Universidade de Aveiro Departamento de Engenharia Mecânica
2014

Pedro Xavier de Sousa Mendes

Intelligent Monitoring of Human Performance in Vehicle Pedals

DOCUMENTO
PROVISÓRIO

Universidade de Aveiro Departamento de Engenharia Mecânica
2014

Pedro Xavier de Sousa Mendes

Intelligent Monitoring of Human Performance in Vehicle Pedals

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob orientação científica de Vítor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro e de Ricardo José da Silva Pascoal, Investigador Pos-Doc do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President

Prof. Doutor xx
xx da Universidade de Aveiro

Vogais / Committee

Prof. Doutor Vítor Manuel Ferreira dos Santos
Professor Associado da Universidade de Aveiro (orientador)

Doutor Ricardo José da Silva Pascoal
Investigador Pos-Doc do Departamento de Engenharia Mecânica (co-orientador)

Prof. Doutor xx
xx da xx

Prof. Doutor xx
xx da xx

Agradecimentos / Acknowledgements

Aqui ficam os agradecimentos que eu quero fazer, à minha tia Felismina e ao meu primo Alberto. Muito obrigado!

Palavras-chave

Palavra-chave 1; palavra-chave 2; etc.

Resumo

Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor placerat fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit, congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet, integer ad vestibulum sagittis felis sodales, dolor sociis mauris, vel eu libero cras. Interdum at. Eget habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec el pellentesque leo, temporibus scelerisque nec.

Keywords Keyword 1; keyword 2, etc.

Abstract

Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor placerat fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit, congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet, integer ad vestibulum sagittis felis sodales, dolor sociis mauris, vel eu libero cras. Interdum at. Eget habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec el pellentesque leo, temporibus scelerisque nec.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Framework

The number of vehicles in operation worldwide is incredibly high and still increasing. In any transportation system safety is a major issue, leading to a huge socio-economic effort to reduce driving accidents, the largest cause of death in Europe, according to the Eurostat [1].

Safety is also one of the concerns underlying the *Atlas Project* [2]. This project was created by the Group of Automation and Robotics of the Department of Mechanical Engineering at the University of Aveiro. The *Atlas Project* is an ongoing work that started with the aim of developing autonomous mobile robots. Nonetheless, it has expanded to the vehicle safety field becoming an endeavour to create a car prototype for research on Advanced Driver's Assistance Systems — the *AtlasCar*, Figure 1.1.

The *AtlasCar* is a Ford Escort SW automobile equipped with sensors that provide data for the study and research of new automatic safety systems, and new challenges related to the autonomous driving. Among these sensors there are planar lasers, a 3D laser and a stereo camera that collect data about the surroundings of the car. The *AtlasCar* also has a GPS receiver that provides precise positioning, an odometry system that measures the velocity and rpm and several computing units responsible for controlling the car and reading all the sensors.

Despite the fact that there are many safety methods regarding the human driver, presently there is a high interest on computer vision-based techniques such as: analysing eyelid patterns, head movements, facial expressions and even searching for pedestrians or obstacles on the car surroundings. However, most of these techniques require extensive computational processing and high financial cost. As an alternative, other techniques like vehicle-driver interfaces are being considered. These techniques aspire to comply several ideal issues such as the following [3]:

- Low cost unit and operation, since an average automobile costumer may not be willing to pay very high prices;
- Real-time monitoring, to assure accuracy in detecting possible risk situations and reacting on time;
- Robustness and independence, meaning that the safety system performance should not be influenced by external factors like traffic, land-scape, weather, and others;

Figure 1.1: *AtlasCar [2].*

- Non-invasiveness, a monitoring system that would not distract or impair the driver performance;
- Easy-to-use and flexible, so it could be user friendly, with a simple and quick installation and compatible with any vehicle brands and models.

For being able to fulfil all of these criteria, the intelligent monitoring of human performance in vehicle-driver interfaces represents a great advantage for the assisted driving and for the study and analysis of driving risk situations. Having this as basis, the *Atlas* team proposed a project with the objective to develop a system to monitor the human performance. Because the vehicle pedals are a meaningful source of data output of the driver patterns, these were considered a powerful tool in monitoring driver performance. Hence, this project was entitled "Intelligent Monitoring of Human Performance in Vehicle Pedals" and was included in the *Atlas Project*, more properly in the *AtlasCar*.

1.2 Objectives

The aim of the "Intelligent Monitoring of Human Performance in Vehicle Pedals" is the creation and development of a system able to measure the performance of the driver actuation in the vehicle pedals. This system was projected to work in an interactive and intelligent way, so that the interactions between the pedals and the driver would be measured and simultaneously displayed or stored.

The interactions between the driver and the pedal provide variables such as the force applied by the driver on the pedal and the consequent pedal position shift. The ultimate objective is to correlate these data with other vehicle variables, such as acceleration, engine speed, surroundings of the car, etc. Regarding the vehicle driver monitoring ideals, the system was aimed to be modular, so that it could be easy-to-use, flexible, non-invasive, robust and low cost.

The project was subdivided in three main parts:

- Development of an easy-to-use and non-invasive hardware, to collect data from the pedals;
- Creation and development of software to monitor and calibrate the system in an interactive way;
- Evaluation and characterization of the driver's behaviour, through the analysis of the data from the pedals and the other variables of the vehicle.

1.3 State of the art

1.3.1 Transports and Safety

In 2011 more than 30,000 people died on the roads of the European Union (EU) and approximately 120,000 people suffered permanently disabling injuries, such as damage to the brain or spinal cord, 240,000 suffered other serious injuries and 1,500,000 minor injuries [4]. The number of vehicle accidents is incredible high and in order to reduce this leading cause of death there is a huge socio-economic effort to improve vehicle safety, safety of infrastructures and driver's behaviour.

Apart from the basic seat belt and air bag safety systems, there are the Advanced Driver's Assistance Systems, which are intelligent vehicle safety systems collectively referred to as *eSafety* systems. An eSafety system consists of equipping vehicles with electronic devices in order to help a driver avoid danger [4]. There are many different types of electronic safety systems for vehicles and new ones are constantly being developed. They range from collision-avoidance and brake-assistance to lane-departure warning systems. For example by activating the brakes if the car is too close to an object or by emitting warning signals if it strays outside its lane. Intelligent speed adaptation, is other example of an advanced system that allows the vehicle to know the speed limit in a specific area and immediately alerts the driver or directly limits the vehicle maximum speed. There are also seat-belt reminders, that emit a warning signal alerting the driver that the seat-belt is not fastened. In addition, electronic stability control, a system that eliminate loss of vehicle control, can help in critical situations such as maneuvers. To prevent a vehicle from being started if the driver is under the influence of alcohol, vehicles can be equipped with alcohol interlocks systems [5, 6, 7, 8].

Regarding driver's behaviour, there are several precautions that could improve road safety, for example: fastening the seat belts, respecting speed limits and not driving under the influence of alcohol or other states. Seat belts are the easiest and cheapest way to avoid injury in a car crash. Regarding the speeding, that not only increases the risk of accident but also the likelihood of severe injuries or death, the prevention should consist of a combination of several improvements, from road design to speed limits and driver education. Drivers should not only respect the speed limits but also keep their attention focused on surrounding traffic at all times. Impairment of attention may be due to risk behaviours, such as driving under the influence of alcohol and drugs or to sleepiness, fatigue, monotony and distraction states due to underestimation of physiological needs,

Figure 1
three bas

- Vis
  che

- Co
  wo

- Ma
  exa

Even
diminish
a system
conseque
prevent a

Figure 1.2: Factors impairing driver performance [3].

1.3.2 Monitoring Human Performance

Some driver behaviours that impair safety driving can already be detected in real-time. For instance, drowsiness and inattention can be captured through image acquisition and processing of facial features such as eye and head movements [3]. However, these image processing techniques are associated with high computational costs and only capture the head or upper body.

As an alternative there are other Advanced Driver's Assistance Systems that not only can monitor the physiological responses of the body while driving but also have the advantage of interacting with the driver — vehicle-driver interfaces. Vehicle-driver interfaces based-systems have the potential to improve safety by observing and interacting with the driver, leading to a decrease in risk behaviours and consequently road accidents [9].

Vehicle-driver interfaces include the accelerator, brake and clutch pedals, the steering wheel and the seat [3]. There are some techniques for vigilance monitoring based on these vehicle-driver interfaces, like steering wheel and seat pressure monitoring combined with actuation on the pedals. For instance, measuring forces using force sensitive resistors (FSR) on pedals, together with monitoring of the steering wheel improves the detection of driver's alertness state [3]. In addition, using a driving simulator, Figure 1.3, the behaviour signals measured on the steering wheel and on the pedals provide the possibility to identify the driver's identity [10]. If the pedal data is combined with seat pressure data, monitoring the pedals also improves the analysis of driver ergonomics [11]. Moreover, by monitoring the engine moment, the acceleration and the steering wheel together with the brake pedal data it is possible to analyse the driver's condition and capabilities [12]. Finally, a system combining the data from the pedal position with the vehicle location, speed, acceleration, fuel consumption and the heart rate of the driver can assist in the characterization of driver behaviour as well as in the identification and mapping of potentially dangerous locations [13].

