25m, 75m and 150m) and for the 3 speeds (30 km/h, 70 km/h and 110 km/h): in total 9 angles. Then it was calculated the average angle at which drivers perceive the car for the 3 speeds, to investigate the influence of this factor (figure 3.1).

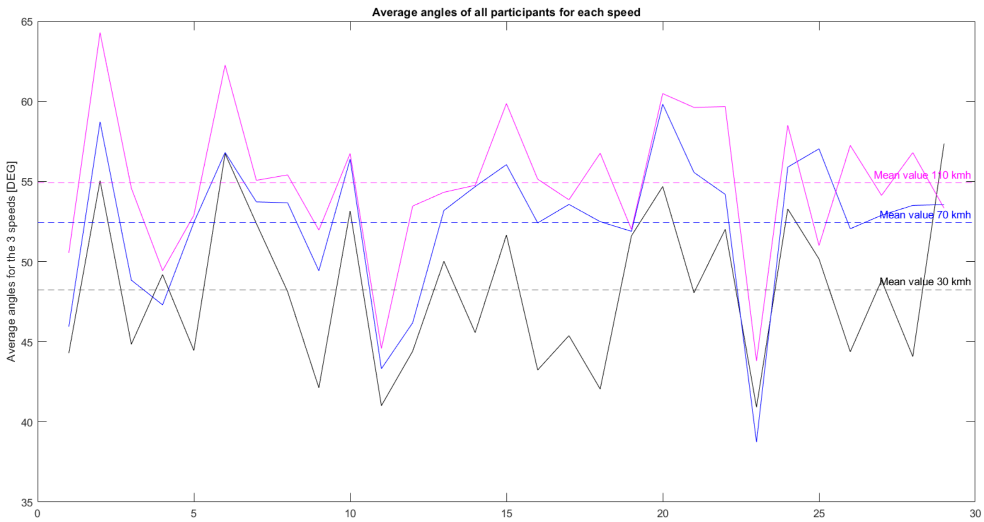


Figure 3.1: Mean angle for each participant

In this graph each abscissa (from 1 to 29) represents each driver, and the corresponding value represents the average angle of perception of the car for the 3 speeds. It is possible to see clearly that the average angle for everyone increases with the speed (represented by the 3 horizontal dashed lines). In fact, generally all drivers show a higher level of perception when the car is approaching at a faster speed, made exception for some drivers. With this research it is possible to conclude that generally speed affects positively the ability of the drivers to perceive an approaching car in front of them, since at higher speed cars can be perceived before slow cars.

The average angle of the field of view at 30 km/h is equal to 48,24°, with a standard deviation of 4,88.

The average angle of the field of view at 70 km/h is equal to 52,42°, with a standard deviation of 4,63.

The average angle of the field of view at 110 km/h is equal to 54,91°, with a standard deviation of 4,60.

The next step is to proceed with the calculation of a linear regression model to see if it is a good estimator for all participants for the angle of the field of view. So, in this case the regression model is calculated from the exact values obtained experimentally, divided in 3 classes to see again a possible influence of the speed on the perception.

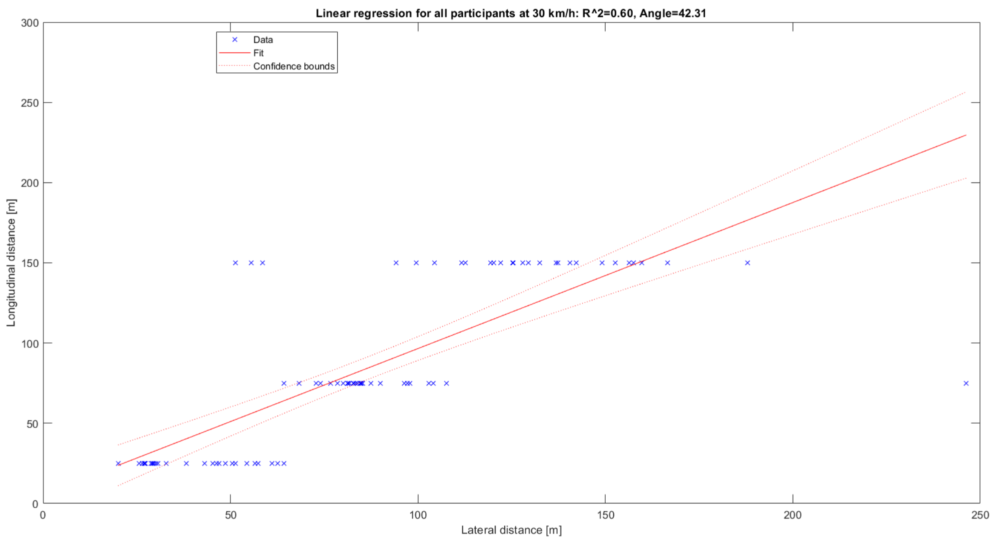


Figure 3.2: Linear regression for all participants at 30 km/h

Linear regression done for all drivers at 30 km/h: an estimated angle of the linear fitting of 42,31° and an $R^2 = 0,60$ so it seems an acceptable estimation.

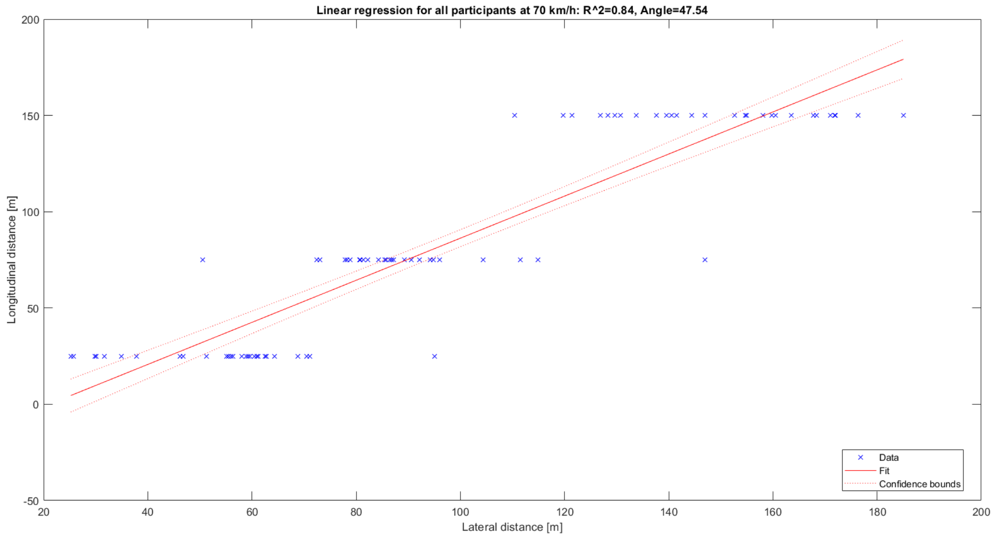


Figure 3.3: Linear regression for all participants at 70 km/h

Linear regression done for all participants at 70 km/h: in this case we can see a better fitting since R^2 has an higher value of 0,84. It shows an higher angle of the fitting of 47,54°, a trend in accordance with the average value calculated previously.

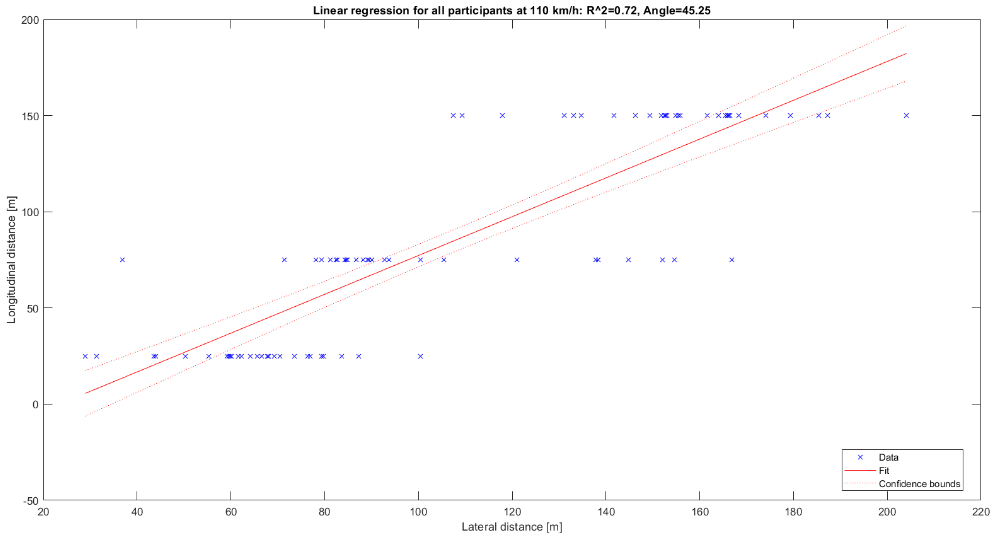


Figure 3.4: Linear regression for all participants at 110 km/h

Linear regression obtained for all participants at 110 km/h: again a good fitting with an R^2 value of 0,72. The estimated angle, obtained from the inclination of the straight line, gives a value of 45,25°.

The next investigation is related to the effect of age on physical performance. In particular, it was repeated the same analysis done till this point, but dividing the group of people in under 30 and over 30. In this case the sample was composed by 22 people under 30 and 7 people over 30.

Let's start with the average values reported for both groups.

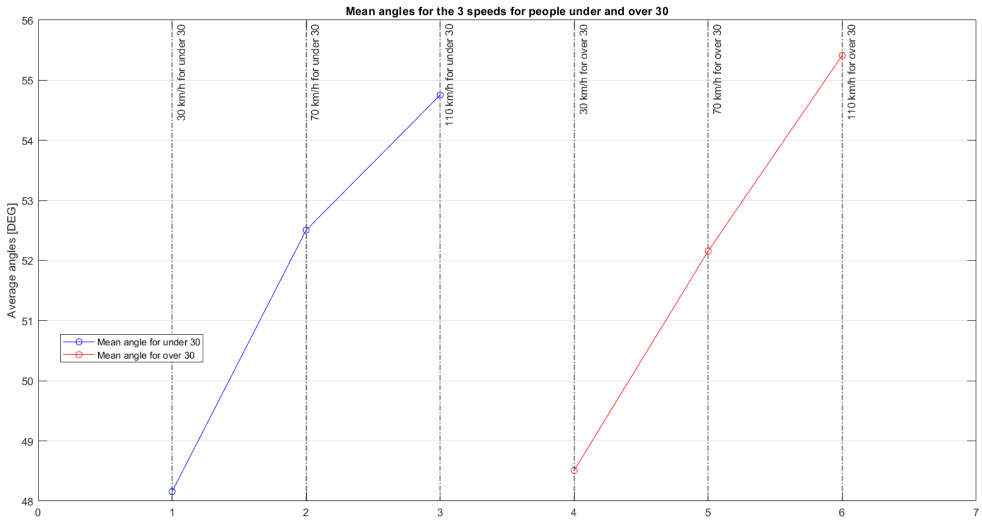


Figure 3.5: Mean angle for under 30 and over 30

The graphs reports the average angles for under 30 and over 30, for the 3 speeds considered. Generally under 30 show a lower angle of vision compared to the over 30, made exception for the speed of 70 km/h.

x=1 represents mean angle for under 30 at 30 km/h that is 48,16° with standard deviation=4,48

x=2 represents mean angle for under 30 at 70 km/h is 52,51° with standard deviation=4,42

x=3 represents mean angle for under 30 at 110 km/h that is 54,75° with standard deviation=4,38

x=4 represents mean angle for over 30 at 30 km/h that is 48,51° with standard deviation=6,37

x=5 represents mean angle for over 30 at 70 km/h that is 52,16° with standard deviation=5,63 x=6 represents mean angle for over 30 at 110 km/h that is 55,41° with standard deviation=5,61

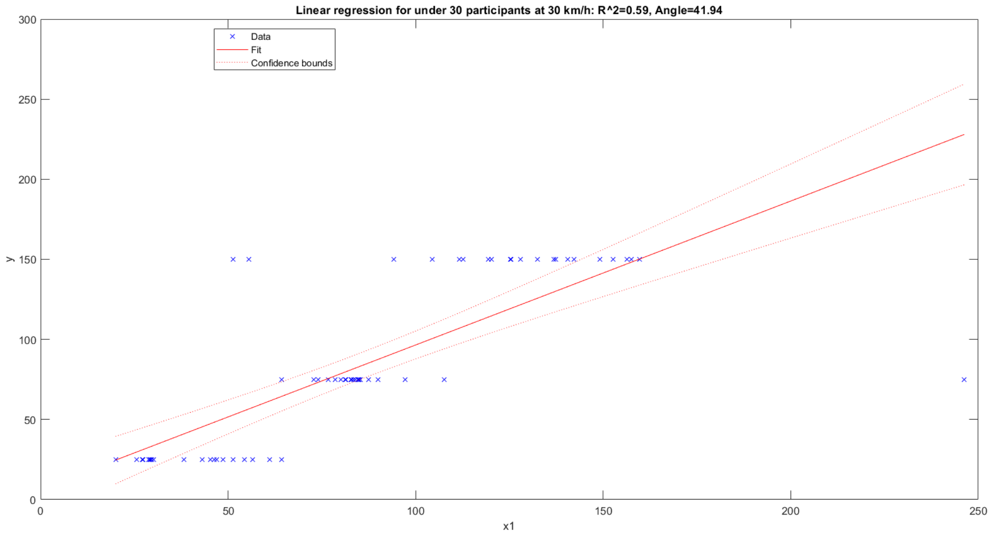


Figure 3.6: Linear regression for under 30 at 30 km/h

Linear regression for under 30 at 30 km/h: an estimated angle of 41,94° and an R^2 of 0,59. The angle is slightly smaller than the linear regression done for all participants.

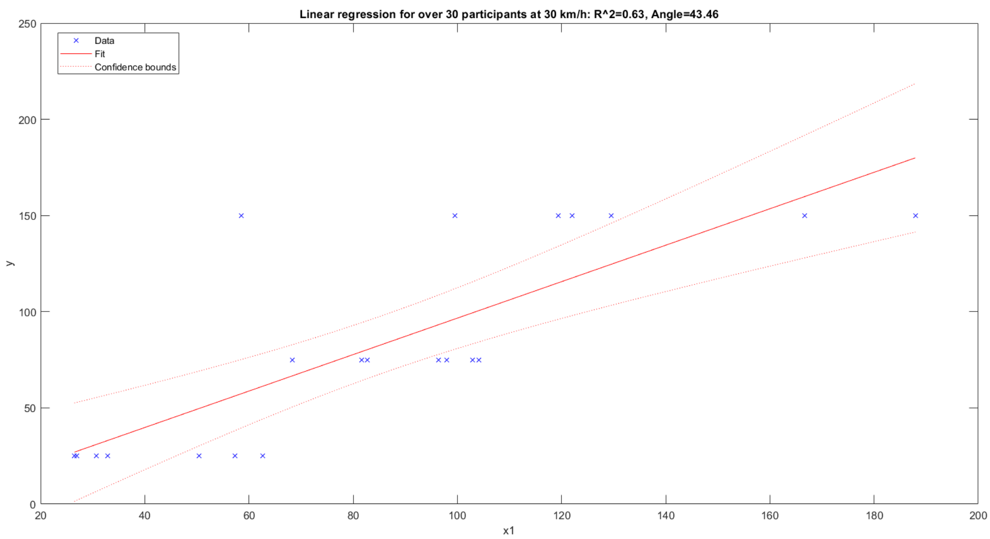


Figure 3.7: Linear regression for over 30 at 30 km/h

Regression for over 30 at 30 km/h: an estimated angle of 43,46° that is bigger than the same value for under 30 and an R^2 value of 0,63.