The availability of vigilance monitoring systems using only output from the pedals is far more restricted. A project, where data from FSR on pedals was used to identify the driver by analysing driving performance patterns is one of the few examples. In this study, different driving patterns were based on previously collected data [14].

A driver performance monitoring system should be low cost, non-invasive, easy-to-use, flexible, and real time accurate [3]. However, the few available systems for monitoring the pedals usually lack modularity and flexibility. A very flexible and easy-to-use system to monitor the pedals, that could be easily installed and readily used in any vehicle or machinery would therefore be of great importance filling a major gap in the field. Moreover, if capable of combining both the ability to measure applied forces on the pedals and the consequent shift in distance between the pedal and the floor of the car, this system would represent an even more accurate and robust method than the presently available.

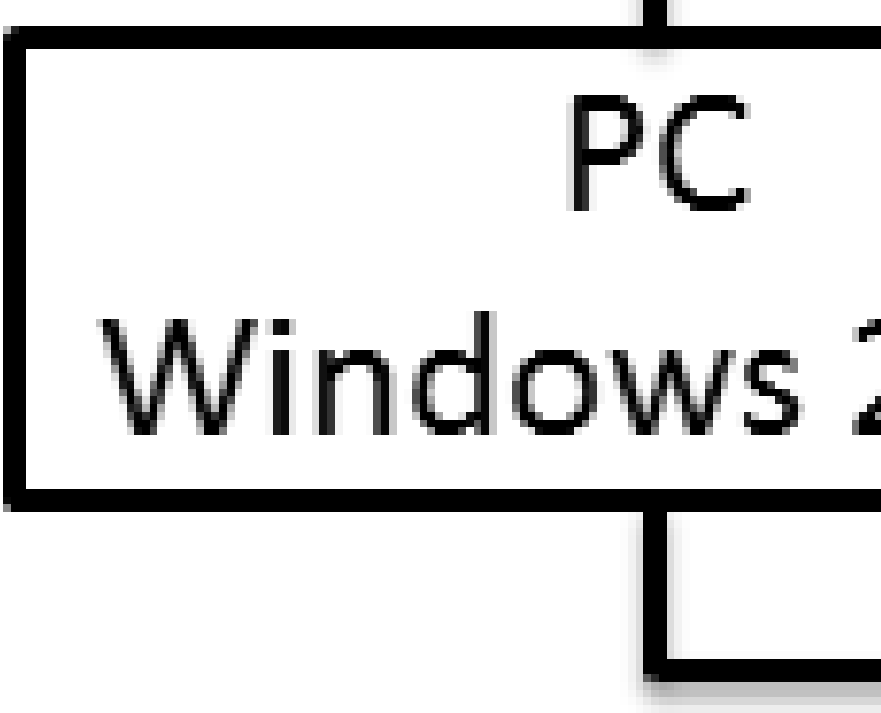


Figure 1.3: Driving simulator to collect behaviour signals [10].

Chapter 2

Hardware Solution

The main objective of the first part of this project was the development of a modular, easy-to-use, flexible, non-invasive, robust and low cost system able to collect data from the pedals.

2.1 Sensors

Firstly, it was thought to use force sensitive resistors (FSRs) to measure the applied forces on pedals; FSRs are often used due to their small size and the ability to measure forces. Nonetheless, there are also other sensors for measuring applied forces such as load cells (LCs). Although there is some reluctance in using LCs, mainly due to their greater dimensions when compared to FSRs, they have higher accuracy and repeatability, which are very important aspects in a system with these requirements. Therefore, and because both sensors have important advantages, both FSRs and LCs sensors were used in this system.

Apart from the ability to measure applied forces on the pedals, it is possible to measure the pedal displacement against the car floor using another type of sensor. For the purpose of measuring the pedal displacement there were used infrared (IR) sensors that work perfectly for the dimensions of the pedal course.

Regarding this, three types of sensors were used on this system: force sensitive resistors (FSRs), load cells (LCs) and infrared (IR) distance sensors.

The FSRs are very thin sensors that have variable resistance depending on the applied force. There are several FSRs configurations, varying in size and capacity. The FSRs used for this system were long-sized and with the capability to measure forces up to approximately 2 Kg [15].

The LCs are sensors composed by strain gauges that are deformed when force is applied. According to the applied force, the tension varies on the LC terminals. There are various different types of LCs, such as: 1) single-point load cells, usually used in small jewelry and kitchen scales, 2) button load cells, used in applications that required a thin form factor and 3) s-type load cells, that can be operated in compression or in tension. The type of the LC used for this system was button load cells with the capability to measure forces up to 50 Kg [16].

The IR distance sensors are built in small packages and have very low power consumptions (50 mA). There are mainly two types of IRs: analog, that provide analog

information abou
if an object is on
outputs based on
cursor, the IR se
from 4 to 30 cm [

In order to r
which is than co
PhidgetInterface-
preliminar tests.
puts and outputs
the PhidgetInterf
do that it was ne
adapters provided

2.1.1 Force S

The force sensitiv
have variable resi
2.1. The sensing
suspended in mat
with the conduct
principle is: the h

Figure 2.1: Interlink Electronics 0.5" Circular 20N FSR [15].

These sensors are easy to use and relatively inexpensive, however, they have some disadvantages such as drift and poor accuracy, with errors up to 25% [15].

Depending on the time and the strength at which they are pressed, the FSR output

value will come ba
drift the final 5%
when setting up a
combination with
even though the F
approximately 2 k
will quickly satur
if the driver is tou

2.1.2 Force L

A Load cell (LC)
2.2. The LC has s
applied; the strain
signal. The electr
it is required the

Figure 2.2: Button Load Cell (0–50kg) [16].

Most designs of load cells use 4 strain gauges, just like a Wheatstone bridge, but there are also other types which only use 2 strain gauges, often called half bridges [16].

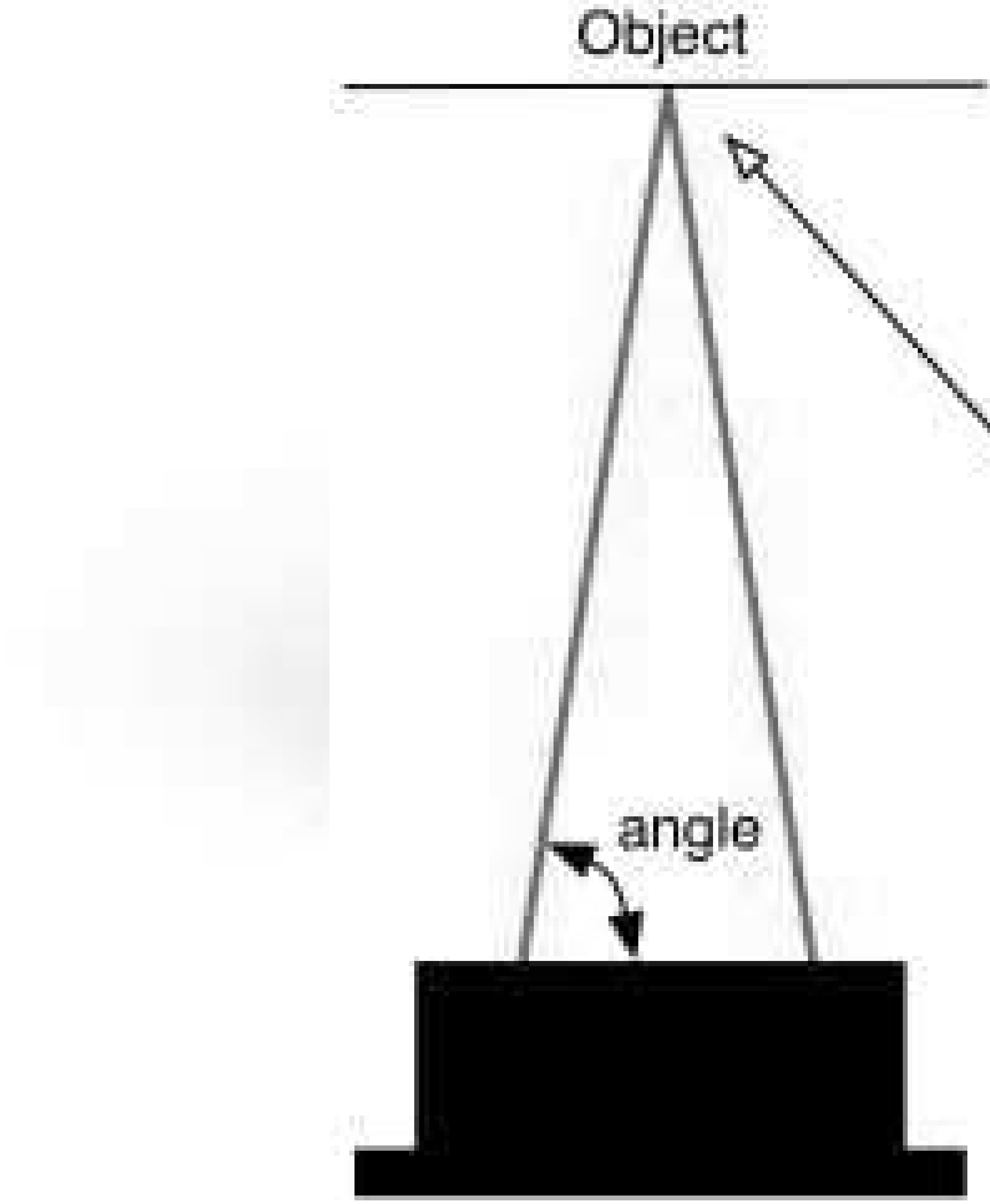
A Wheatstone bridge is an electrical circuit that measures an unknown resistance by balancing two nodes of the bridge circuit. It has two input terminals and two output terminals consisting of four resistors configured in a diamond-like arrangement. In the case of the load cells, these resistors are strain gauges [18]. Strain gauge load cells offer accuracies from 0.03% to 1% full scale. The load cell used in the system has a small weight range thus its accuracy full scale is 1%.