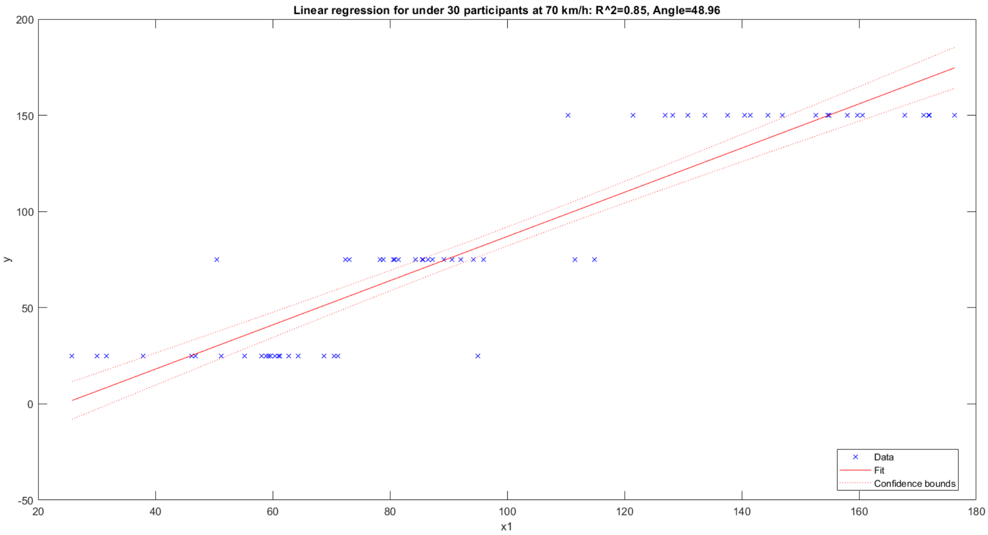


Figure 3.8: Linear regression for under 30 at 70 km/h

This is the regression done for under 30 people at 70 km/h: an estimated angle of 48,96°, which seems to be in accordance with the trend of higher angle at higher speeds, and an R^2 value of 0,85, so a quite good fitting.

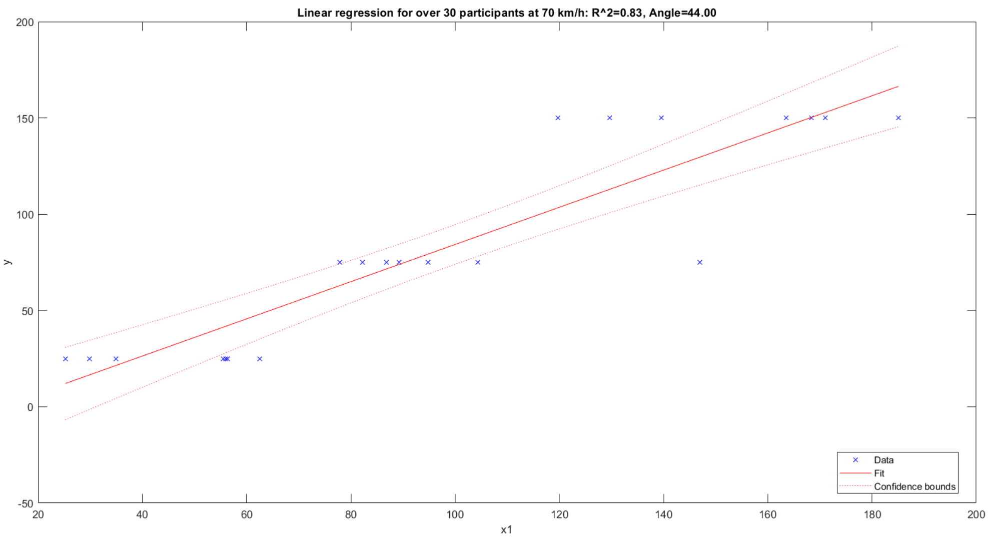


Figure 3.9: Linear regression for over 30 at 70 km/h

Regression for over 30 at 70 km/h: an estimated angle of 44,0° that is lower than under 30 people, in accordance with the trend seen by the average values. The R^2 value is equal to 0,83.

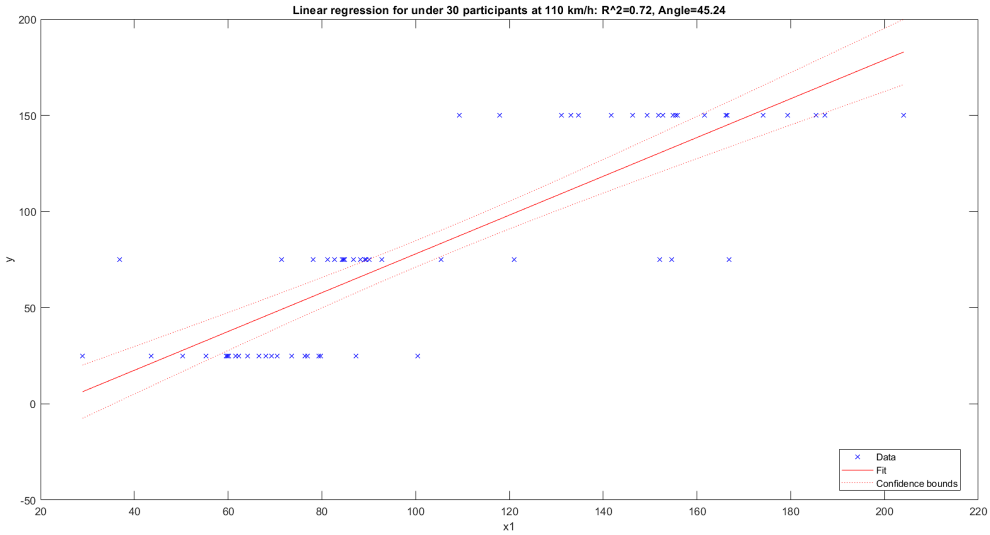


Figure 3.10: Linear regression for under 30 at 110 km/h

This graph is related to under 30 people at 110 km/h: an estimated angle of 45,24° which is lower than 70 km/h but sensitively bigger than 30 km/h and an R^2 value of 0,72, showing the fact that in this case experimental values are more dispersed around the mean value.

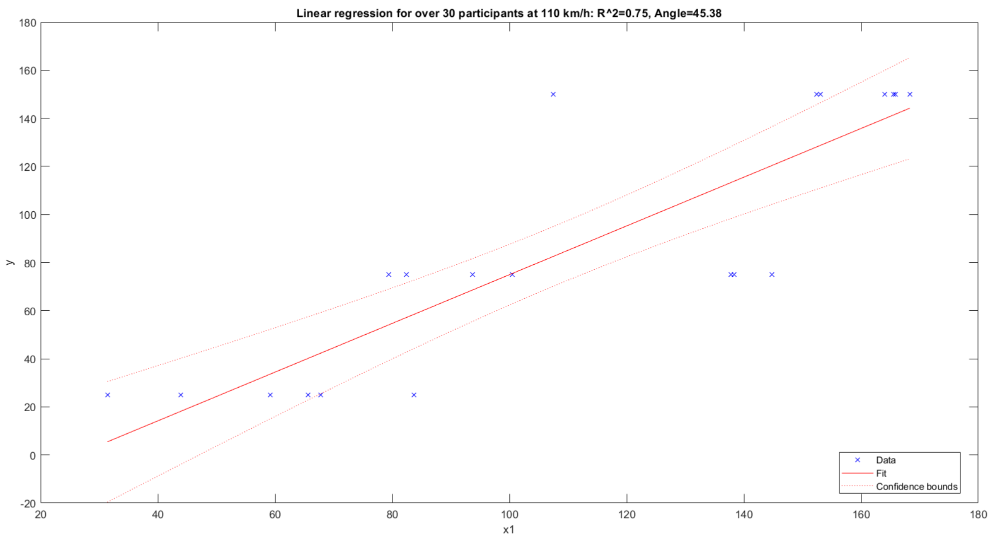


Figure 3.11: Linear regression for over 30 at 110 km/h

Lastly, the regression for over 30 people at 110 km/h: the estimated angle is equal to 45,38° which is slightly bigger than the same value obtained for under 30 people. R^2 is equal to 0,75.

3.4. Second test: disappearing car

Here is reported an example of the data extracted from the second part of the test: the blue stars represent the position where the parked cars were placed, while the red stars represent the position where the driver stopped the ego car. The origin (x=0, y=0) is the point where all drivers start the test (figure 3.12).

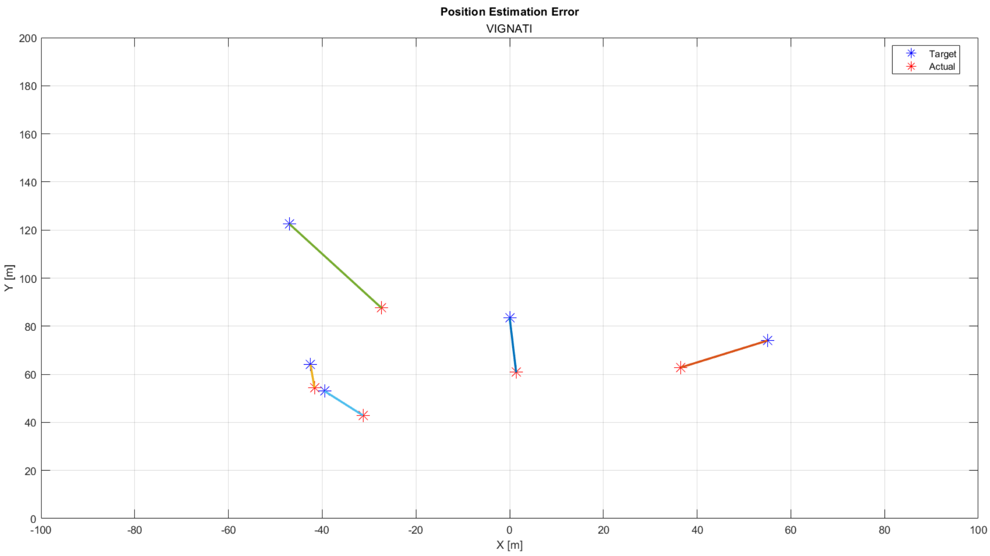


Figure 3.12: Example of the data collected

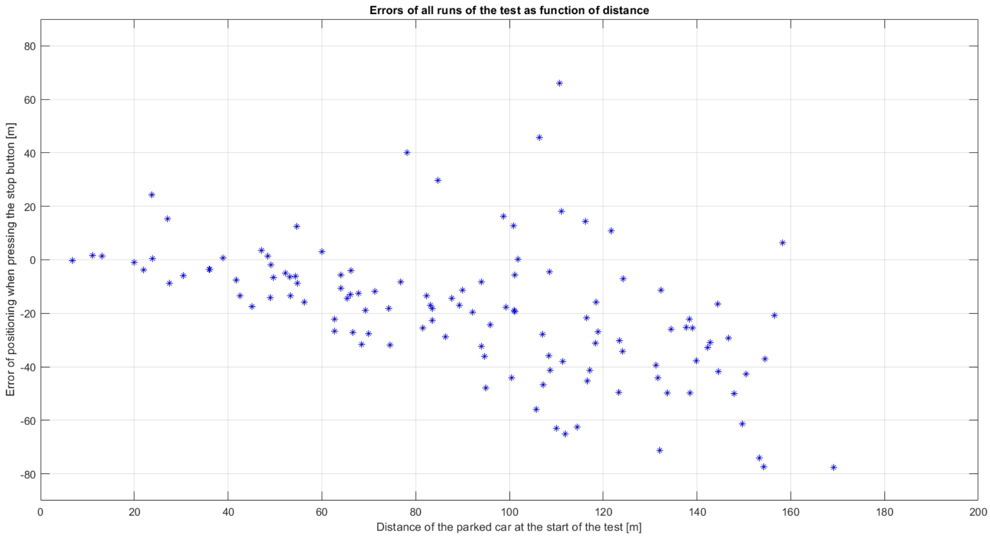


Figure 3.13: Error on position as function of distance

This graph (figure 3.13) shows the error of distance of the drivers as function of the initial distance of the parked car. A positive error means that the driver stopped further than the parked car, a negative error means that the driver stopped closer than the parked car. It is possible to see a trend: when the parked car is close to the ego car, the error is usually kept low by the drivers, and when its distance progressively increases, also the error seems to increase linearly.

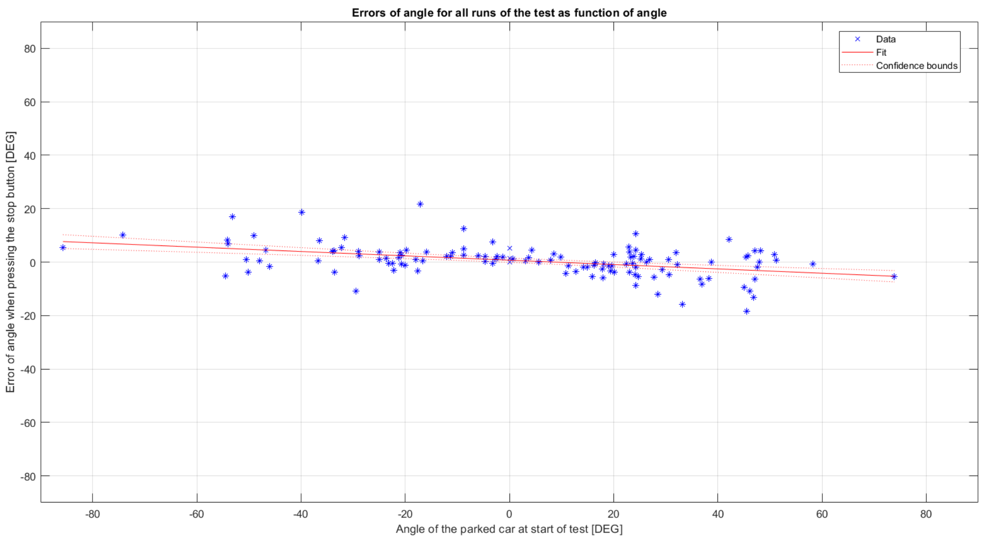


Figure 3.14: Error on angle as function of angle

This graph shows the error on the angle as function of the angle of the parked car. This trend suggest that drivers tend to underestimate the angle of the car. It's possible to notice that generally drivers commit a bigger error on the evaluation of the distance, compared to the angular position. In fact, the max error on the distance is of 74,34 %, while the max error on the angle is of 57,15 %.

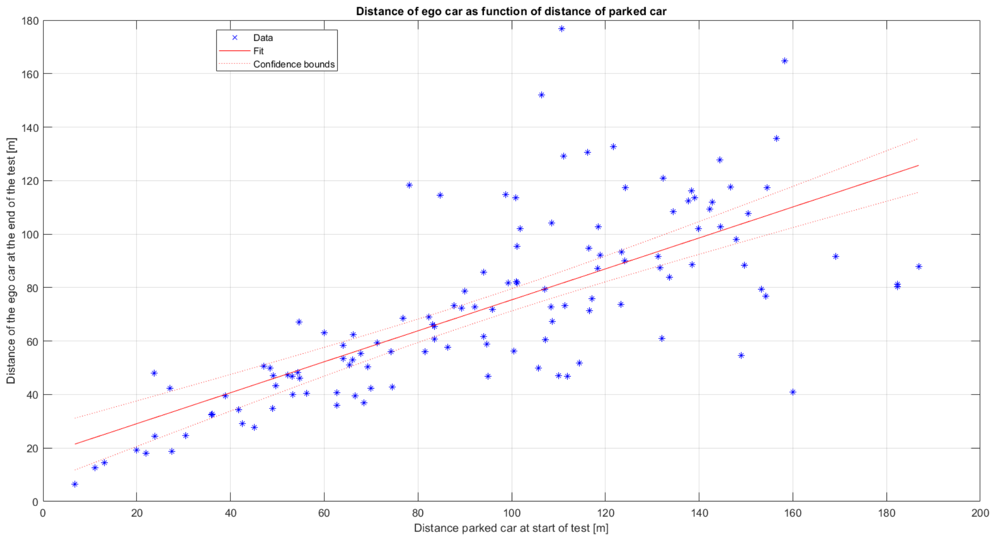


Figure 3.15: Distance of ego as function of distance of parked car

In this graph it is reported the distance of the ego car as function of the parked car. The linear interpolation gives as result $R^2 = 0, 52$, an acceptable value to describe the linear relationship between these 2 quantities.