The load cell electrical output (mV/V) can be converted to force or weight using equati

Where
going t
is nece
measu
it. On
solve t
when i
in this
of usin
the se
when

2.1.3

The Sh
object,
infrare
return
Triang
distan



(a) (b)

Figure 2.3: a) Sharp Distance IR Sensor (4-30cm) [17]. b) Sharp IR detector angle of reflection arrival for near and far object [17].

Due to the trigonometry involved in calculating the distances based on the reflected light incident angle, the output of the sensor is non-linear, Figure 2.4. By analysing the graph it is possible to observe that the output within the range (4 – 30 cm) of the sensor is not linear. Also, when the range is closer to the minimum distance (4 cm) the output drops quickly, which means that the system will respond as if the object is moving away from instead of approaching. This particular feature could represent a problem for

distances lower than 4 cm
adjust the IR sensor orient

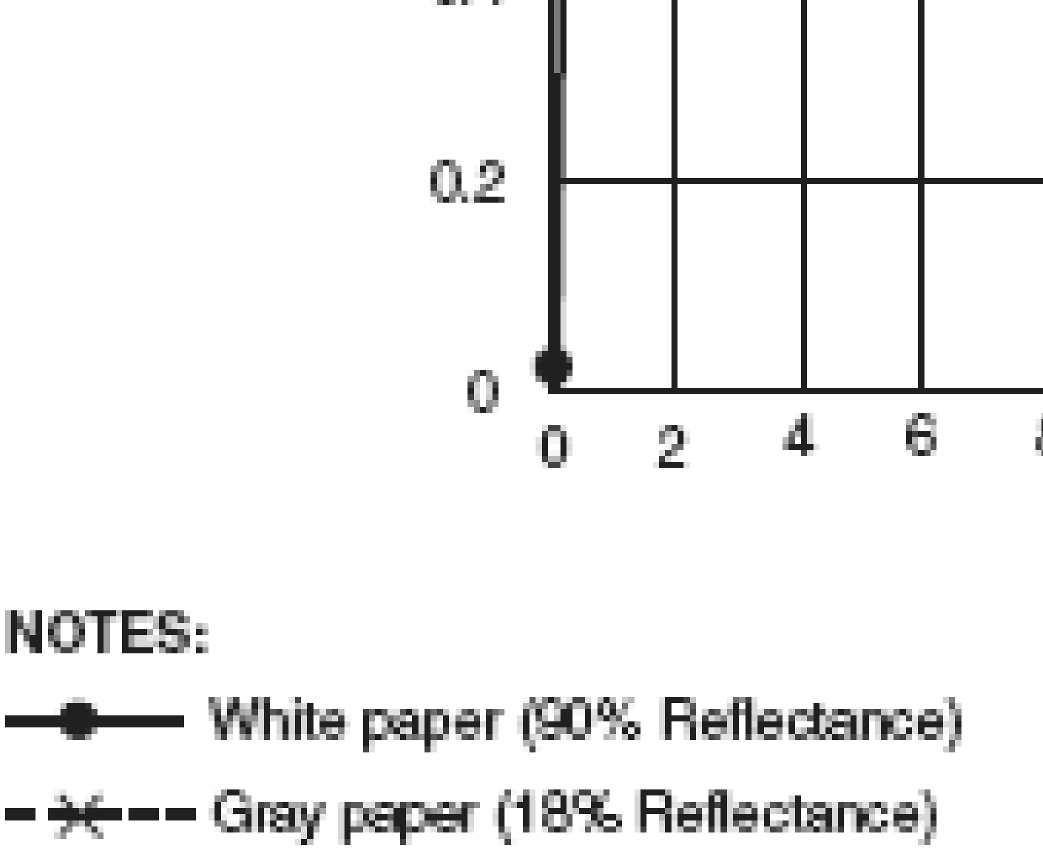


Figure 2.4: Sharp IR output voltage vs distance.

Regarding the sensor values, it is possible to convert the analog output from the sensor into distance using equation (2.2):

$$Distance(cm) = \frac{2076}{SensorValue - 11} \tag{2.2}$$

Equation (2.2) is only valid for digital output values between 80 and 530 bytes, which represent approximately 0.3V at 30cm and 3.1V at 4cm, based on standard values from Sharp. In order to have a better performance of the sensor it is suggested to derive the equation for the actual sensor values. However, using the standard equation on this system, the IR sensors provide good results with the standard formula, so it is not necessary to derive their own equation [19].

2.2 Mechanical System

As mentioned, this system was composed of three different types of sensors: force sensitive resistors (FSR), load cells (LCs) and infrared (IR) distance sensors.

In order to install the sensors on the pedal, a case able to lodge 2 LCs, 2 FSRs and 2 IR sensors was developed. When designing the case, special attention was given to the dimensions, because the case cannot be an obstacle for the driver. For this reason, and also to make it lighter, the case was manufactured in aluminium with the smallest dimensions possible to lodge the sensors; 30 x 80 x 14 mm, Figure 2.5.

The load cells and force sensitive resistors were meant to be fixed on the case, whereas the infrared sensors were planned to be installed on the sides of the case, with the possibility of adjusting their orientation, Figure 2.6. The reduced dimensions of the FSRs made it possible to install them beneath the LCs, making the system more redundant and thus more accurate.

The case is composed by two components, the bottom component where the sensors are installed and the top component that consists of a touch sensitive lid. Both bottom and top components cannot be fixed together, otherwise the force applied by