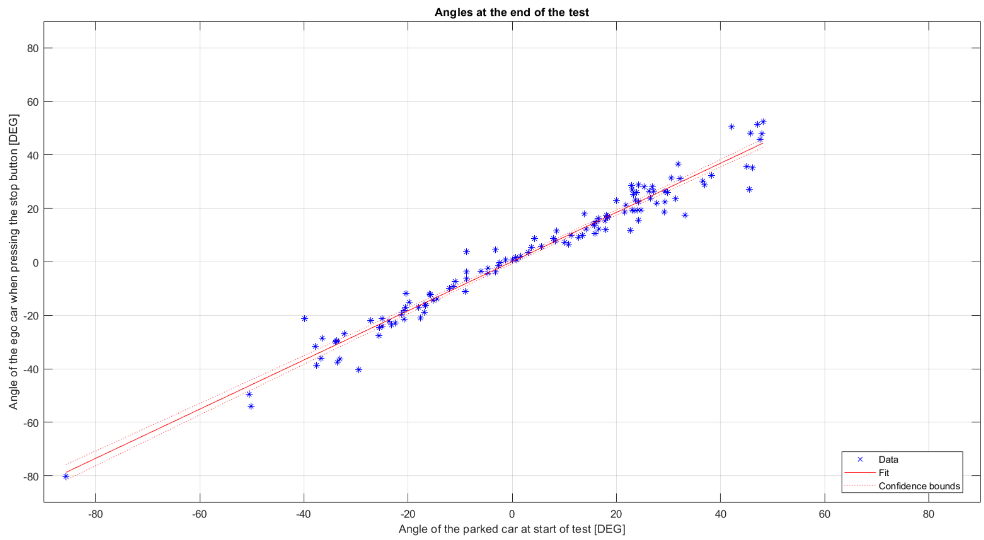


Figure 3.16: Angle of ego as function of angle of parked car

In this graph it is reported the angle of the ego car as function of the angle of the parked car. In this case the linear interpolation has $R^2 = 0,97$, so it describes quite well the linear relationship between these 2 quantities.

Now the same analysis is repeated but dividing the group of people in under 30 and over 30.

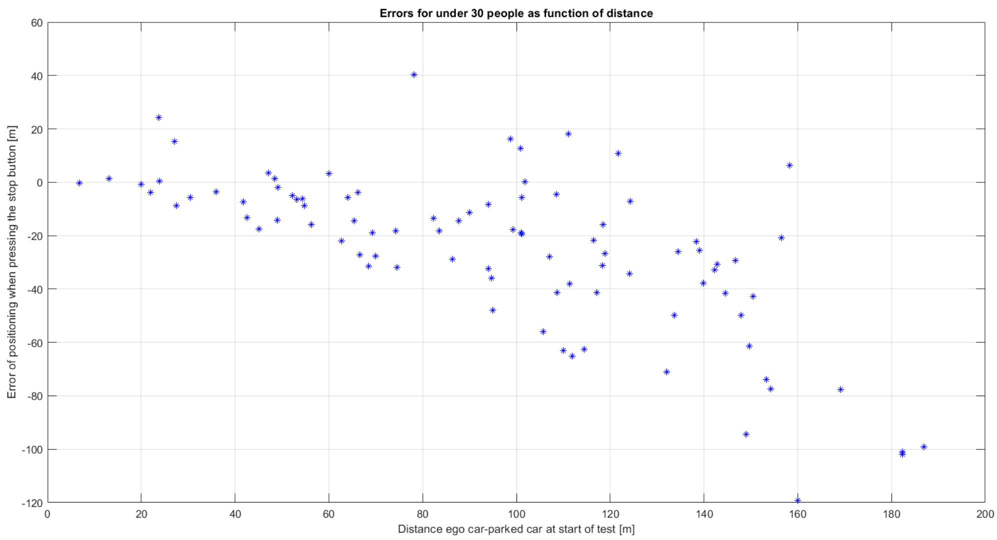


Figure 3.17: Error as function of distance for under 30

This is the error of distance as function of the initial distance for under 30 people.

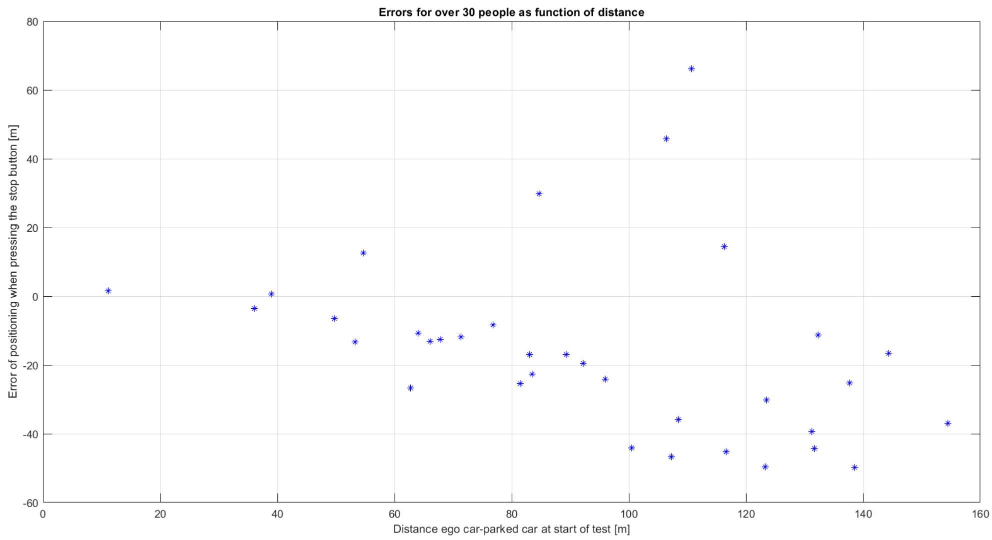


Figure 3.18: Error as function of distance for over 30

This is the error of distance as function of the initial distance but considering over 30 people. This trend is present in both groups. Under 30 show higher magnitude of the error, compared to over 30.

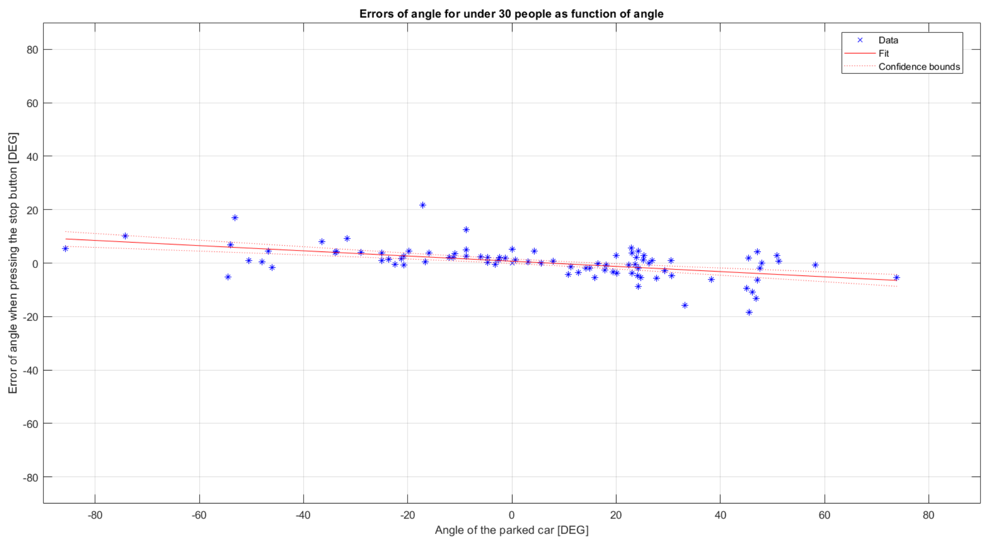


Figure 3.19: Error on angle as function of angle for under 30

This is the error on the angle as function of the angle of the parked car for under 30.

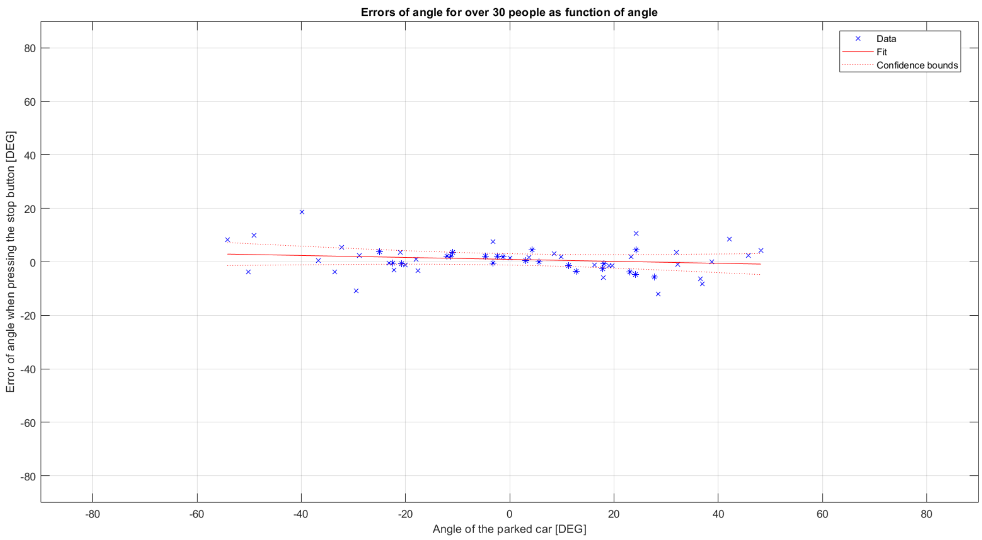


Figure 3.20: Error on angle as function of angle for over 30

This is the error on the angle as function of the angle of the parked car for over 30.

3.5. Third test: 6 cars

For this test some histograms are used to represent results. The aim is to put in evidence what are the main factors that influence the perception of the cars approaching from both sides, whether it is the distance or the speed of those cars.

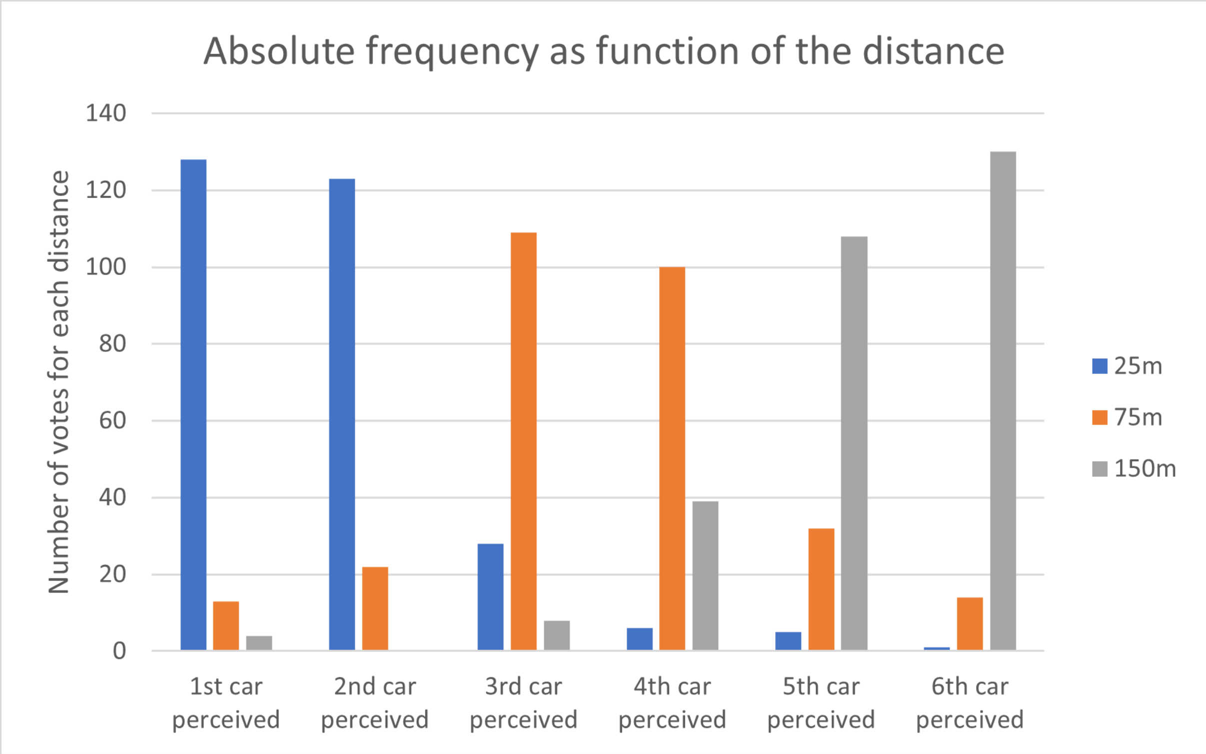


Figure 3.21: Frequency as function of the distance

In this graph it is reported the frequencies at which cars are classified as first, second and so on. As expected, the first 2 cars to be perceived are those closer to the driver (at 25 m). Then the third and the fourth are the 2 at 75 m, which corresponds to the second row of cars, and at last the 2 remaining cars most distant from the driver.

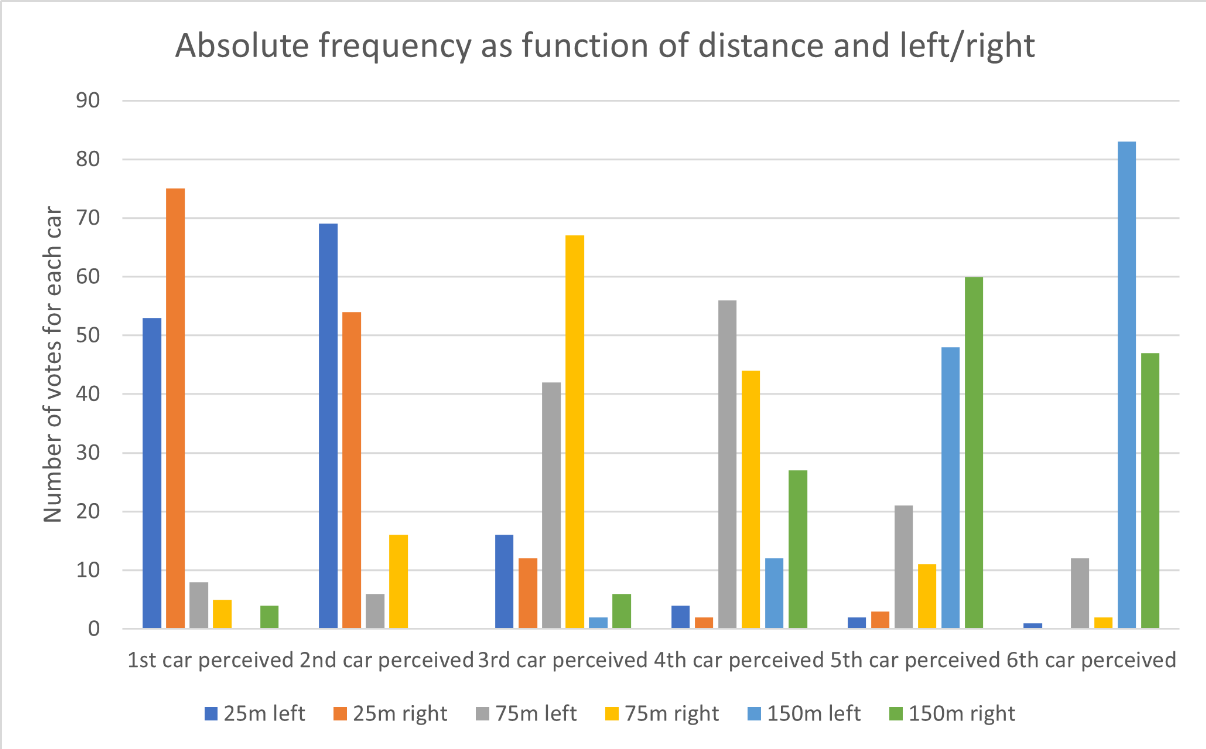


Figure 3.22: Frequency as function of the distance, left and right

The investigation went more in depth to find additional factors to influence the perception. In this case it was considered also the side where the car approaches the driver, and these are the results. The cars coming from the right are perceived earlier than the ones coming from the left. So in order, it is perceived the first row of cars, and between these two the first one is the right one, followed by the left one. The same trend applies also to the second and third rows of cars.

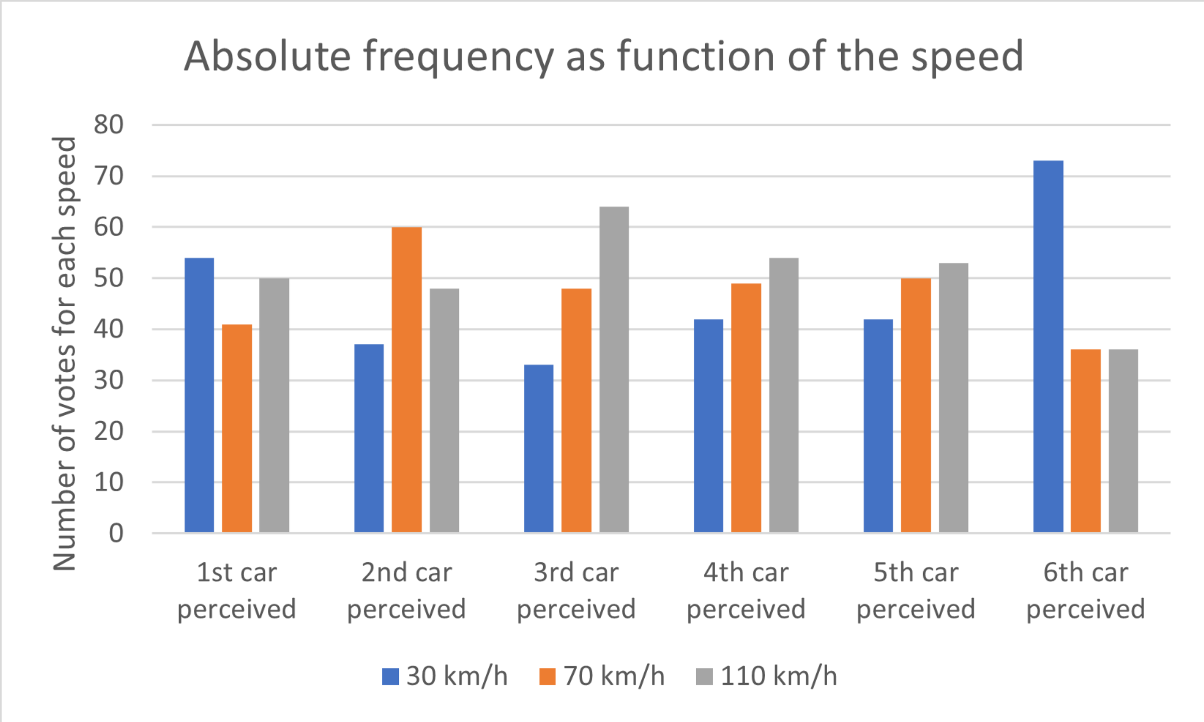


Figure 3.23: Frequency as function of speed

This graph (figure 3.23), shows the frequency as function of the speed of the cars approaching. At first sight, it seems that there is not a linear relationship between the speed and the frequency. It just seems that cars coming slower are perceived more frequently than the others. As a consequence, it was necessary to keep in account also the side of the cars.

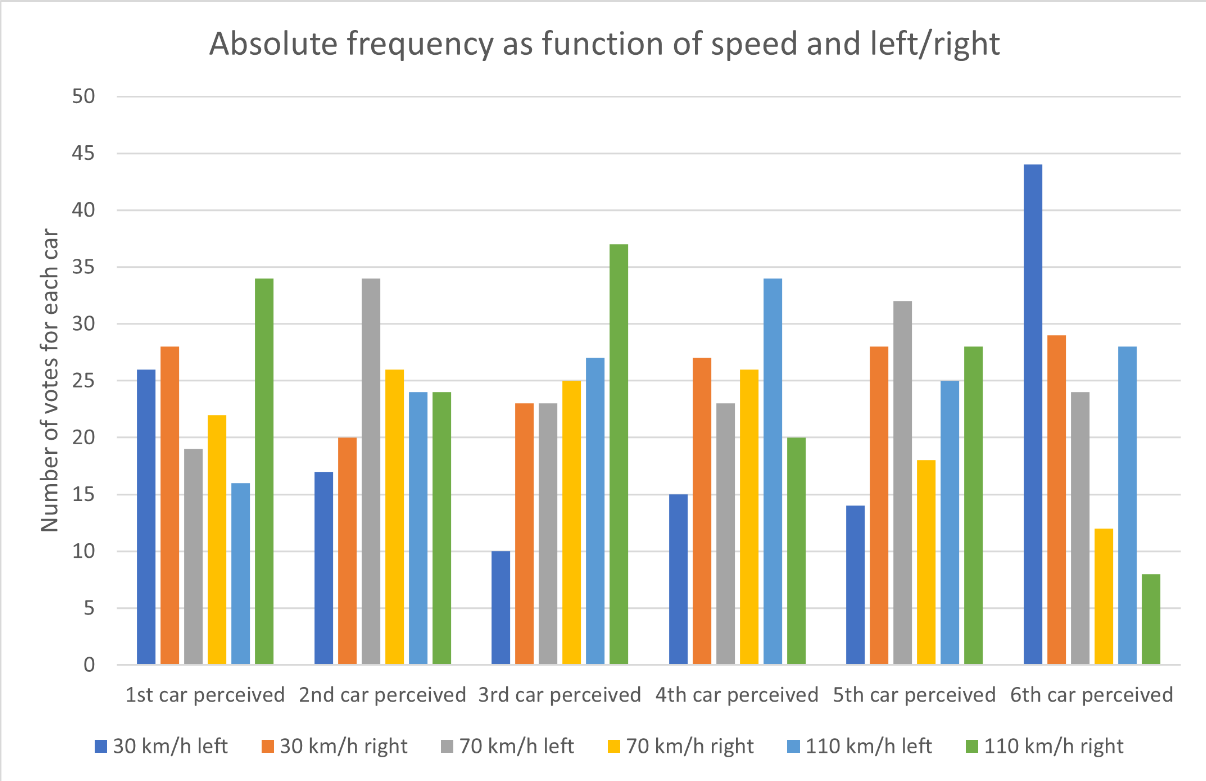


Figure 3.24: Frequency as function of speed, left and right

By considering each car individually, it is possible to see clearly that in reality cars at lower speeds are perceived less frequently than the faster ones. It is possible to see that, differently from the case of the distance, the speed seems to have less capacity to attract the attention of the drivers. In fact the histogram of the distance shows 2 very high columns and 4 very low columns, while in this case the height of the columns is more comparable. Generally, cars coming faster are perceived more often than the ones coming slower. In fact just the last car perceived shows an higher frequency of cars coming at 30 km/h. So it is possible to conclude that generally speed has less capacity to draw the attention of the driver, compared to the distance.

3.6. Data collected for test B

For this test it was collected data regarding the cars interacting in the scenario and the data coming from the eye tracker, in particular:

- For the first test 'moving disappear car', it was collected the speed of the car and the longitudinal distance from the ego car. From the eye tracker it was extracted the coordinates of gaze of the driver in the instant the car disappears, the instant the car reappears and the instant the driver moves the eyes to align with the car. Since the output of the glasses is a video, the coordinates of gaze correspond to the pixel of the point where the driver was looking. The resolution of the video is HD, meaning that the pixels can go from 0 to 1920 horizontally and from 0 to 1080 vertically.

- For the second test ‘straightroad’, it was collected only data coming from the eye tracker, because all 6 runs of the test are always equal and coordinates of the cars were decided before the start of the test. In this case, just like the previous scenario, it was collected the gaze data of the driver.
- For the third and last test ‘crossroad’, it was collected just the data coming from the eye tracker, because all 4 runs of the test are always equal for each driver and coordinates and speeds of the cars where determined before the start of the test. For this scenario it was collected the gaze data of the driver, just like for the first and second tests.

3.7. Data analysis test B

To perform the data analysis of this test, it was necessary to extract first the values from the data of the eye tracker using ‘Tobii Pro Lab’, which is the software supplied by the company that produces the eye tracker. This software gives in output excel sheets for each run of the tests. Then these files are imported into MATLAB and analysed with 3 scripts specifically created for each scenario.

3.8. First test: moving car that disappears

To analyse data from the first test, it was necessary to create a correspondence between the distance covered by the eye measured in delta pixels, and the distance covered by the cars measured in meters. To do so, it was performed a calibration session at the simulator. Basically, it was identical to the first test, with the exception that the car never disappears. It was taken the gaze of data of this scenario and imported in ‘Tobii Pro Lab’. The simulation was repeated 2 times for 3 longitudinal distances and for the 3 speeds considered. Then in MATLAB it was created a correspondence: for each speed it was calculated the delta pixel of the gaze between 2 intervals of time, the distance run by the car for the 3 distances. Then, it was possible to create a linear regression describing the ratio the distance covered by the car and the delta pixel, for whatever longitudinal distance going from the minimum to the maximum distance. This was repeated for the

3 speeds. So, in conclusion, with there linear regressions, it is possible to know for each test, how much is the distance estimated by the driver's eyes, by knowing the delta pixel of the gaze from the instant the car disappears, to the instant the car reappears. If the distance estimated is lower than the real distance run by the car, of course the driver underestimated its speed and viceversa.

To calculate the estimated speeds, these equations were used: First of all, from the linear interpolation from calibration, it is obtained the relation delta meter/delta pixel (dx/dp) for the longitudinal distance of that specific run of the test. Then it is calculated deltaL:

deltaL=dx/dp*deltapixel

Where deltapixel is the difference in pixel from the instant the car reappears and the instant the driver realigns the gaze to the car (presuming the driver didn't exactly estimate the position of the car). The value deltaL represents the error in meters of estimation of position of the car, plus the distance travelled by the car during the two instant of time (the correction of gaze usually equal to around 0,4s so the distance travelled by the car in that time is relatively small).

Then it is possible to calculate deltax: If the driver was looking after the car (overestimate the speed)

deltax=deltaL-vcar*(t3-t2)

If the driver was looking behind the car (underestimate the speed)

deltax=deltaL+vcar*(t3-t2)

The last term vcar*(t3-t2) represents the distance run by the car during the time required by the driver to look exactly on the car. So deltax gives estimation of the distance between the position of the car the instant reappears and the position of gaze of the driver in that same instant.

Finally, it is possible to calculate the estimated velocity as:

est vel=(50m+deltax)/(time during which the car disappears)

The time at denominator is calculated as

time=50m/(speed of the car in that particular test)

The distance considered is of 50m since this is the distance in which the car isn't visible for the driver.

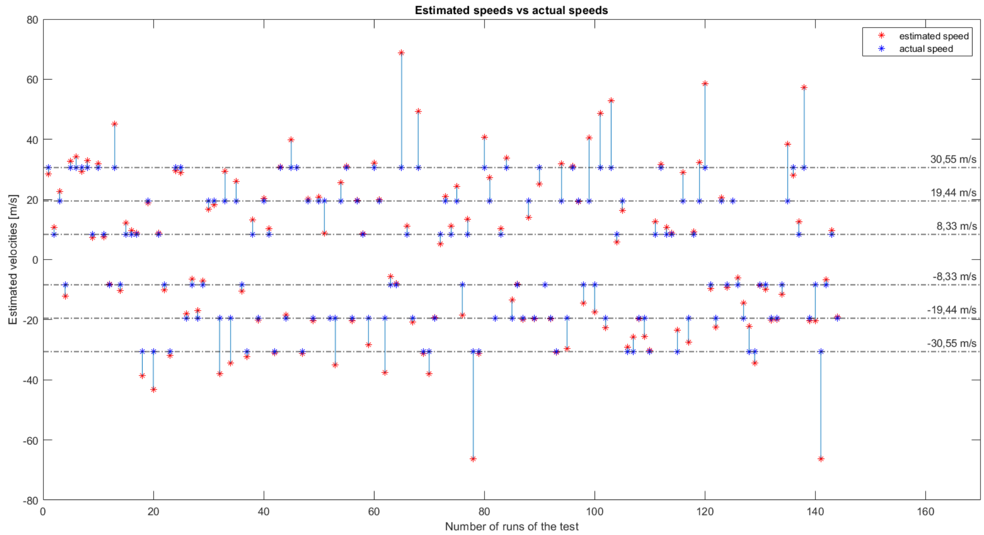


Figure 3.25: Estimated speeds vs actual speeds

In this graph it is reported the estimated speeds for all the runs of the test, which are 6 for each of the 28 participants. Some runs had to be eliminated due to problems of signal of the eye tracker. The dashed lines represents the 3 speeds of the test converted from km/h to m/s:

- 8,33 m/s corresponds to 30 km/h
- 19,44 m/s corresponds to 70 km/h
- 30,55 m/s corresponds to 110 km/h

It is possible to see these values with positive and negative sign, to represent the direction of motion: positive speed means that the cars were moving from left to right, while negative speed means that cars were moving from right to left. Red dots indicate the speed estimated by the driver for that particular run of the test, and the blue dot the actual speed of that test. So, if the red dot is close to the line, that estimation was accurate. If the dot is higher in module, it means that the speed was overestimated, if the dot is lower in module, it means that the speed was underestimated.

Now there are graphs showing the errors of this test. First there are 6 histograms indicating the errors for each of the 6 speeds, in m/s.