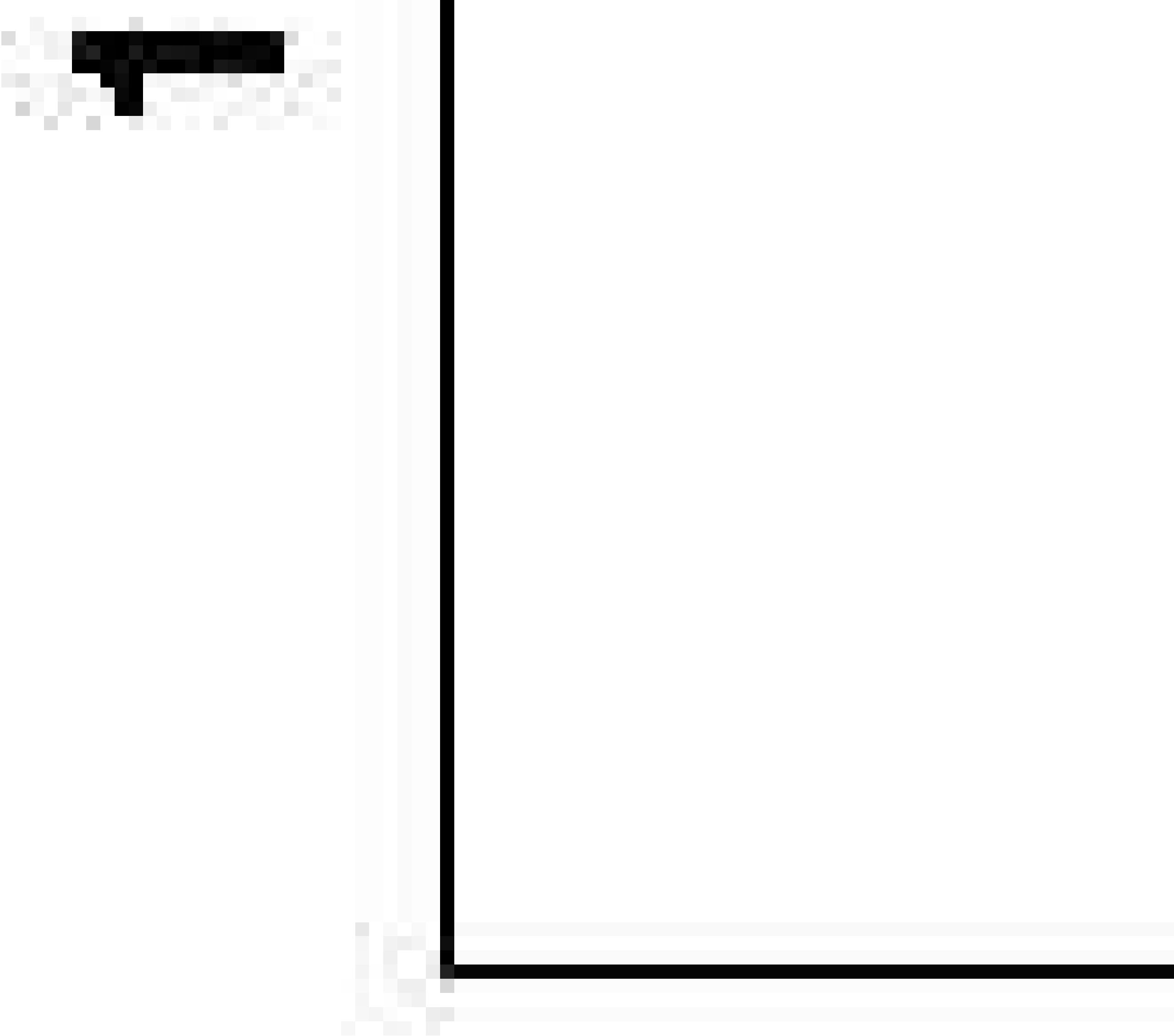


Figure 2.5: Case mechanical drawings.

the driver on
dissipated thr
this, it was ne
bottom and th
sensors would
tion/uninstalla
on Velcro strip

Figure 2.6: a

(a) (b)

Figure 2.7: a) Case fixing system. b) Case with the IR orientation system installed on a gamepedal.

2.3 Electronic Devices

As mentioned, the output values of the sensors need to be conditioned and amplified. In order to do that two different types of boards from Phidgets were used, one for the infrared (IR) distance sensors, and for the force sensitive resistors (FSR), and other custom made board for the force load cell (LC).

The board used for the FSR signal conditioning was the Voltage Divider, Figure 2.8, which provides an interface to resistance sensors such as: FSRs, light sensors, heat sensors and bend sensors. Because the Voltage Divider was used, there was no need to connect the FSR sensor to a resistor and then to an analog input [20]. Also, this board has a

high precision potentiomet
To optimize the voltage d
screw on the potentiomete
to increase it [21].

Figure 2.8: Force Sensitive Resistor Adapter from Phidgets [20].

The LCs are sensors that require an instrumentation amplifier (in-amp) to amplify their output value, which is in millivolts. Therefore, it was developed a circuit board to mount the sensors, amplify the signals and connect to the PhidgetInterfaceKit 8/8/8 responsible for the data acquisition, Figure 2.9. This board, also named LC adapter, was designed using the CadSoft Eagle Software and hosts an instrumentation amplifier, AD623, with a gain that can be calculated using equation (2.3):

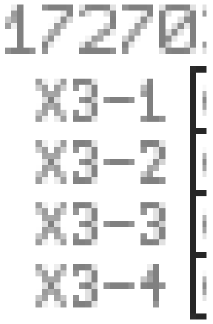
$$G = 1 \times \frac{100k}{Rg} \tag{2.3}$$

Where the Rg is the resistor connected to pin 1 and 8 of the in-amp. The resistor installed in the board has 160 Ω; placing this value on the equation it is possible to calculate the gain of the board that is approximately 600.

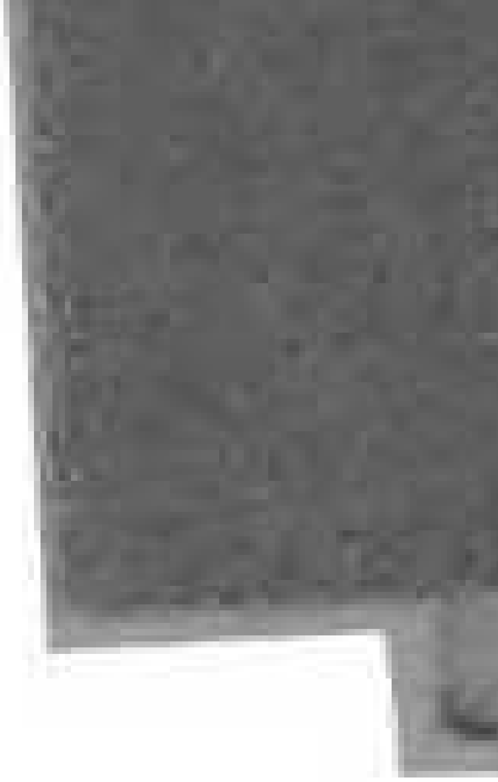
For the IR distance sensors signal conditioning, the IR Distance Adapter board was used, Figure 2.10. This board provides an interface to an analog input on the PhidgetInterface-kit 8/8/8 board, mentioned before, and regulates the power requirement of the connected sensor [22]. Although the specified current consumption is low, they may draw much more current for short periods during measurements. Because the system will have several sensors installed, the IR adapter boards are an important aspect to prevent the PhidgetInterfaceKit 8/8/8 from overload due to high current consumption.

2.4 Acquisition System

As mentioned, for the data acquisition, the PhidgetInterfaceKit 8/8/8 board was used, Figure 2.11 [23]. This board is provided with a MCP23S17, which is a 16-bit input/output (I/O) expander with Serial Peripheral Interface (SPI) interface. The MCP23S17 device consists of multiple 8-bit configuration register for I/O and of a system master responsible for enabling and disabling these inputs or outputs. The data for each I/O is kept in the



(a)



(c)

Figure 2.9: a) Force Load Cell adapter circuit scheme. b) Force Load Cell adapter PCB circuit. c) Force Load Cell Adapter.

Figure 2.10:

corresponding register
Also, the board has a M
MCP3008 is a device th
to a digital value [25].
controller. This device i
the communication used

Figure 2.11: PhidgetInterfaceKit 8/8/8 Acquisition Board from Phidgets [23].

Nonetheless, instead of using the PhidgetInterfaceKit 8/8/8 it is also possible to use an Arduino, for example the Arduino Nano v3.1, Figure 2.12 [27]. The Arduino Nano v3.1 is a single-board microcontroller, with smaller dimensions, more flexible and more accessible than the PhidgetInterfaceKit 8/8/8. Actually, as mentioned above the Arduino Nano was used on preliminar tests to read the data from the sensors, however,

for convenience the Phid
boards were also from I
robust.

Figure 2.12: Arduino Nano v3.1 [27].

Besides the Arduino it is also possible to use a single-board computer (SBC), for example a Rasberry pi. These arduino or SBC solutions are more suited for a final approach due to their lower costs and firmware flexibility.

However, to apply these solutions would be required more electronic devices to assure the signal conditioning. In case of using an Arduino, to be able to send the data from the sensors to it without distortion, etc, it would be necessary to use tension to current (V to I) converters.

2.5 Setup

After the conclusion of the hardware and electronic devices, it is possible to assemble all of the components and create a setup to start monitoring the pedals, Figure 2.13. To do so, it is necessary to:

- Mount the sensors on the case 2: force load cells, 2 force sensitive resistors and 2 infrared distance sensors;
- Connect the sensors to the signal conditioning boards: 2 voltage dividers, 2 IR adapter and 2 LC adapter;
- Connect the signal conditioning boards to the acquisition board, PhidgetInterfaceKit 8/8/8;
- Connect the acquisition board to the computer, launch the software interface and start monitoring.