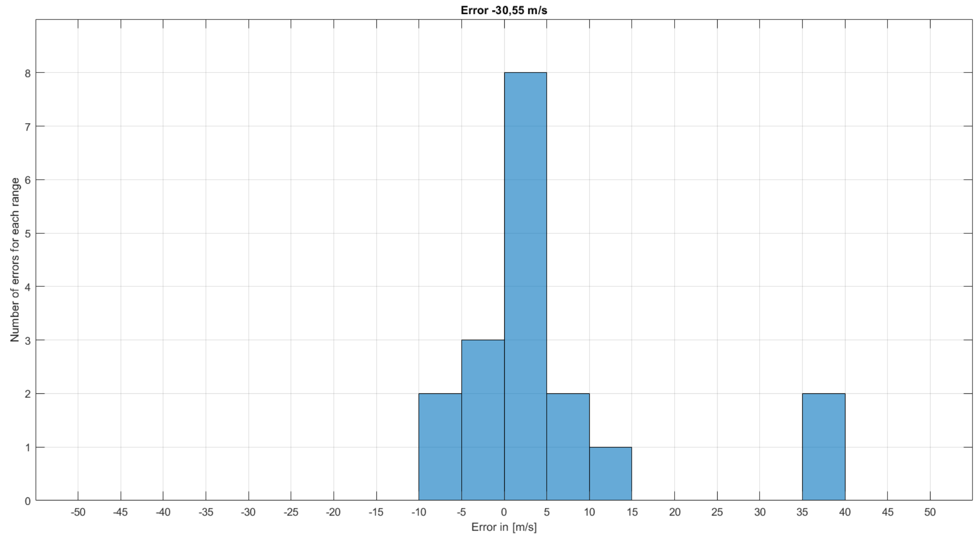


Figure 3.26: Histogram with errors for $-110km/h$

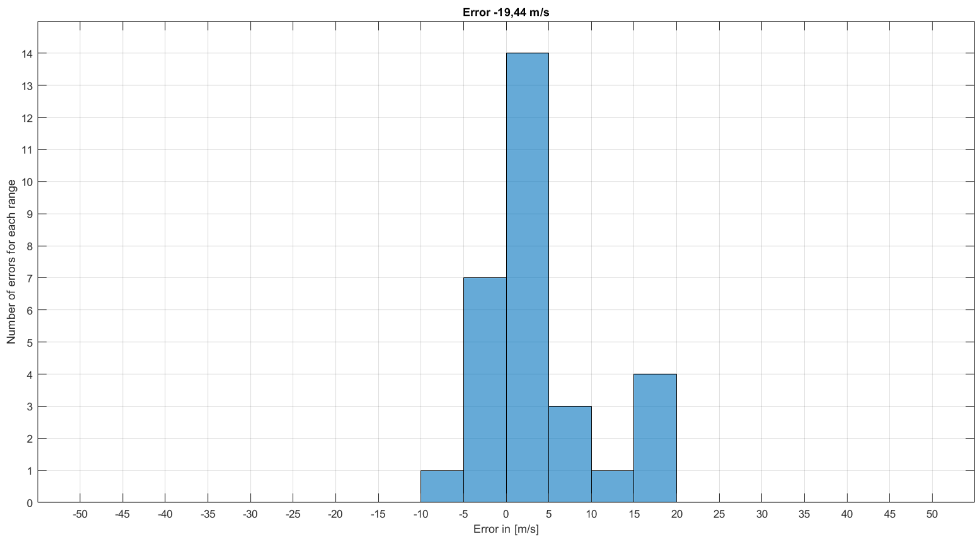


Figure 3.27: Histogram with errors for $-70km/h$

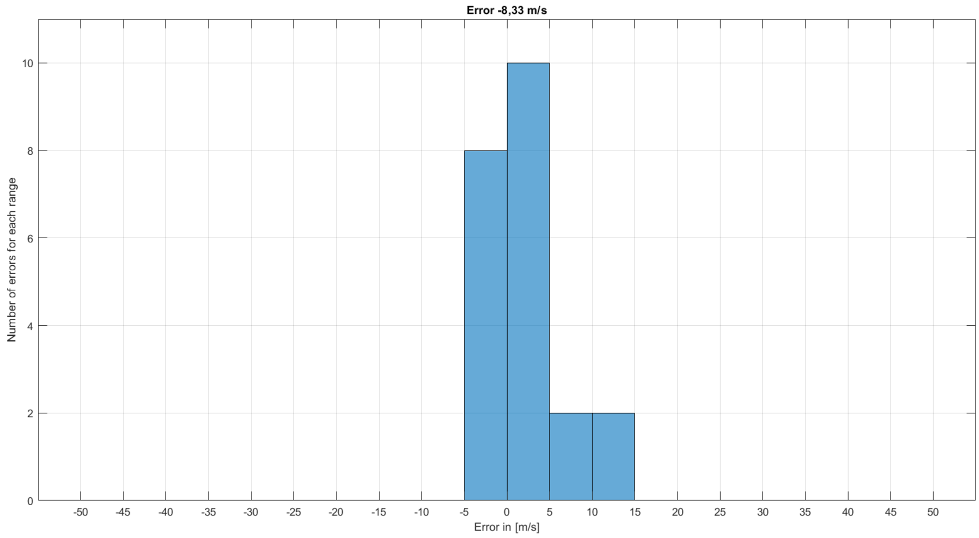


Figure 3.28: Histogram with errors for $-30km/h$

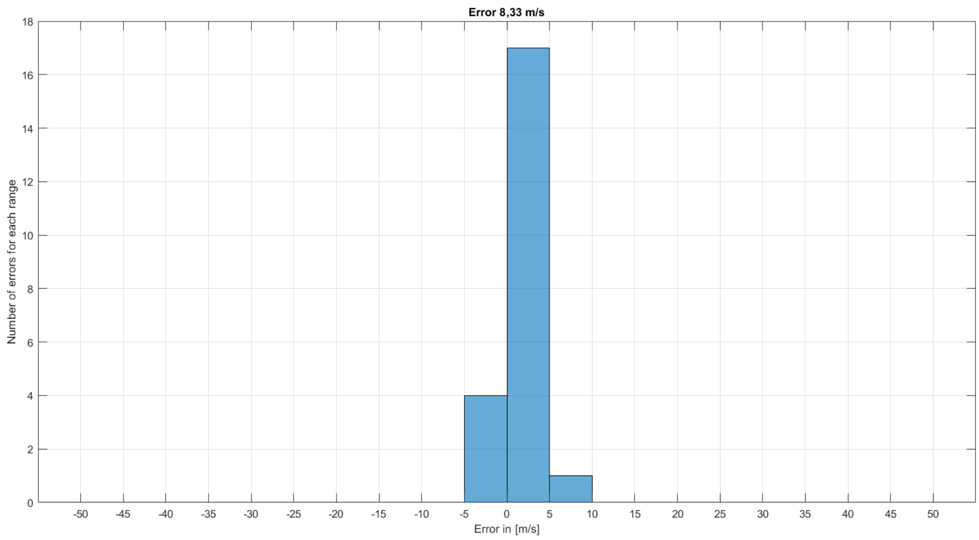


Figure 3.29: Histogram with errors for $+30km/h$

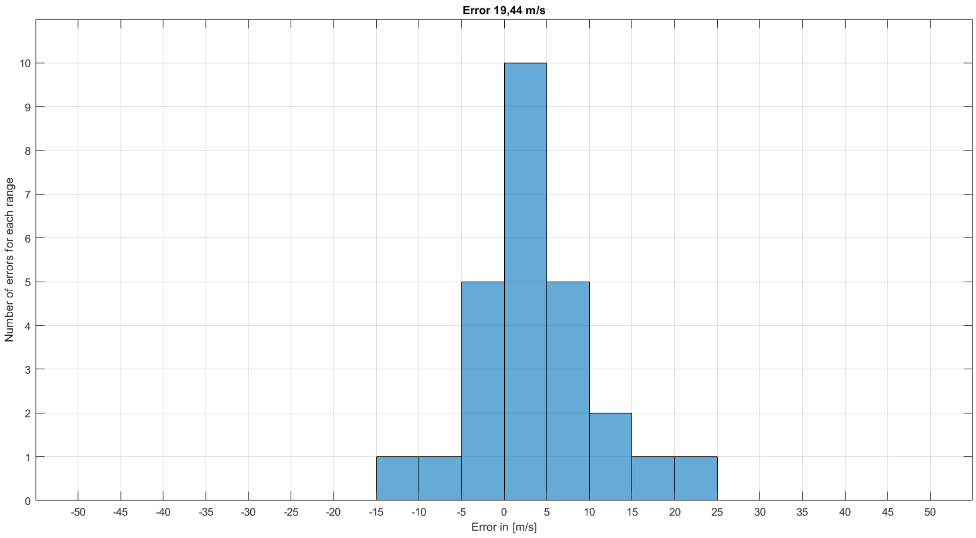


Figure 3.30: Histogram with errors for $+70km/h$

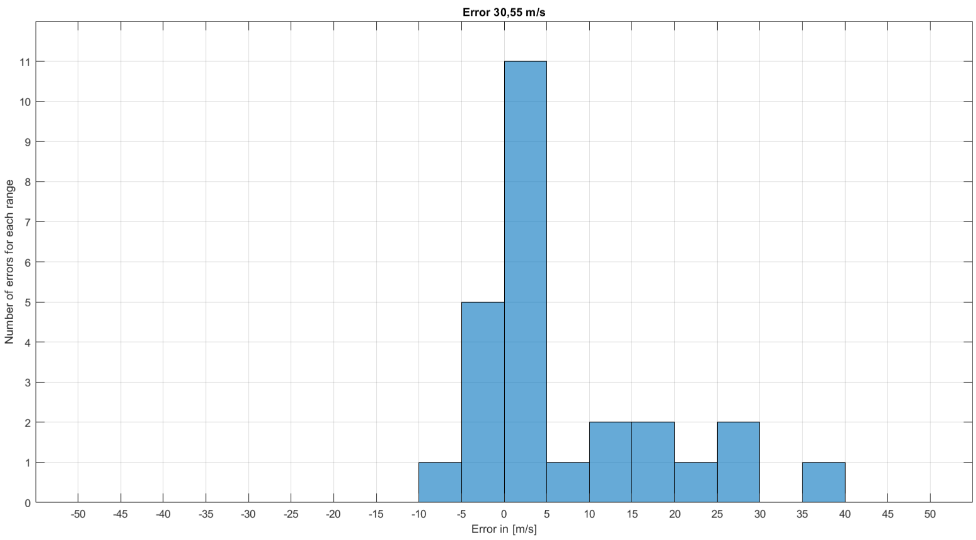


Figure 3.31: Histogram with errors for $+110km/h$

The graph for 8,33 m/s (30km/h) is the one which is concentrated more near the zero error value, while the graph for 30,55 m/s (110 km/h) is the one which is more dispersed around the zero value and shows higher error in magnitude.

Now there are the graphs showing the aggregated errors for the same value of speed with positive and negative sign, so 3 in total $(+-8,55,+-19,44,+-30,55)$.

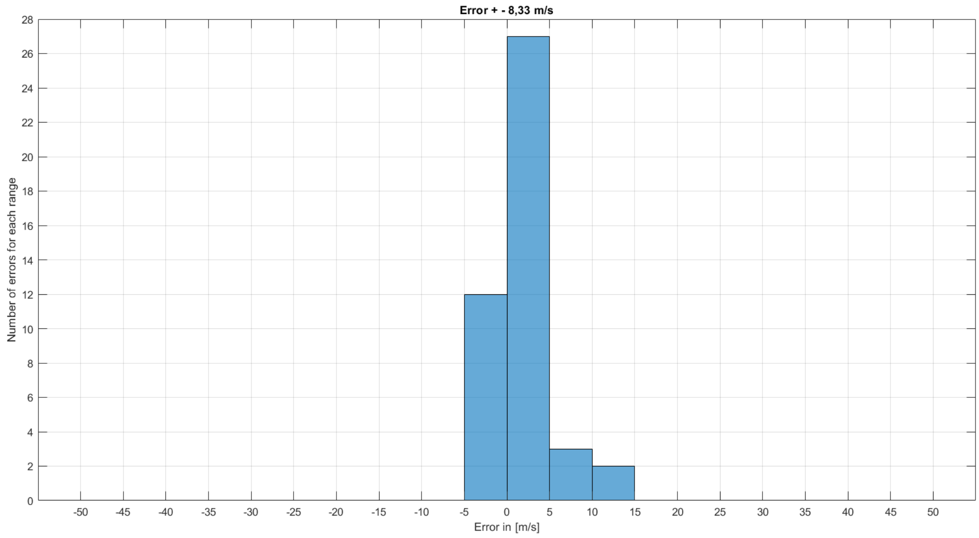


Figure 3.32: Histogram with errors for $+-30km/h$

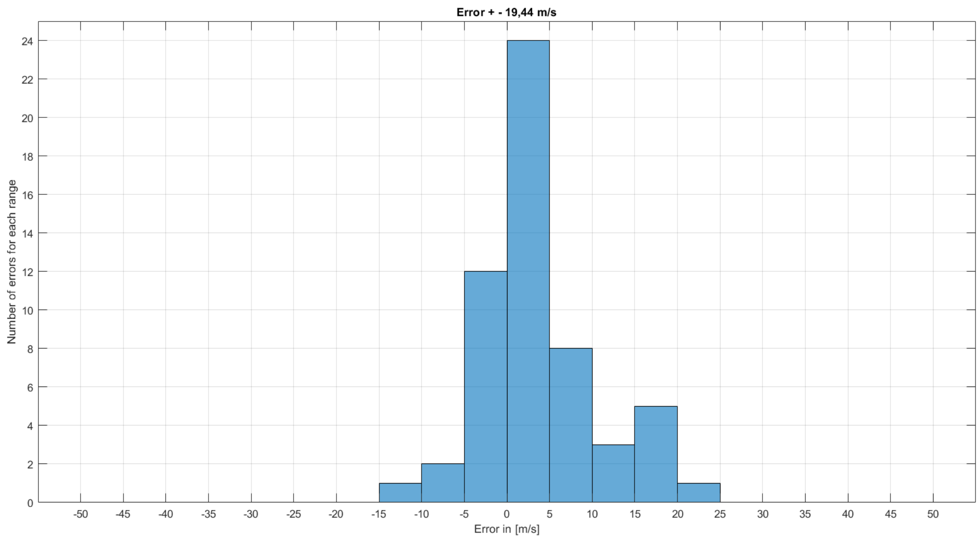


Figure 3.33: Histogram with errors for $+-70km/h$

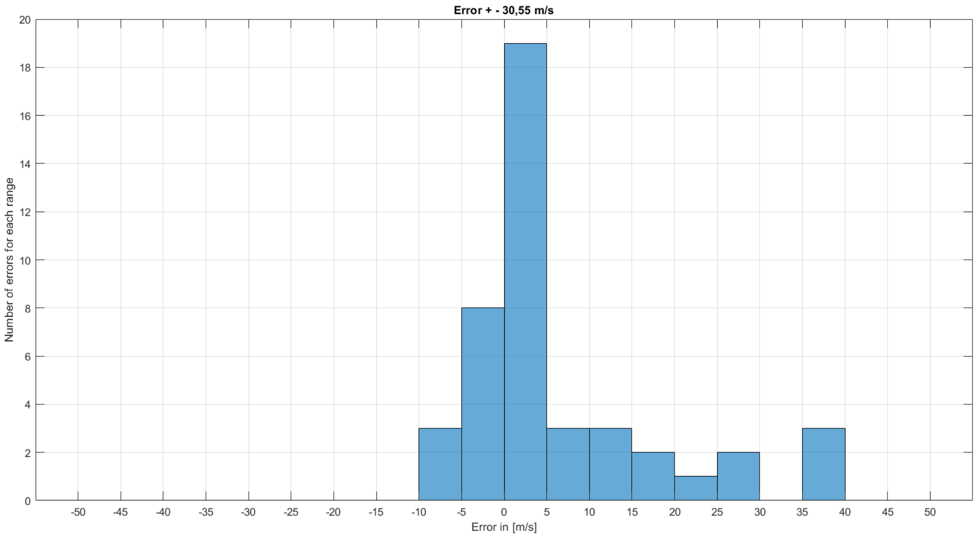


Figure 3.34: Histogram with errors for $+-110km/h$

It's possible to notice a trend: the higher the module of the speed, the higher the module of the error. Another interesting fact is that generally there is more probability to over-estimate the speed than to underestimate it, since the error are more translated towards positive values of the error rather than negative ones.

Now there are the histograms reporting the percentage errors, to put them in perspective with respect to the speed.

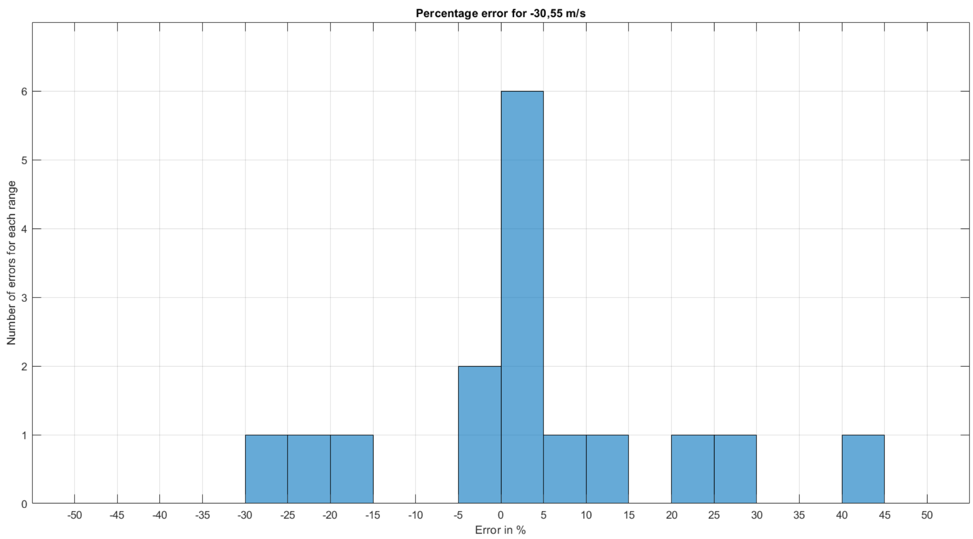


Figure 3.35: Histogram with errors for $-110km/h$ in percentage

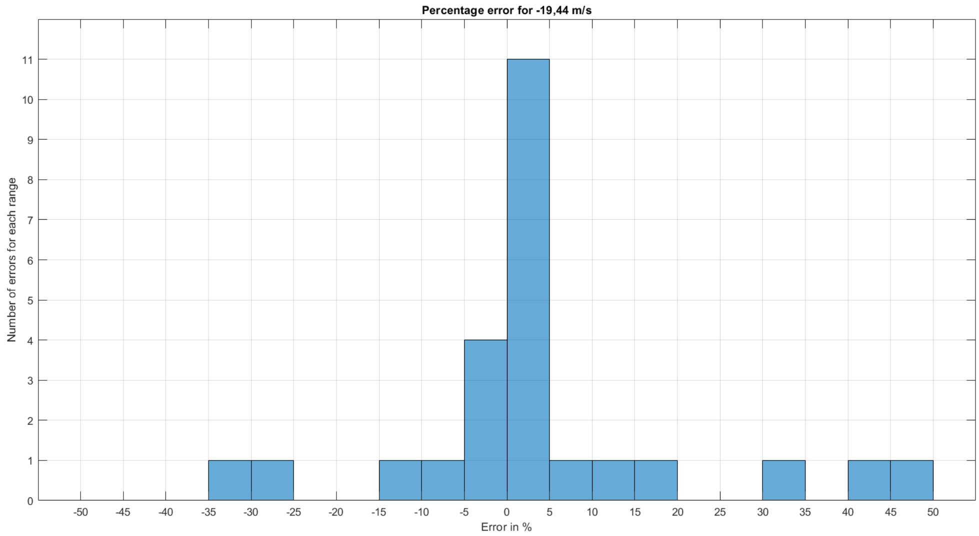


Figure 3.36: Histogram with errors for $-70km/h$ in percentage

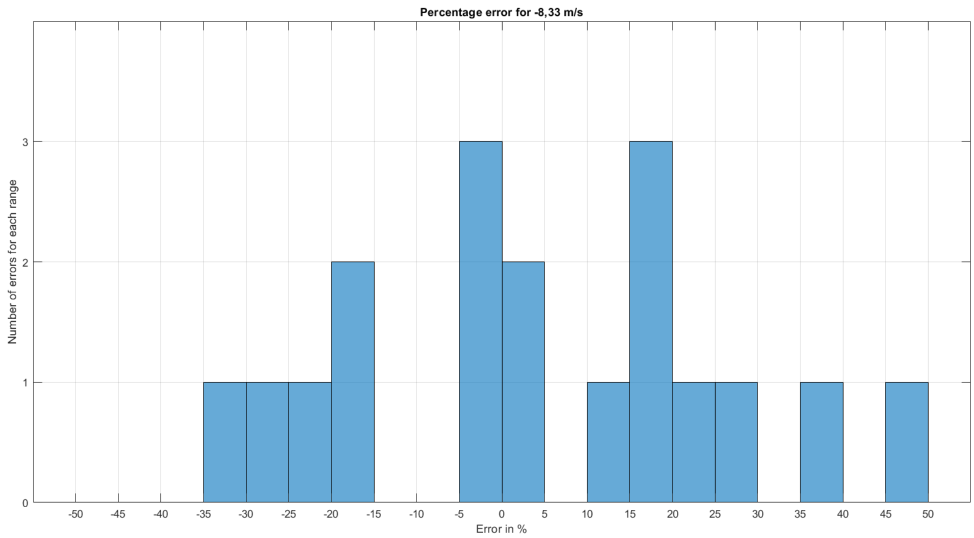


Figure 3.37: Histogram with errors for $-30km/h$ in percentage

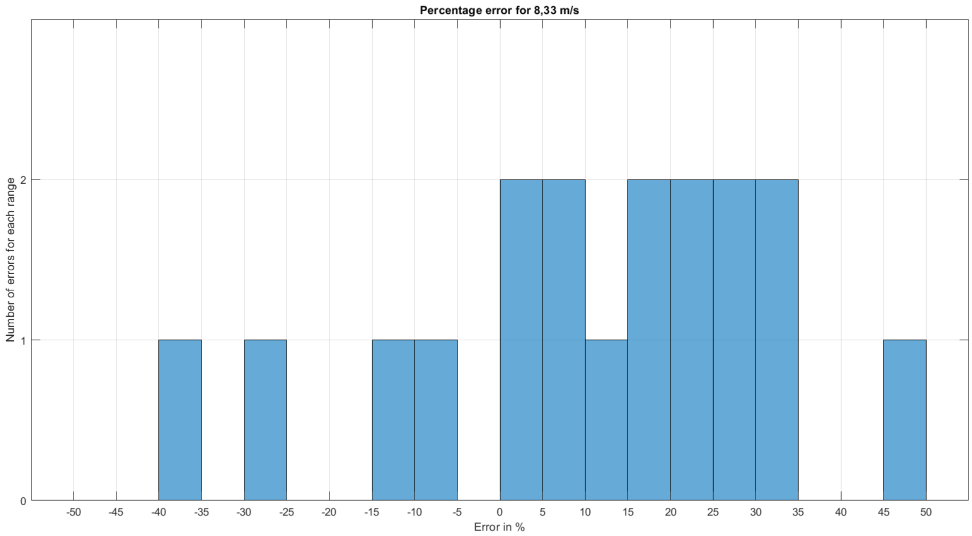


Figure 3.38: Histogram with errors for $+30 km/h$ in percentage

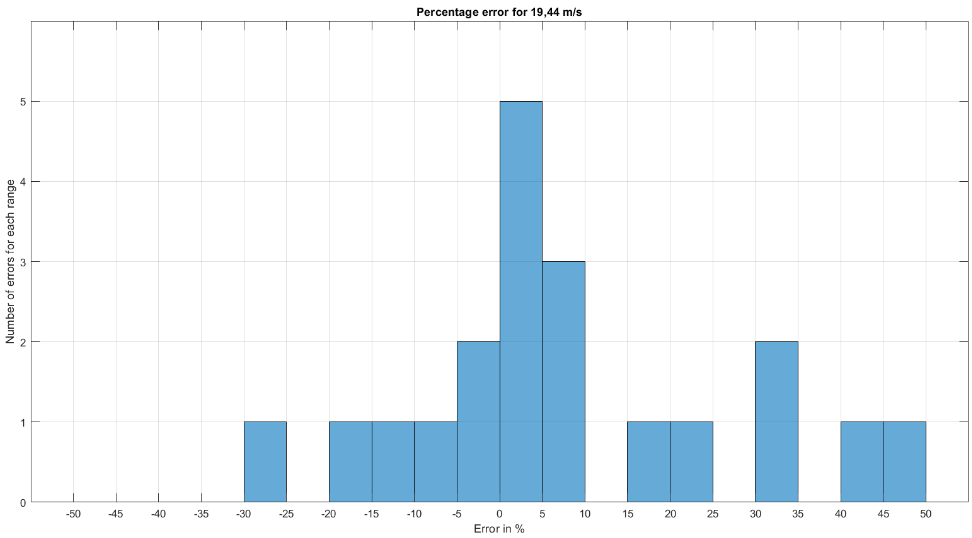


Figure 3.39: Histogram with errors for $+70 km/h$ in percentage

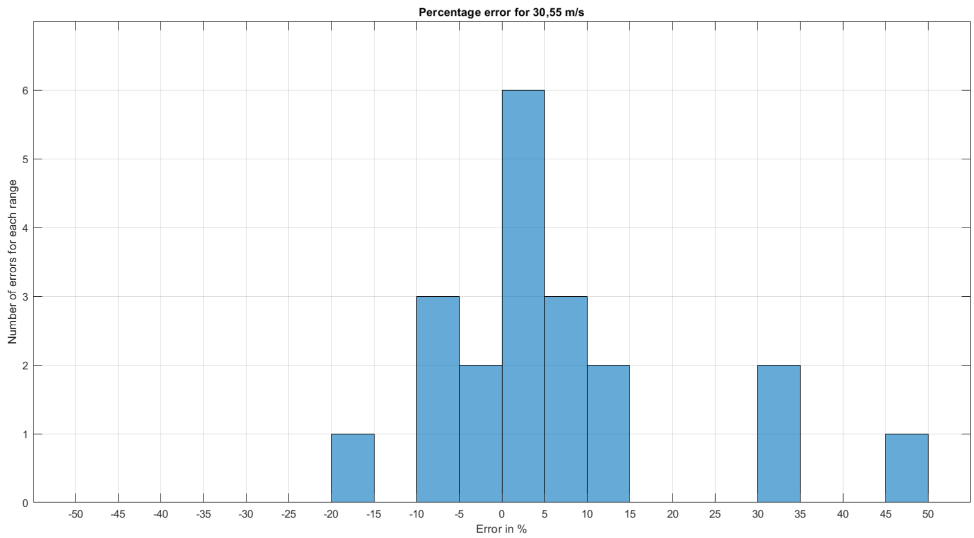


Figure 3.40: Histogram with errors for $+110km/h$ in percentage

Here we can see that just like before, the max error appear at positive error, which corresponds to overestimation. Generally the error is limited to 50% of the value of the speed.

3.9. Second test: straightroad scenario

In this test it was used the AOI tool from 'Tobii Pro Lab', to overlap areas of interest on the video of the scene. In this case it was selected areas for the roof, central portion of the road, dashboard, left and right windows, left and right mirrors. Once all videos were overlapped, they were exported in excel sheets. In these files are reported, for each instant of time, in which area the driver was looking. Then, with a Matlab script, it was possible to create a table, for each run of the test, indicating the order of the areas in which the driver looked and for how long. This scripts takes as input each excel sheet, removes intervals of time smaller than a constant tau=0,1s to delete possible undesired fluctuations of the gaze, before proceeding with the creation of the order of gaze.

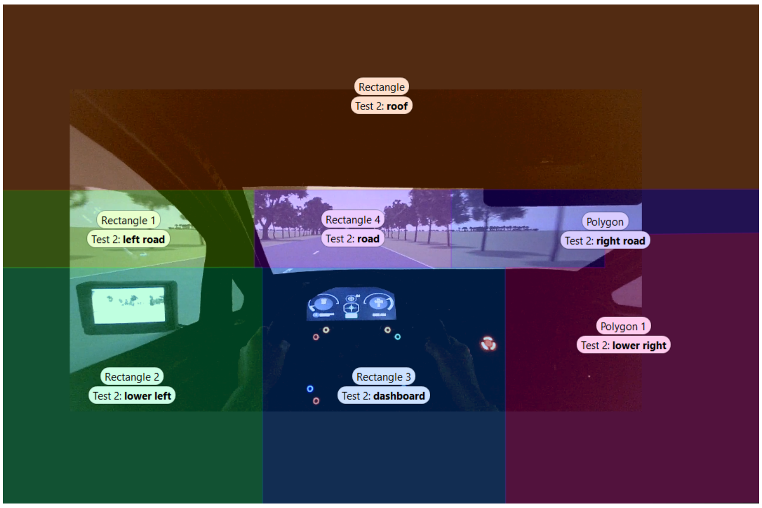


Figure 3.41: This is the mask applied with the Areas Of Interest

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	'road'	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'
2	6.9910	7.1310	8.6340	9.0740	11.3790	14.3040	16.5880	18.2310	18.9120	20.6150	21.4970	28.8500	29.9120
3	7.1110	8.5530	9.0540	11.2980	14.2840	16.5680	18.2110	18.8920	20.5950	21.4770	28.8300	29.8920	37.2050
4	0.1200	1.4220	0.4200	2.2240	2.9050	2.2640	1.6230	0.6610	1.6830	0.8620	7.3330	1.0420	7.2930
5	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
6	'road'	'left road'	'right road'	'lower left'	'lower right'	'dashboard'	'roof'	'area'	[]	[]	[]	[]	[]
7	21.3770	0	0	0	0	8.4750	0	'tot time [s]'	[]	[]	[]	[]	[]
8	7	0	0	0	0	6	0	'n° fixation'	[]	[]	[]	[]	[]
9	3.0539	0	0	0	0	1.4125	0	'mean time...	[]	[]	[]	[]	[]

Figure 3.42: This is the result obtained for the 1st sub-case

Here it is reported an example of the data obtained from this test: a series indicating the order of the areas the driver looked, starting time, finishing time and duration of that interval of time. Then, from line 6, it is reported the total time spent in each area, number of times the driver looked there and average time of each fixation.

It is interesting to notice how this table may change depending on the scenario considered: in this case, the table reported in the picture refers to the first case, where the driver has to drive on a straight trajectory, with no other cars present on the scenario. It is possible to see that the driver only looks on the road in front and to the dashboard, to keep under control the speed of the car. All the other drivers had the same behaviour: no one looked outside the lateral window since there was nothing important to look at.

	1	2	3	4	5	6	7	8	9	10
1	'road'	'lower left'	'road'	'dashboard'	'road'	'dashboard'	'road'	[]	[]	[]
2	3.1960	10.6290	11.0700	12.4720	13.0140	20.7670	21.2080	[]	[]	[]
3	10.5890	10.9900	12.4520	12.9930	20.7480	21.1880	21.9500	[]	[]	[]
4	7.3930	0.3610	1.3820	0.5210	7.7340	0.4210	0.7420	[]	[]	[]
5	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
6	'road'	'left road'	'right road'	'lower left'	'lower right'	'dashboard'	'roof'	'area'	[]	'side [s]'
7	17.2510	0	0	0.3610	0	0.9420	0	'tot time [s]'	[]	10.1320
8	4	0	0	1	0	2	0	'n° fixation'	[]	[]
9	4.3128	0	0	0.3610	0	0.4710	0	'mean time...	[]	[]

Figure 3.43: This is the result obtained for the 5th sub-case

Figure 3.43 reports another table showing the 5th case, where there is a parked car and a moving car overtaking it. In this case, some drivers look on the left mirror ('lower left'), some look on the left window to check the moving car and the parked one. Whatever is the case, generally the side mirror or the side window are looked just one time, and then the gaze goes back to the road in front of the car.

3.10. Third test: Crossroad scenario

For this test, the procedure is identical to the one previously described for the scenario straightroad. The only difference regards the AOI (areas of interest): in this case there are the roof, the central portion of the road, left part of the road (what is visible from the left window), right part of the road (what is visible from the right window and the side part of the road visible from the windshield) and the dashboard. Side mirrors were not considered in this case since they were never used during these maneuvers by drivers.

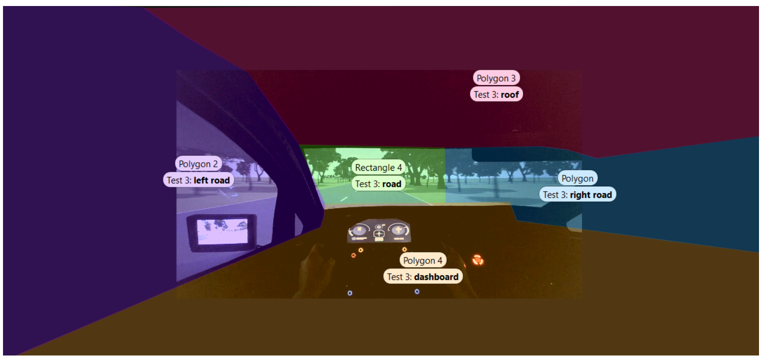


Figure 3.44: This is the mask applied with the Areas Of Interest

	1	2	3	4	5	6	7	8	9	10
1	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'	'dashboard'	'road'
2	2.5420	2.8030	3.6640	4.0450	5.3470	5.6880	6.9900	7.3910	9.5750	9.8760
3	2.7830	3.6440	4.0250	5.3270	5.6680	6.9700	7.3710	9.5550	9.8560	10.8370
4	0.2410	0.8410	0.3610	1.2820	0.3210	1.2820	0.3810	2.1640	0.2810	0.9610
5	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
6	'road'	'left road'	'right road'	'dashboard'	'roof'	[]	[]	'area'	[]	'stop [s]'
7	15.7450	4.6280	2.4240	4.1500	0	[]	[]	'tot time [s]'	[]	13.7330
8	17	5	5	11	0	[]	[]	'n° fixation'	[]	[]
9	0.9262	0.9256	0.4848	0.3773	0	[]	[]	'mean time...	[]	[]

Figure 3.45: This is the result obtained for the 2nd sub-case

This is the table with one result obtained in this test. There's a clear difference the previous one. In this case, much more time is spent looking laterally, since drivers must check that the road is clear before entering the intersection. Also, the number of fixations to the left and right is higher, since drivers check more times before and after the moving car passes in front of the ego car. In general, results for this test show a much longer list of different areas where the sight was concentrated. Indeed, in the previous test, very often happens that the driver only looks on the road ahead and to the dashboard, making the list much shorter.

4 | Model of perception

This chapter will present the development of a model of perception, exploiting the results obtained from the tests at the simulator. In fact, these tests were conducted to study the behaviour of drivers and characterise them through numbers, with scientific approach. Thanks to the data collected and elaborated, it was possible to obtain linear regressions that describe the error to estimate the distance and the angle of a car. These are the starting point to create the model.

First, the model is created in Simulink. The co-simulation is conducted together with MATLAB, Simulink and WorldSim. Similarly to the scenarios launched during the tests, the MATLAB script loads the scenario settings and other variables, then it launches the Simulink model and WorldSim. The Simulink script receives as input the position of the ego and traffic car, and applies a bias of distance and orientation to the position of the traffic car, according to the results calculated from tests. Then the information is sent to WorldSim that proceeds with the real time simulation of the motion of the cars.

4.0.1. The bias of distance and angle

This part is the inner core of the model of perception, where the actual calculation happens. Inside the Simulink model there are 2 blocks called ‘state manager’from the Worldsim library, that receive the position of the ego and traffic car. These information are sent to a MATLAB function block. Here it is chosen which angle of the field of view to use. For this target, they are used values obtained from the first test of the car approaching from the side.

Depending on the speed of the vehicle (30, 70 or 110 km/h) and age of the driver (under 30 or over 30) it is selected the proper angle. The field of view of the driver is modelled as a isosceles triangle having the vertex in the centre of the car. The sides are inclined with respect to the longitudinal distance of the car with the angle just described. The last side of the field of view is in transversal direction, with distance from the ego car calculated as v*2,5; v is the speed of the ego car, and 2,5 represents the look ahead distance, measure used to indicate how further the driver is looking.

The first thing that the Simulink block does, is to calculate if the traffic car is inside or outside of the field of view, exploiting the coordinates of the two cars. If the car is inside, the bias are calculated, otherwise nothing changes. This is how the bias are created.

First, from the list of errors done for each longitudinal distance, they are put together in 'bins' of 20m each, from the closest to the furthest. Then for each group, it is calculated the median, 25th percentile, 75th percentile, the maximum and the minimum values estimated.