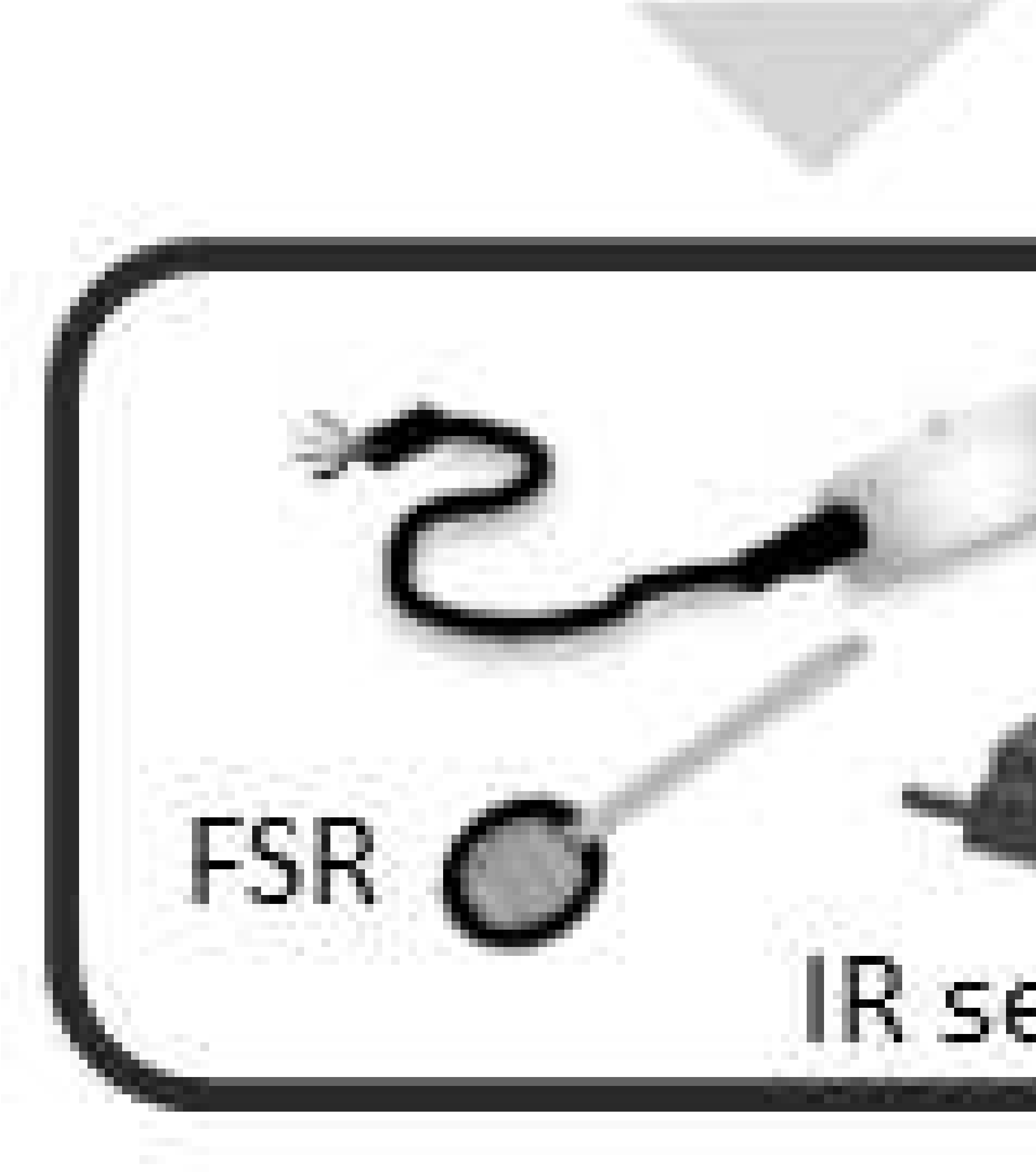


Figure 2.13: Setup scheme.

Chapter 3

Software

The second main objective of this project was the creation and development of software to monitor the pedals and to calibrate them in an interactive and intelligent way. This software development was divided in two parts: a program responsible for the interface and a program responsible for monitoring and publishing the data collected from sensors.

3.1 Development Tools

As for every member of the Laboratory for Automation and Robotics (LAR), in order to develop the software, the use of C/C++ languages and the Robotic Operating System (ROS) environment, already implemented in the LAR and on the *AtlasCar*, were strongly encouraged.

Regarding this, the software was developed under a Linux platform using C/C++ programming on the ROS structure, which provides libraries and tools to help the software development. In addition, the GTK+ toolkit was used to develop the graphical user interfaces, together with the Glade Software, which is an useful software for designing interfaces. In addition, it was necessary to use firmware libraries in order to communicate with the data acquisition system.

3.1.1 Robot Operating System

The Robot Operating System (ROS) is a flexible software development environment designed especially for the creation of robot software. Due to its modular structure, the ROS is able to assist the large-scale projects, dividing them into smaller modules with a more specific application. Besides reducing the size, it also reduces the complexity of the project by making it easier to debug and **comprehension** [28].

The ROS is organized into packages, which are directories containing software from each module, and allows to establish communication between them. This communication is performed through messages: first, a module (node) publishes a message on a ROS Topic and then, another node subscribes to that topic and automatically start to receive messages from the node that is publishing. While communicating, both nodes are decoupled, Figure 3.1. This communication structure allows to publish simultaneous topics and these topics can be subscribed for multiple nodes. In this way, a module can fulfil multiple tasks and be easily integrated in a simple or multiple research project [29].

Figure 3.1: Basic ROS communication concept [29].

Another important aspect provided by ROS is the ability to store ROS data messages into *bags*. *Bags* are files capable of storing all the published data, being considered the primary mechanism in ROS for data logging. These are usually created by the *rosbag* package, which subscribes one or more ROS topics so that the messages are stored as they are received. After recorded, *bags* can be played back and manipulated by tools in the *rosbag* and in the *rxbag* [29].

Due to the important aspects of ROS, the *Atlas Project* has been migrating every application to this environment. This is the main reason why members of the LAR are encouraged to implement it in their projects.

3.1.2 GTK+ and Glade Software

One of the most important aspects of an application is the interface that is provided to interact with the user, who expect those interfaces to be graphical. Thus, for this system, a graphical interface to the user was designed using GTK+ and the Glade Software.

The GTK+, or GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces. GTK+ is an object-oriented application programming interface that relies on multiple libraries such as: the GLib Object System (GObject) and the GIMP Drawing Kit (GDK), each providing a specific class or functionality. In order to provide the necessities of building graphical user interfaces, the GTK+ offers a complete set of widgets such as [30]:

- Windows: normal windows or dialogs;
- Displays: labels, images, progress bars and status bars;
- Buttons and toggles: check buttons, radio buttons, toggle buttons and link buttons;
- Numerical and text data entry and multi-line text editor;
- Menus, toolbars and layouts.

The GTK+ w
important aspect

Figure 3.2: The widget hierarchy of GTK+ [30].

On the other hand, the Glade software is a user interface designer which complements the GTK+. Glade is a rapid application development tool that allows quick and easy development of user interfaces for the GTK+ toolkit. The designed user interfaces are saved as XML files and then can be loaded using the *GtkBuilder* in GTK+. With Glade it is possible to separate the user interface layout from the code and this is very useful when developing a more complicated user interface, with a large number of widgets [31].

3.1.3 Firmware Libraries

In order to read the data from the sensors, the PhidgetInterfaceKit 8/8/8 which is the main part of the acquisition system was used. For the communication with this board it was necessary to install the proper drivers and libraries, which are provided by Phidgets.

Before starting to develop the software, it was important to understand how the libraries work and which functions are required to get the data from the board. There are four essential functions for this board to work, which are:

- void *start()*;
- int *AttachHandler()*;
- int *DetachHandler()*;
- int *ErrorHandler()*;

For example, b
functions to it, w
8/8/8 will be att
board is attached
CPhidgetInterface
the ID of the boa
(which ranges fro
board, and 3) a v

3.2 Interfa

As mentioned, th
Glade and GTK+
system. This inte
data received fron
file responsible fo
and the acquisitio

The interface
in the main file.
3.3, where the use
2) the *New calibra*
3) the *Load calibr*
and monitor the p

Figure 3.3: Interface main menu.

By selecting the *New calibration menu*, a dialog box, where it is possible to enter the username and the number of pedals to calibrate, is shown. By clicking in *Continue*, the

new calibration
In order to do
instructions, Fi
and a save file

(a)
pe

Figure 3.5: Calibration instructions scheme.

Afterwards, it is possible to load the calibration from the file that was created. When the *Load calibration menu* is selected, an open file dialog is shown and the user can select the calibration file he wants to load. When the calibration file is loaded, the data from sensors is automatically calibrated and displayed.

In order to create a new calibration or to load an existing one, it is necessary to read the data from the acquisition system in real time. However, the interface is loaded by using the GTK+, which is a blocking function and thus blocks computer processing.

Figure 3.6: Interface load calibration menu.

To make it possible the use of other functions at the same time as the GTK+, it was necessary to create a thread, making the computer processor work in parallel instead of working in a single process. In this way, it is possible to run the interface and at the same time run the functions responsible for collecting the data from the acquisition system.