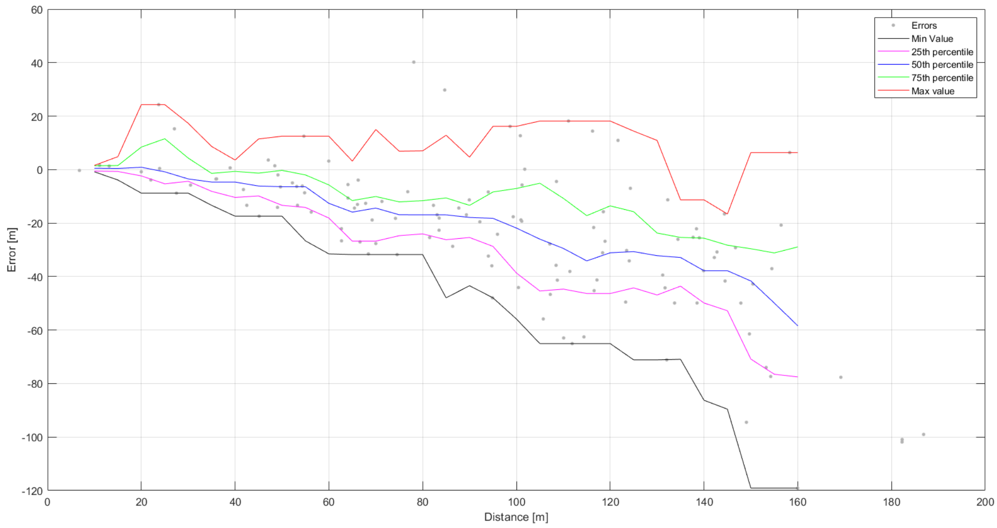


Figure 4.1: The percentiles of the distance

Then linear interpolations are calculated, one for each of these 5 lines. All these interpolations pass through the origin, to represent the fact that when the car is closer, the error reduces.

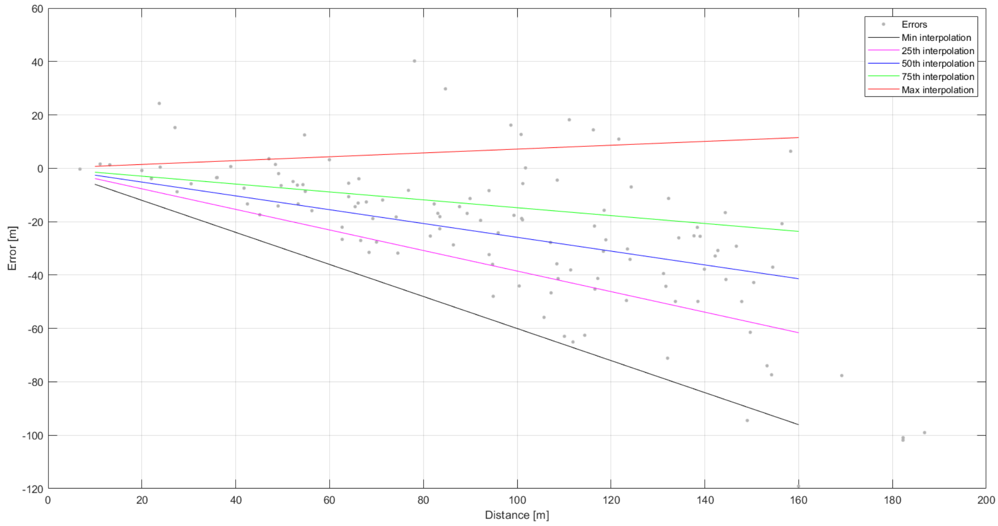


Figure 4.2: The interpolations of the distance

These are the angular coefficients:

- m = -0,6003
- m = -0,3850
- m = -0,2586
- m = -0,1478
- m = +0,0719

To start the simulation, a random number between 0 and 1 is selected. This number decides which kind of perception will affect the actual position of the traffic car. If the number is 0,5, the behaviour will be the average observed in the tests. If the number is lower, the behaviour will underestimate the distance even more. If the number is higher than 0,5 the behaviour may even overestimate the distance of the traffic car.

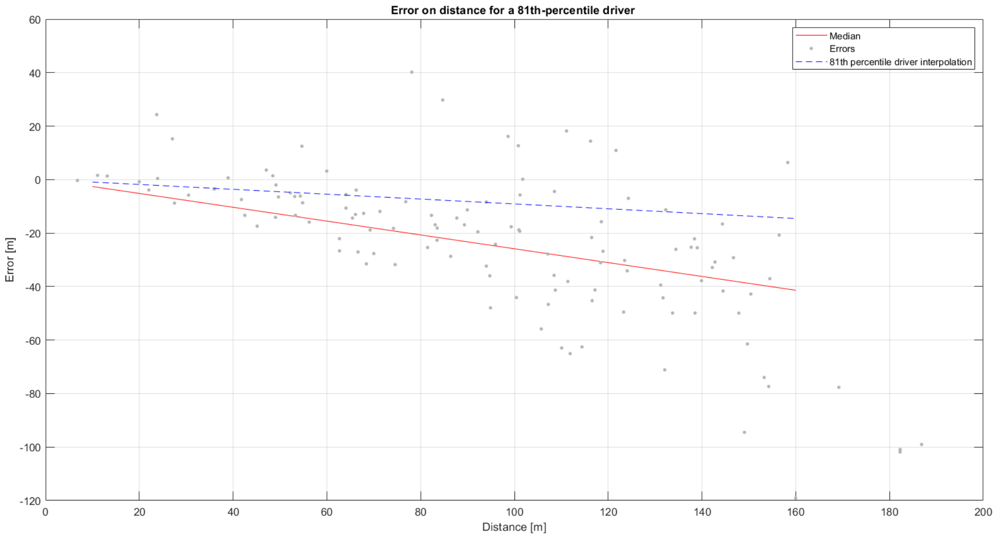


Figure 4.3: An example of percentile

In this graph the blue dashed line represents the 81th percentile driver, according to the errors collected during the tests.

The same procedure is done for the angles, diving them in left and right, since the distribution of data is slightly different from left to right.

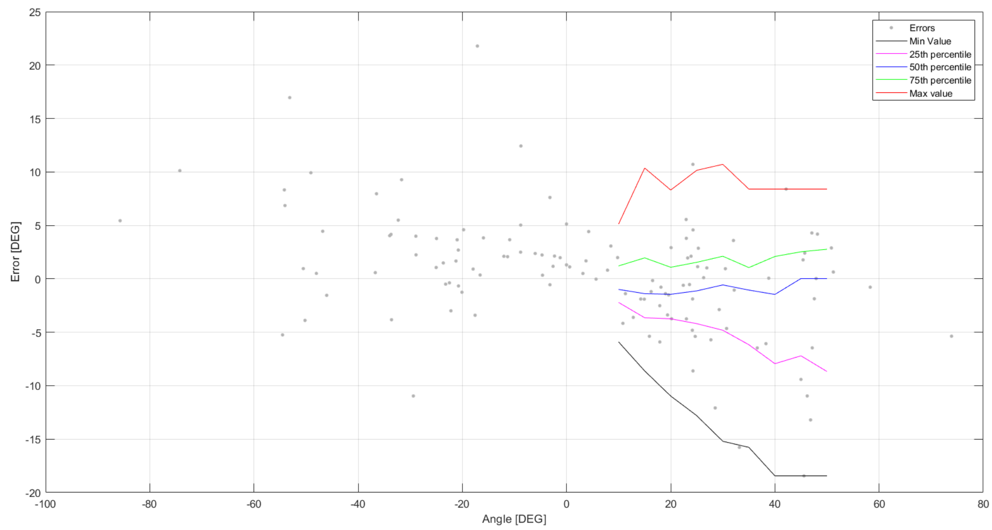


Figure 4.4: These are the percentiles for angles to the right

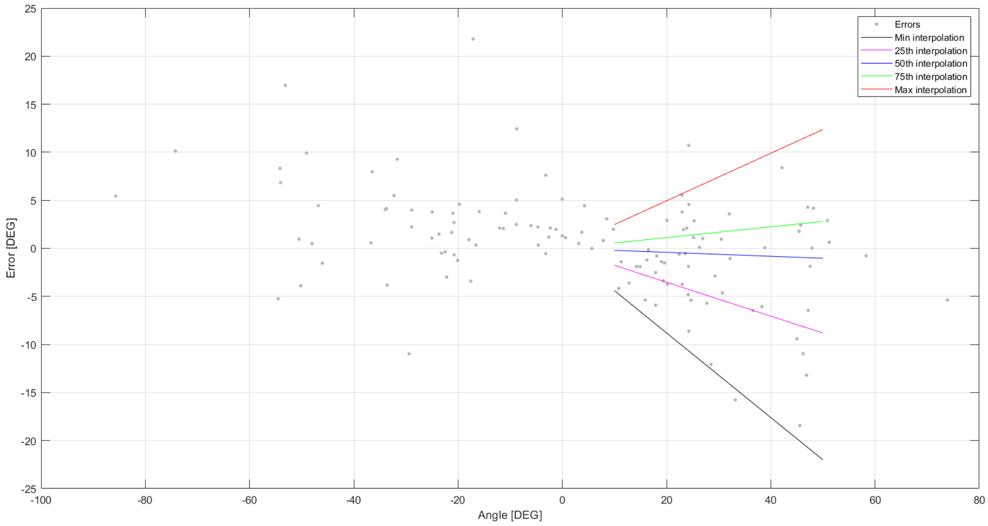


Figure 4.5: These are the interpolations for angles to the right

These are the angular coefficients:

- m = -0,4401
- m = -0,1763
- m = -0,0206
- m = +0,0560
- m = +0,2474

These are the linear interpolations. The red line represents the median. It is slightly tilted downward, meaning that the average tendency is to underestimate the angulation of the car wit h respect to the longitudinal distance of the car. As for the distance, a random number lower than 0,5 will result in an even higher underestimating behaviour, while a number higher than 0,5 may lead to overestimate the angulation of the traffic car.

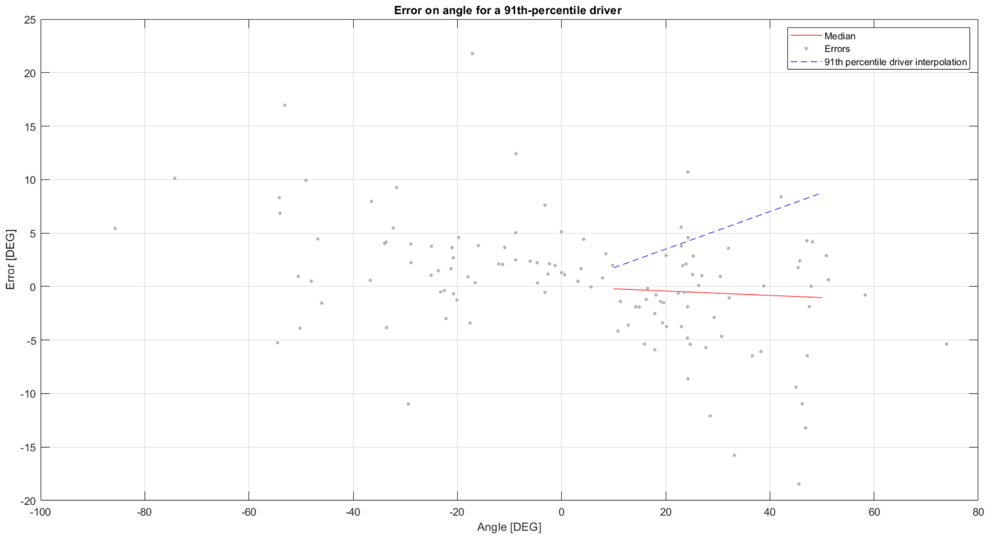


Figure 4.6: Example for angles to the right

This is an example for a 13th percentile driver. In this case, the underestimation is accentuated with respect to the average.

The same considerations are done for the angles to the left.

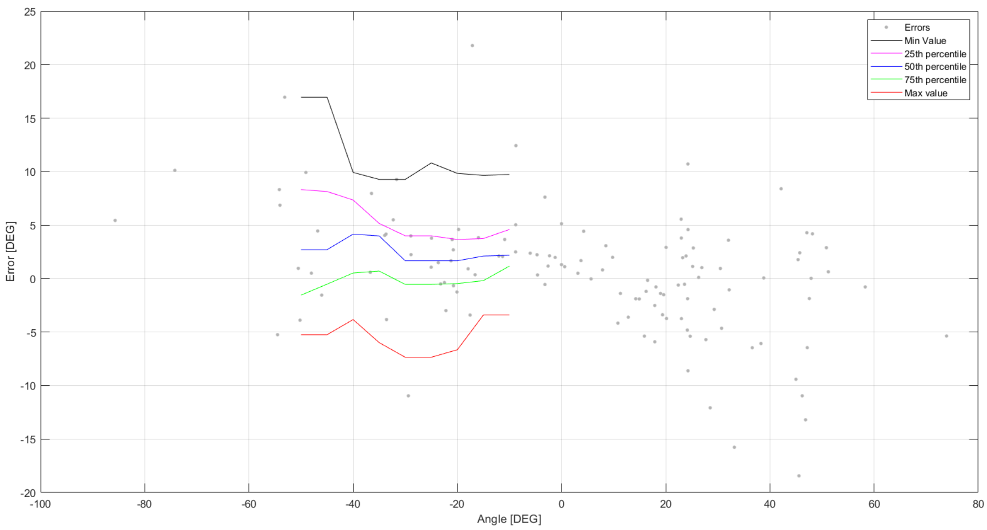


Figure 4.7: These are the percentiles for angles to the left

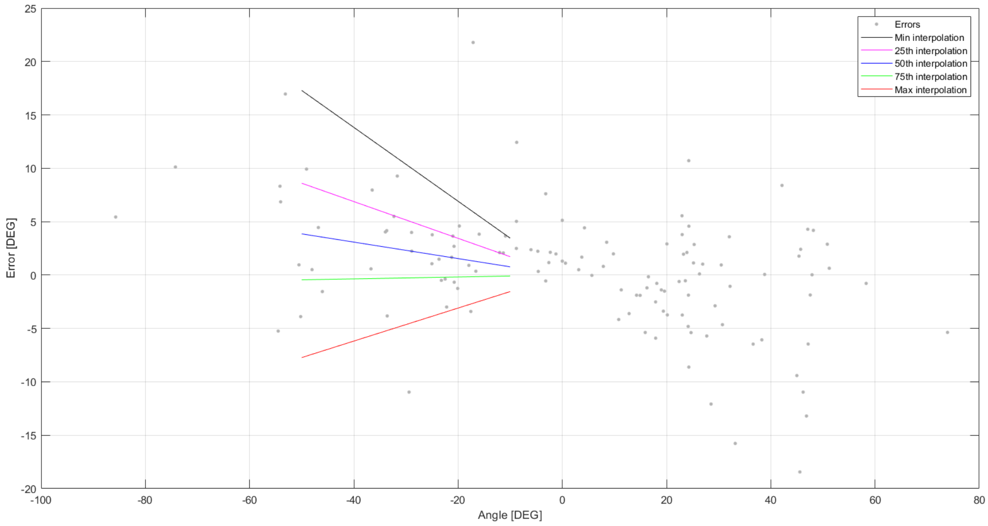


Figure 4.8: These are the interpolations for angles to the left

These are the angular coefficients:

- m = -0,3458
- m = -0,1720
- m = -0,0772
- m = +0,0090
- m = +0,1546

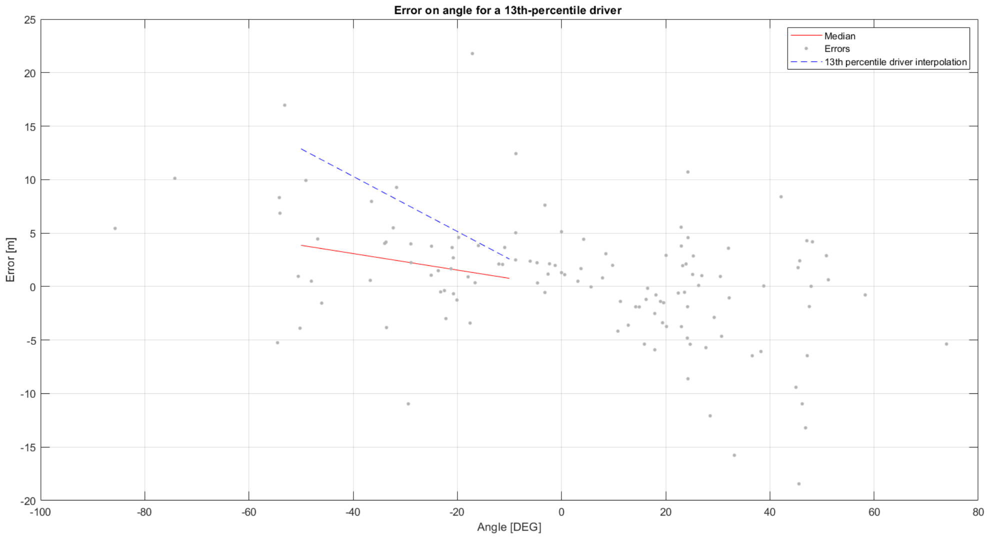


Figure 4.9: Example for angles to the left

This is an example for a 91th percentile driver.

At the start of the simulation a random number is selected. Based on its value, a linear interpolation is created between the 5 available. The random number is the same for the distance and the angle. Since all interpolations pass through the origin, these lines are characterised by just the angular coefficient. These coefficients are stored in the workspace and taken to the Simulink ambient. Here the bias of distance and angle is calculated multiplying the angular coefficient and the effective distance ego-traffic car. The same goes for the angle.

Bias dist = mDist*effective distance

Bias angle = mAngle* angle between ego and traffic car

Then the bias on x and y directions are calculated as:

xbias=bias dist*cos(angle+bias angle)

ybias=bias dist*sen(angle+bias angle)

Finally these values are summed to the prescribed trajectory of the traffic car.

4.1. Implementation of the model of perception

The model of perception is now applied to a scenario to verify that it operates properly. The scenario is a section of an highway, with divided directions of traffic and 4 lanes for each. The ego car starts on the fourth lane from the right, while the traffic car is in the second lane. The ego car starts behind the traffic and travels at higher speed, so that the two cars get progressively close.

Figure 4.10: This is the start of the test seen from the ego car

Figure 4.11: This is the start of the test seen from the top

For this test, it was selected the 50th percentile, meaning that the model will behave like the average driver. This means that the model will underestimate the distance of the car, and underestimate slightly the inclination of it. The two cars start with a longitudinal distance of 81,59m, and a lateral distance of 7,3m. Figure 4.12 shows the positions of the cars after 10s from the start.

Figure 4.12: This is the visual from the top of the results

Figure 4.13: This is the perspective from the ego car of the results

The black car represents the traffic car with the imposed model of perception, while the white car represents the traffic car unbiased. The results are consistent with the model for 2 reasons:

1. The black car is closer compared to the real position; in fact the 50th percentile underestimates the distance of the car.
2. The black car is closer to the longitudinal direction compared to the real case; the change is so small that can be verified by the coordinates and not by the scene. In fact in the tests drivers perceived the angles much more precisely than the distance. The biased car is positioned, with respect to the ego, 25,2m in longitudinal direction and 5,7m to the right. So the car is inclined with an angle of 12,94°. The unbiased car is positioned, with respect to the ego, 31,7m in longitudinal direction and 7,3 m to the right. So the car is inclined with an angle of 12,99°.

It is possible to assert the consistency of the model also from the plot of the bias applied during the simulation.

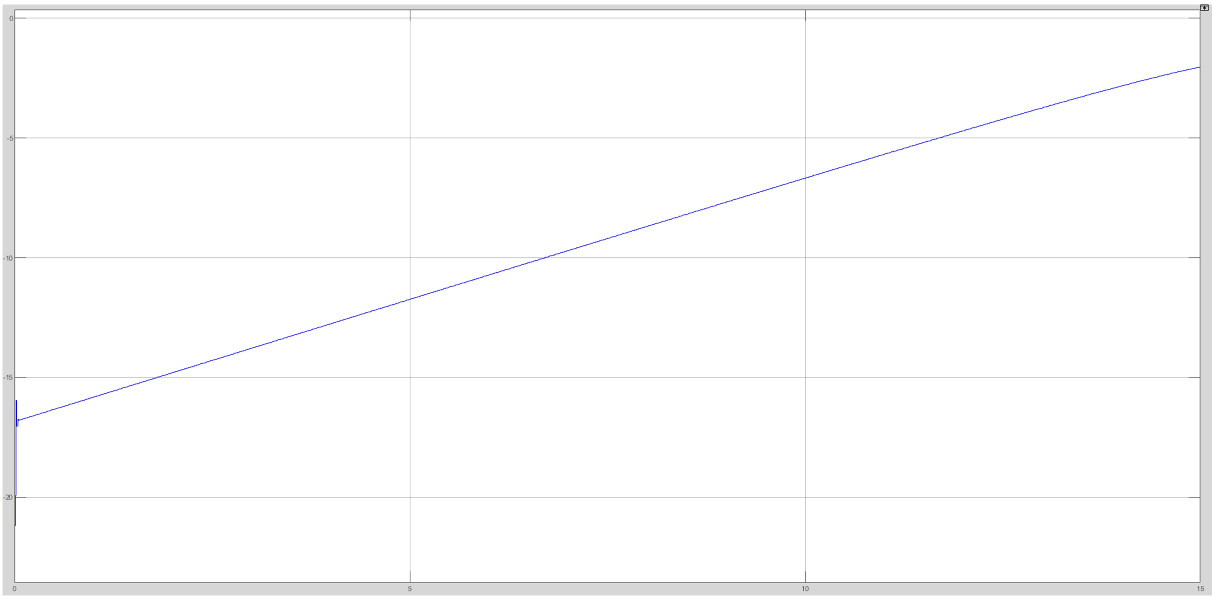


Figure 4.14: Bias applied on the distance

On the x axis there is time and on the y axis there is the bias applied to the distance. Since the model underestimates the distance, the bias is negative, and its value reduces when the two car get closer, because at lower distances the errors decrease.

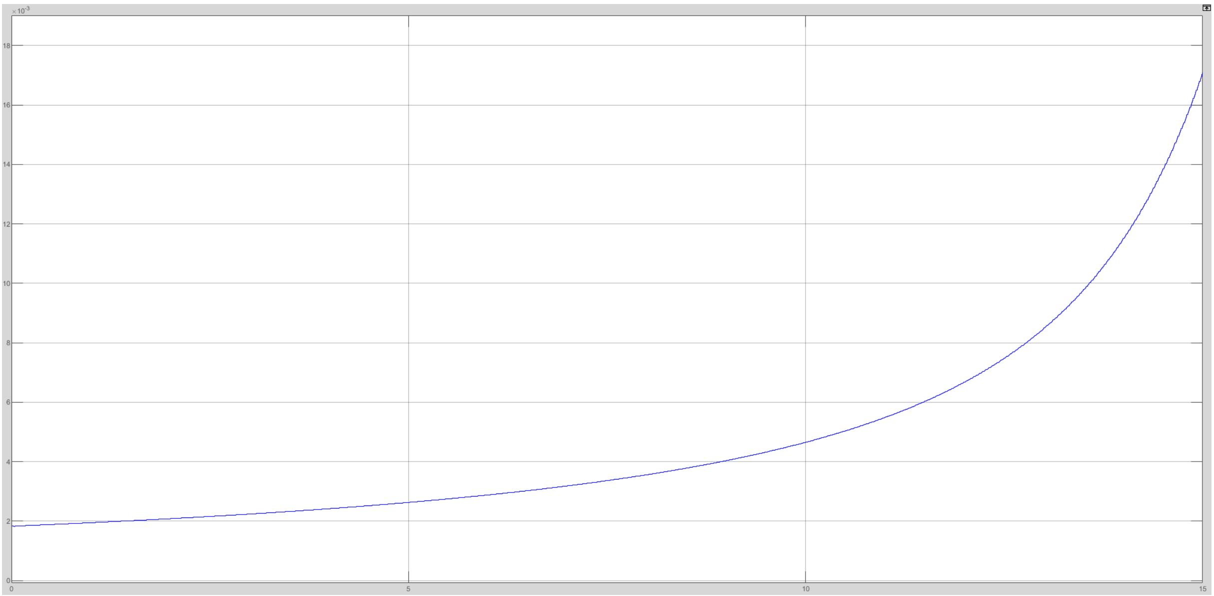


Figure 4.15: Bias applied on the angle

On the x axis there is time and on the y axis there is the bias applied to the angle. The value of the bias increases in time because when the ego car gets closer to the traffic, this one appears more to the side, at an higher angle. So an higher angle of positioning means an higher error of perception of the real angle. The value is positive because the model applies a positive rotation (counterclockwise) to the traffic car, in order to make it appear closer to the longitudinal direction. This happens because, as just said before, this particular model underestimates distance and angle of positioning of the car.

5 | Conclusions and future developments

This thesis was conducted with the target to study and characterise the visual perception of drivers. This is required because with the study of the behaviour of human drivers, it is possible to create “human-like” autonomous vehicles. This is of crucial importance because people may accept more easily ADAS (Advanced Driver Assistance Systems) when these systems behave similarly to them. Another advantage is that with a model of a human driver, it is possible to create ambient of simulation in which autonomous vehicles interact with vehicles that behave similarly to the real ones.

The first part of this work was devoted to the creation of the scenarios used during the tests at the simulator. The result obtained is a set of 6 scenarios created each one with the aim to measure one different property of the driving behaviour of drivers that took part to the tests. To reach this goal it was necessary to have some “traffic cars” that would move inside the scenarios in a precise manner. This goal has been reached thanks to many attempts of simulation of the tests.

The second and third parts were dedicated to explain how tests were conducted and what data was collected. The target was to get information about the behaviour of drivers and extrapolate some parameters to be used as a starting point for the creation of a “human-like” autonomous driver.

The fourth part was related to the creation of a model of perception of drivers. In this case the target was to put together all the information elaborated to create a model capable to describe the perception of vehicles around, just like an average person would do. Once the model was created, the next step was to run it on Simulink to show visually the results and the ability of this model to do what was expected to do.

After reviewing the results obtained from this thesis it is possible to draw some conclusions regarding the behaviour of drivers. First, drivers generally tend to believe that a car is closer than it really is, with errors that may get quite high up to 50%. Regarding the angular position, drivers usually perceive it smaller than it really is, and the errors are generally quite low. Distance has more influence than speed to the perception of a vehicle, as reported by drivers. Regarding the speed of a moving vehicle, generally drivers show a good ability with a slight tendency to overestimate it, in contrast with what was obtained before. Taking a look at the model of perception, it can effectively change the position of cars around to show how real drivers perceive a vehicle in the surrounding area. As in the example shown in the last chapter, it makes the traffic car appear closer than it really is, putting into evidence the underestimation of distance of drivers. The same logic goes for the angle at which cars appear; in fact, the model changes the position so that the angle appears lower than it really is. The magnitude of the change on the angle is much lower compared to the change in distance, since it was found out in this thesis that drivers estimate the angle much better than the distance.

Another result obtained is related to the methodology developed during this thesis to deal with the eye-tracker. First of all, thanks to LAN cables present inside the cockpit of the simulator, it was possible to establish a direct linkage from the control room PC to the eye tracker. After some setup it was possible to receive as input the status of the eye tracker. This gave possibility to automatically start the registration of each run of the test from the pc, as soon as the driver pressed the start button on the cockpit, instead of doing it manually, thus improving efficiency. Regarding the data analysis, it was necessary to create custom masks to analyse the case of the intersection and the straight road, and refining it a few times to perfectly overlay on the different areas present inside the cockpit. Lastly, the most manual and slow part was the one to analyse the perception of speed by the drivers, which couldn't be done automatically. It was necessary to put a marker to indicate the different instants of time for all the runs of the test. But then, once the data was exported in excel sheets (same procedure done for all tests done with the eye tracker) it was much easier to import it onto MATLAB tables and proceed with the calculation of the estimated speeds.

Regarding the possible future developments of this work, it is hoped that the study conducted in the second test with the eye tracker will be a starting point to conduct more tests in the field of autonomous driving vehicles. By knowing the possibility that a driver is looking in a certain direction, ADAS can be set to work together with vehicles driven by humans, calculating the best action to take considering the nature of the attention of drivers. Also the model of perception can be a fundamental tool to create virtual scenarios in which autonomous vehicles interact with vehicles that behave just like the real ones.

Bibliography

[1] A. R. W. Huang and Chihsiuh Chen. A low-cost driving simulator for full vehicle dynamics simulation. *IEEE Transactions on Vehicular Technology*, vol. 52(no. 1):pp. 162–172, January 2003.

[2] A. Amouri H. Arioui S. Espie L. Nehaoua, H. Mohellebi and A. Kheddar. Design and control of a small-clearance driving simulator. *IEEE Transactions on Vehicular Technology*, vol. 57(no. 2):pp. 736–746, March 2008.

[3] H.J. Shin Y.T. Son S.W. Kim M.W. Suh T.Y. Koo, B.Y. Kim. Development of the sungkyunkwan university driving simulator (skud) for human-machine interface studies of car navigation systems. *International Journal of Automotive Technology*, vol. 11(no. 5):pp. 743–749, 2010.

[4] L. Yu J. Hu B., Zhang. Kinematics and dynamics analysis of a novel serial-parallel dynamic simulator. *Journal of Mechanical Science and Technology*, vol. 30(no. 11):pp. 5183–5195, 2016.

[5] A. Emadi L.Bruck, B. Haycock. A review of driving simulation technology and applications. *IEEE Open Journal of Vehicular Technology*, vol. 2:pp. 1–16, 2021.

[6] Figure 1 https://commons.wikimedia.org/wiki/file:antoinette_sim _2.jpg/media/file:antoinette _sim _2.jpg.

[7] Gough's machine https://www.parallemic.org/reviews/review007.html.

[8] Figure 2 hexapod https://www.researchgate.net/figure/comparative-display-of-a-general-gough-stewart-platform-a-and-a-cable-robot-b_fig1 _318234661.

[9] Figure 3 nads-1 https://nads.uiowa.edu/nads-1gid=1pid=1.

[10] Figure 4 toyota driving simulator https://global.toyota/en/download/14221274/.

[11] Figure 5 dim 150 https://www.vi-grade.com/en/products/dim150 _dim250 _dynamic _simulator/.

[12] Figure 6 and figure 7 dim 400 at polimi https://www.vi-grade.com/en/products/cable-driven_dim_dynamic_simulator/.

[13] Figure 8 cockpit dim 400 https://www.drismi.polimi.it/media-downloads/?lang=it.

[14] Figure 9 layout of the simulator https://www.vi-grade.com/en/products/cable-driven_dim_dynamic_simulator/.

[15] Concurrent pc https://concurrent-rt.com/products/hardware/ihawk/.

[16] Specifications of simulator https://www.drismi.polimi.it/the-driving-simulator/.

[17] Vi-carrealtime https://www.vi-grade.com/en/products/vi-carrealtime/.

[18] Vi-worldsim https://www.vi-grade.com/en/products/vi-worldsim/.

[19] Roadrunner https://it.mathworks.com/products/roadrunner.html.

[20] Comolli, f., ballo, f., gobbi, m., mastinu, g., instrumented steering wheel: accurate experimental characterisation of the forces exerted by the driver hands for future advanced driver assistance systems, asme 2018 international design engineering technical conferences and computers and information in engineering conference,2018, american society of mechanical engineers digital collection.

[21] Eye tracker https://www.tobii.com/products/eye-trackers/wearables/tobii-pro-glasses-3.

[22] Reaction time to press a button https://www.heraldopenaccess.us/openaccess/time-of-reaction-in-using-the-pushing-of-button-in-experiment-of-evoked-cognitive-potentials-r300.

List of Figures

Acknowledgements

Voglio ringraziare il Professore Stefano Melzi e il Dottorando Antonio Cioffi per avermi dato la possibilità di partecipare a questa ricerca sperimentale: è stata una grande sfida, talvolta molto difficile, che mi ha dato anche grande soddisfazione.

Grazie a Edoardo, mi hai dato una grande possibilità, sto imparando tantissimo da te e spero che un giorno, grazie a te, arriverò dove sogno.

Grazie ai miei genitori, perché non ve lo dico mai abbastanza. Mi avete dato tutto il supporto di cui ho avuto bisogno per riuscire a superare la sfida più grande che io abbia mai affrontato. Grazie alla mia famiglia, perché stare con voi è un piacere e una fortuna che ha allietato le giornate più dure.

Grazie ai miei compagni di corso, sinceramente non avrei potuto sperare di trovare persone migliori. Abbiamo passato tantissimo tempo insieme e ci siamo fatti forza quando le sfide sembravano impossibili da raggiungere, ma insieme ce l'abbiamo fatta.

Grazie ai miei amici di sempre, siete il mio punto di riferimento, siete persone che stimo e mi avete dato grande forza negli anni.

Grazie agli amici della villetta e ai nuovi arrivati, perché avete portato una boccata d'aria fresca e nuovi stimoli, avete arricchito la mia vita.

Grazie anche alle persone care che oggi non ci sono più: ho avuto una infanzia ricca di amore e sono fiero delle persone che siete state. Sono sicuro che siete fieri di me e questo mi dà gioia.