In order to access the sensor values, another set of functions using the Phidgets library were created inside this thread. Among this set of functions, two were responsible for the reading rate: one at 40Hz to read the infrared (IR) sensor values, and one at 100 Hz to read the load cell (LC) and the force sensitive resistor (FSR) values. These reading rates had to be different due to the different performance of the sensors. Inside these functions, there are other functions responsible for the actual reading of the data and, if a calibration file is loaded, they process the data calibration: *readIRSensors*, *calibrateIRSensors*, *readLCSensors*, *calibrateLCSensors*, *readFSRSensors* and *calibrateFSRSensors*.

3.3 Monitoring and Publishing

As mentioned, besides the interface, another program was created. This program was developed to be responsible for monitoring and publishing data on ROS. Thus, in order to launch this program an existing calibration is required.

The monitoring and publishing program is meant to be launched when the user want to monitor the pedals and when a calibration file already exists. Similarly to the interface, this program also has functions responsible for reading the data from the sensors. However, instead of launching an interface, it loads the calibration XML file and with the calibration data as a basis, the received data is calibrated and published on ROS. This program publishes two types of messages: 1) a default message with the type sensor_msgs/range, which is the type of message published by the infrared (IR) sensors and 2) a pressure_cells/SenVal, which is the type of message published by the load cells (LC) and the force sensitive resistors (FSR), Figure 3.8.

These messages can be displayed using the ROS rviz, which is a ROS package that

Figure 3.7: Code organization diagram.

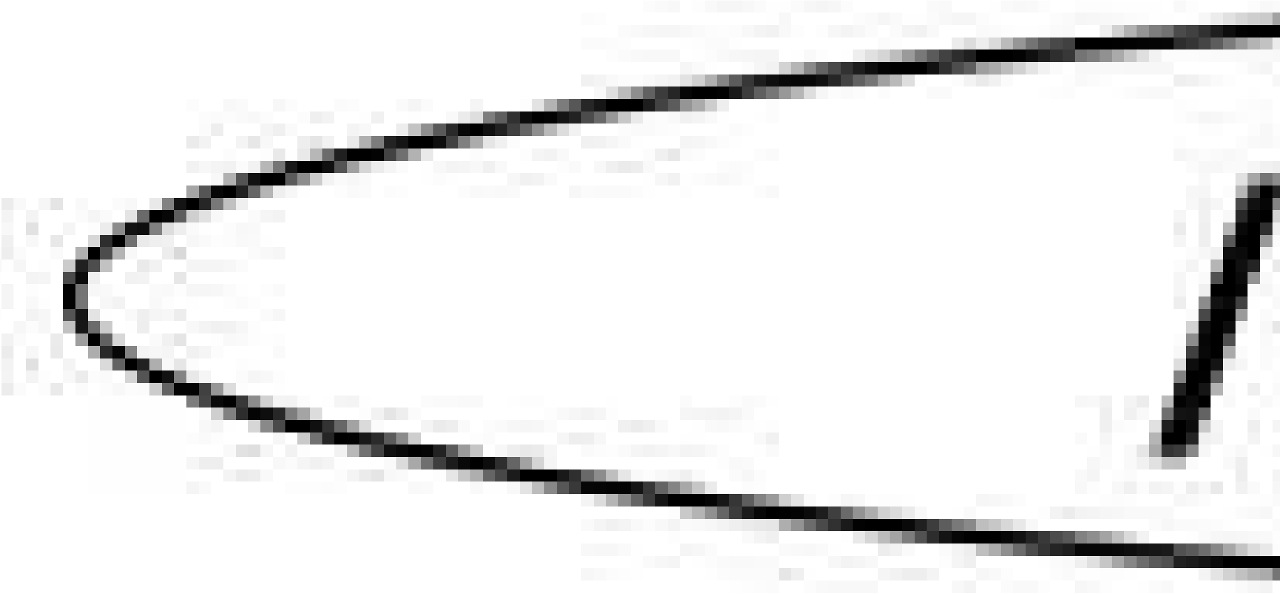


Figure 3.8: Published messages on ROS.

allows the visualizatio
type for the SenVal m
presented, Figure 3.9;
values and the red col

Figure 3.9: Published data displayed on Rviz.

Chapter 4

Results and Discussion

As mentioned on the first chapter, this project aspired to develop a system that could be able to fulfil several ideal issues, such as: flexible, easy-to-use, non-invasiveness, robustness, real-time monitoring and low cost.

In order to validate if the system was capable of comply these requirements, several experiments were performed. These experiments were conducted in two states: in stationary driving situations and in real driving situations.

4.1 Stationary Driving Situations

In stationary driving conditions, it was possible to test if the system was flexible and easy-to-use. In order to do that, the system was installed in various vehicles and in different pedals, while the car was off.

The first test was performed in a Peugeot 206, Figure 4.1. The system was installed in the clutch and brake pedal.

Also, the system was installed and tested in a X, Figure X.

For all the tested vehicles and pedals, the system revealed an effortless and quick installation, making possible the validation of the flexibility of the system and its ease of use.

4.2 Real Driving Situations

In real driving conditions, it was possible to test the non-invasiveness and the robustness of the system, and if it was actually possible to monitor the pedals in real-time. Similar to the previous tests, these were also performed in various vehicles and for different pedals.

In order to validate those conditions, the system was installed and tested in during several driving situations, which are:

- The same driving route for the same vehicle and different drivers; **is it possible to detect different behaviours??**
- The same driving route for different vehicle and the same driver; **is the performance the same?? — independent of the vehicle**

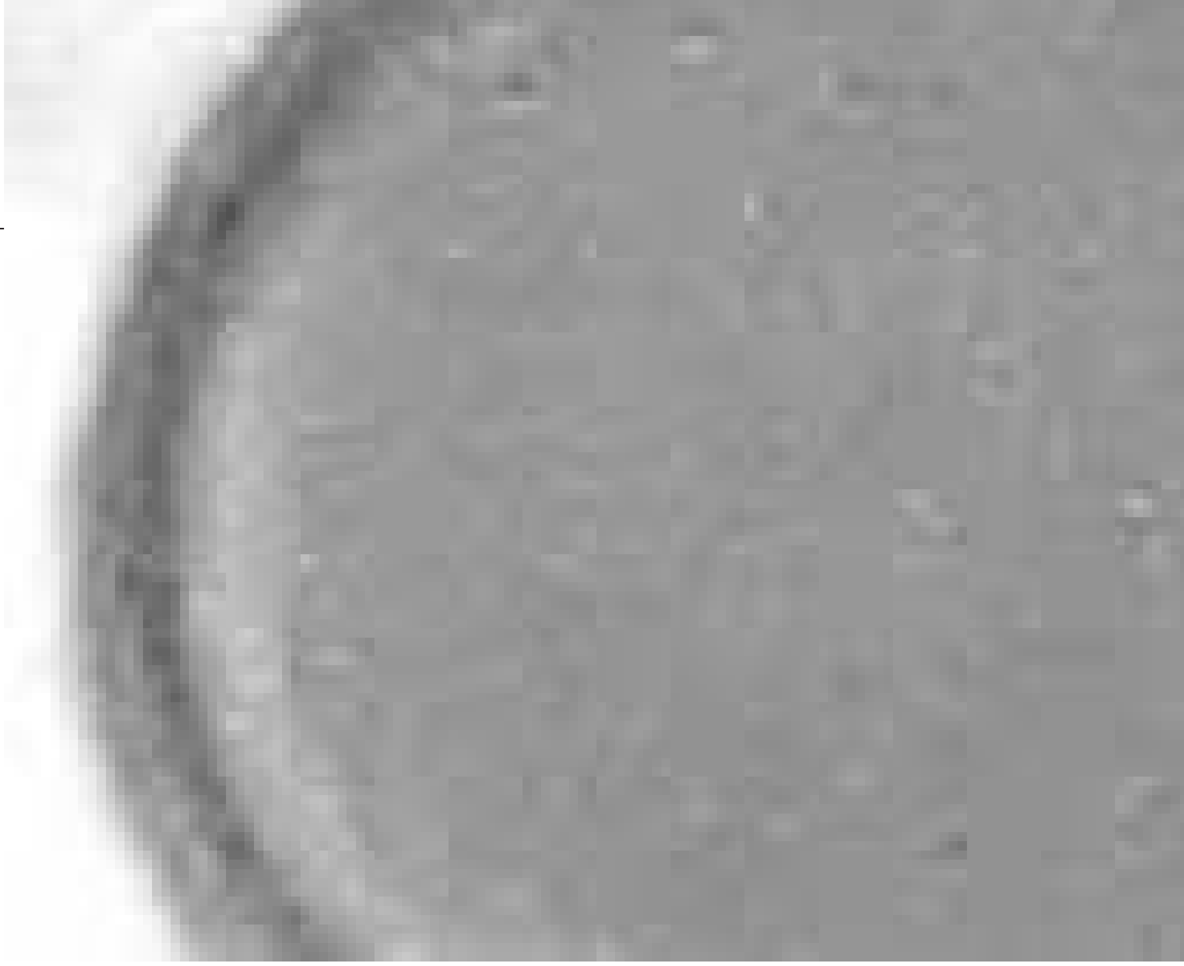


Figure 4.1: Setup installed on a Peugeot 206.

- A driving route with dangerous situations for the same vehicle and different drivers; **is it possible to predict risk situations?? and how is the driver behaviour in those risk situations: aggressive, passive, ...??**

- Use a calibration from one driver in a different one. **is it possible to detect that it is not the right calibration??**

PRELIMINAR TESTS??? :

Before performing these tests, it was made a preliminary test in different conditions as these above, in order to see the system capabilities. For this preliminary test, it was used the same vehicle for two different routes, Figure 4.2. The first route the system was installed in the clutch pedal, while the second one was installed in the brake pedal.

(a) (b)

Figure 4.2: a) Clutch collecting data route. b) Brake collecting data route [32].

With this preliminary test, due to the fact that the system was not a nuisance to the driver, it was possible to validate the non-invasiveness condition of the system. Also, while driving it was possible to test the robustness of the system. The system showed to be very robust in terms of collecting data, however, in terms of the fixation on the pedals it was not much effective.

Comparing the collected data from the clutch, Figure 4.3 and 4.4, and from the brake, Figure 4.5 and 4.6, it was possible to

Finally, in addition to these results for the both states, this system was meant to be low cost, however due to the high costs of the acquisition and signal conditioning boards, this prototype cannot be considered that way. Nonetheless, instead of using the PhidgetInterfaceKit 8/8/8 acquisition board it is possible to use other devices, for example the Arduino. As mentioned before, these devices are more flexible and have lower costs than the Phidgets board, thus, using this kind of devices would make this system more low cost.

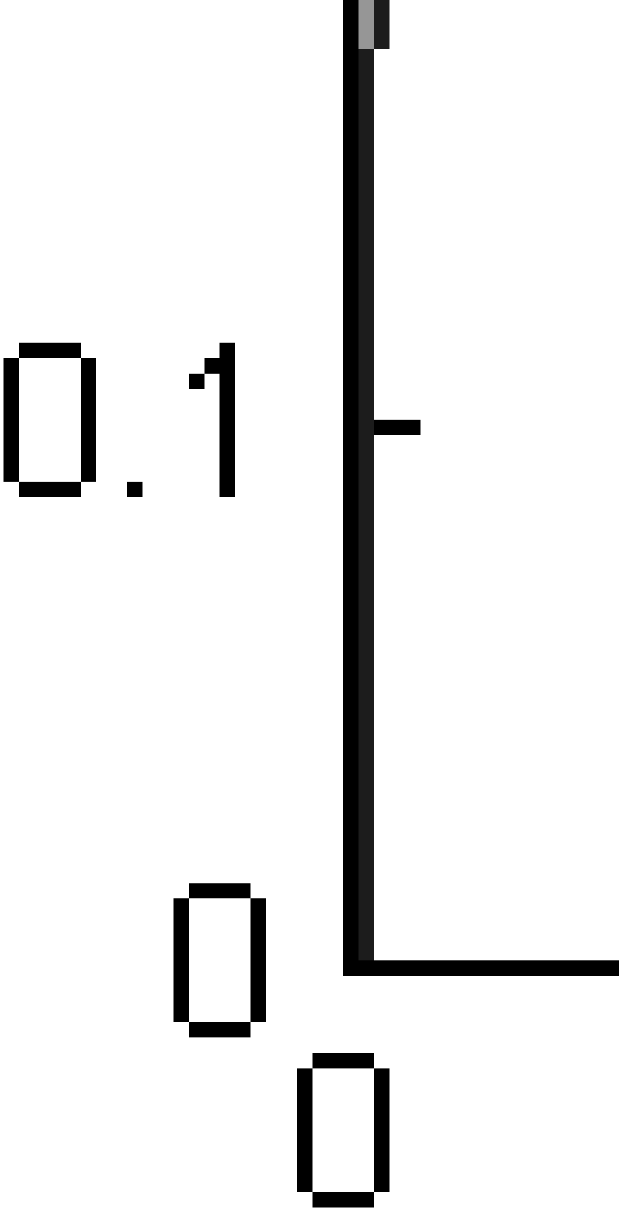


Figure 4.3: Collected data from clutch (LC and FSR).

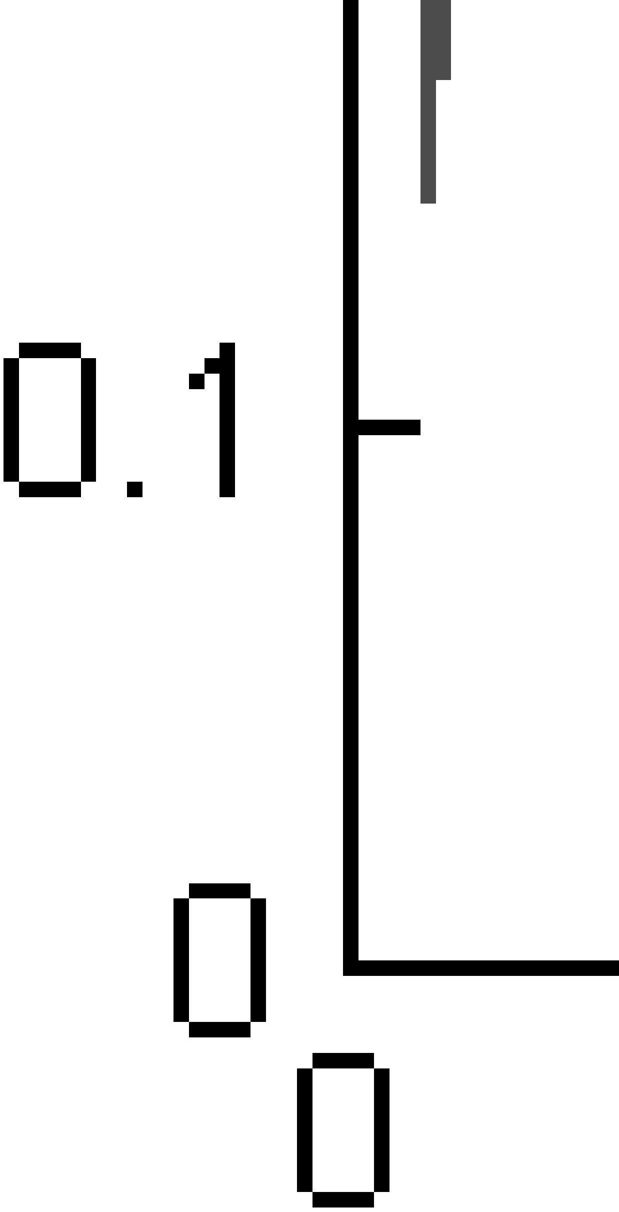


Figure 4.4: Collected data from clutch (IR).

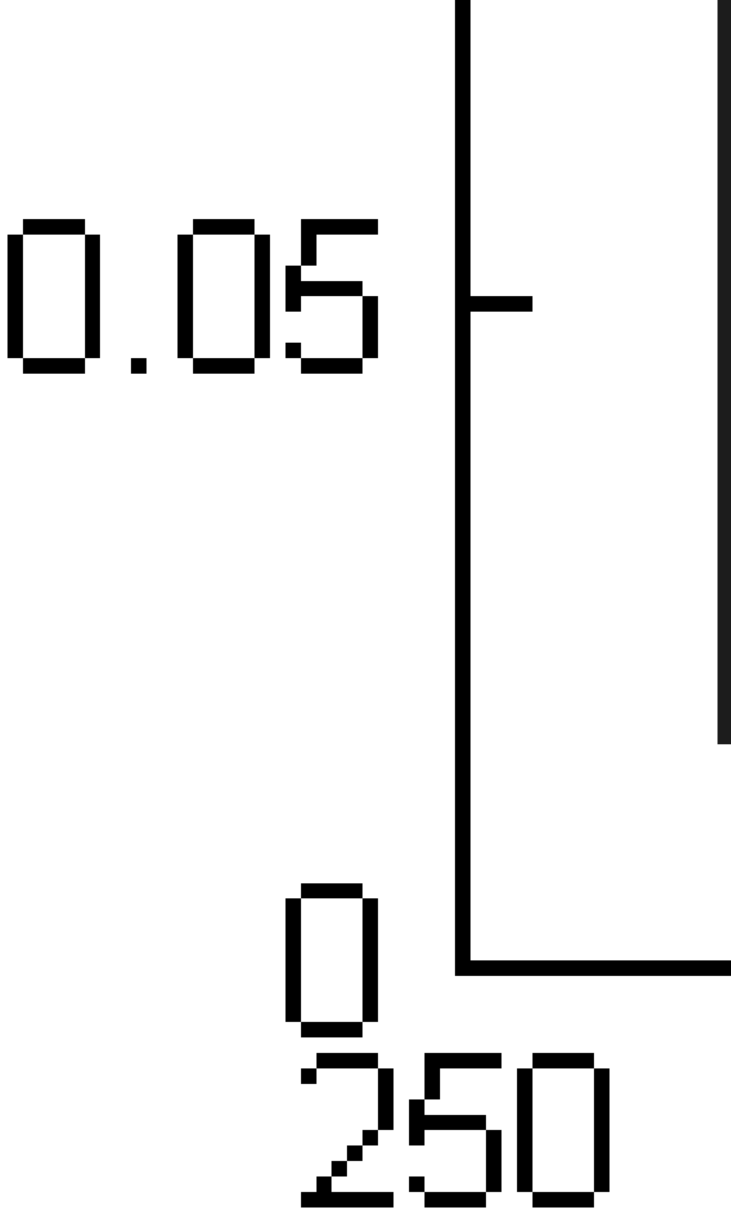


Figure 4.5: Collected data from brake (LC and FSR).

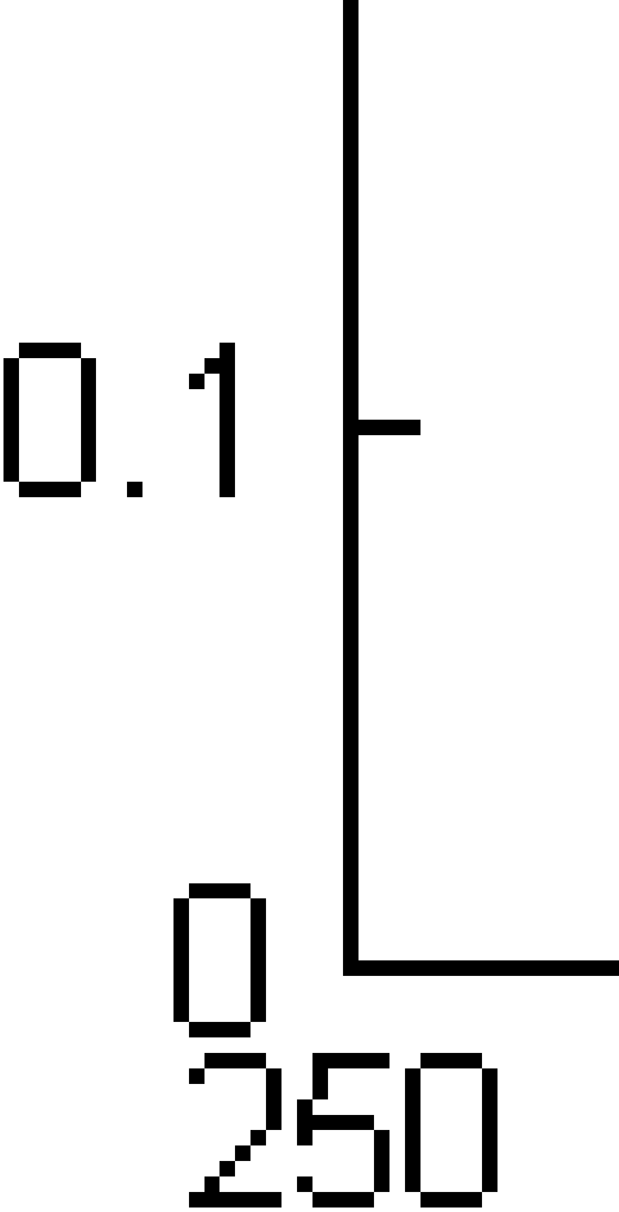


Figure 4.6: Collected data from brake (IR).

Chapter 5

Conclusion and Future work

Future work:

- Using Arduino to read data, instead of Phidget interfacekit board, communicating via BLE with the Interface/Client.
- Interface in Android Application or IOs Application.
- Much better, using a raspery Pi with local wifi network and an browser interface using HTTP protocols. (UNIVERSAL SOLUTION FOR ALL DEVICES)

References

[1] Eurostat, "Causes of death statistics - statistics explained." `http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Causes_of_death_statistic`

[2] "ATLAS project." `http://atlas.web.ua.pt/`.

[3] A. V. Desai and M. A. Haque, "Vigilance monitoring for operator safety: A simulation study on highway driving," *Journal of safety research*, vol. 37, no. 2, pp. 139–147, 2006.

[4] E. Commission, "Road safety - european commission." `http://ec.europa.eu/transport/road_safety/index_en.htm`.

[5] O. M. Carsten and F. Tate, "Intelligent speed adaptation: accident savings and cost-benefit analysis," *Accident Analysis & Prevention*, vol. 37, no. 3, pp. 407–416, 2005.

[6] C. M. Farmer and J. K. Wells, "Effect of enhanced seat belt reminders on driver fatality risk," *Journal of safety research*, vol. 41, no. 1, pp. 53–57, 2010.

[7] A. Lie, C. Tingvall, M. Krafft, and A. Kullgren, "The effectiveness of electronic stability control (ESC) in reducing real life crashes and injuries," *Traffic Injury Prevention*, vol. 7, no. 1, pp. 38–43, 2006.

[8] D. J. Beirness and P. R. Marques, "Alcohol ignition interlock programs," *Traffic Injury Prevention*, vol. 5, no. 3, pp. 299–308, 2004.

[9] A. Doshi and M. M. Trivedi, "Tactical driver behavior prediction and intent inference: A review," pp. 1892–1897, 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2011.

[10] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, I. Katunobu, K. Takeda, and F. Itakura, "Driver identification using driving behavior signals," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 3, pp. 1188–1194, 2006.

[11] S. Brook, R. Freeman, G. Rosala, F. Campean, and N. Dixon, "Ergonomic data measuring system for driver-pedals interaction," *SAE International*, pp. 1–8, 2009.

[12] F. Mueller and A. Wenzel, "Intelligent monitoring and adaptive competence assignment for driver and vehicle," pp. 1101–1106, IEEE Intelligent Vehicles Symposium, 2007.

[13] J. G. Rodrigues, F. Vieira, T. T. Vinhoza, J. Barros, and J. P. S. Cunha, “A non-intrusive multi-sensor system for characterizing driver behavior,” pp. 1620–1624, IEEE Annual Conference on Intelligent Transportation Systems, 2010.

[14] A. J. L. Oliveira, “Sistema de monitorização da condução de um automóvel,” *Master Dissertation*, 2012.

[15] Automation and Robotics, “Tactile sensors.” `http://www.southampton.ac.uk/~rmc1/robotics/artactile`

[16] Phidgets, “Load cell primer - phidgets support.” `http://www.phidgets.com/docs/Load_Cell_Primer`.

[17] Acroname, “Sharp infrared IR ranger comparison.” `http://www.acroname.com/articles/sharp.html`.

[18] E. Tutorials, “Wheatstone bridge circuit and theory of operation.” `http://www.electronics-tutorials.ws/blog/wheatstone-bridge.html`.

[19] Phidgets, “1101 user guide - phidgets support.” `http://www.phidgets.com/docs/1101_User_Guide`.

[20] Phidgets, “1121_0 - voltage divider.” `http://www.phidgets.com/products.php?product_id=1121`.

[21] Phidgets, “1121 user guide - phidgets support.” `http://www.phidgets.com/docs/1121_User_Guide`.

[22] Phidgets, “1101_0 - IR distance adapter.” `http://www.phidgets.com/products.php?category=2&product`

[23] Phidgets, “1018_2 - PhidgetInterfaceKit 8/8/8.” `http://www.phidgets.com/products.php?category=0&product_id=1018_2`.

[24] Microchip, “MCP23S17 - interface- serial peripherals.” `http://www.microchip.com/wwwproducts/Devices.aspx?product=MCP23S17`.

[25] Microchip, “MCP3008 - mixed signal- successive approximation register (SAR) A/D converters.” `http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010530`.

[26] Cypress, “CY7C64215: enCoRe III full - speed USB controller - cypress.” `http://www.cypress.com/?rID=14171`.

[27] Arduino, “Arduino board nano.” `http://arduino.cc/en/Main/ArduinoBoardNano`.

[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system,” vol. 3, 2009.

[29] ROS.org, “ROS/Concepts - ROS wiki.” `http://wiki.ros.org/ROS/Concepts`.

[30] A. Krause, “Foundations of GTK+ development,” *Apress*, 2007.

[31] G. Software, “Glade - a user interface designer.” `https://glade.gnome.org/`.

[32] Google, “Google maps.” `https://maps.google.pt/`.