Universidade de Aveiro Departamento de Engenharia Mecânica
2014

Pedro Xavier de Sousa Mendes

Monitorização Inteligente da Atuação Humana nos Pedais de um Veículo

Intelligent Monitoring of Human Actuations on Vehicle Pedals

Universidade de Aveiro Departamento de Engenharia Mecânica
2014

Pedro Xavier de Sousa Mendes

Monitorização Inteligente da Atuação Humana nos Pedais de um Veículo

Intelligent Monitoring of Human Actuations on 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, Estagiário Pos-Doc no Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President	**Prof. Doutora Margarida Isabel Cabrita Marques Coelho** Professora Auxiliar da Universidade de Aveiro
Vogais / Committee	**Prof. Doutor José Luis Costa Pinto de Azevedo** Professor Auxiliar da Universidade de Aveiro
	Prof. Doutor Vítor Manuel Ferreira dos Santos Professor Associado da Universidade de Aveiro (orientador)

Agradecimentos / Acknowledgements

Aqui ficam os meus sinceros e profundos agradecimentos,
Ao Professor Vitor Santos e ao Doutor Ricardo Pascoal, pelo incentivo, apoio e constante motivação. Por todas as sextas de manhã, que me ajudaram a manter o rumo!

A todos os membros do LAR, foi um prazer enorme trabalhar convosco, em especial ao Jorge Almeida e ao João Torrão, por todo o tempo e paciência dedicada aos nossos projectos.

Ao Engenheiro António Festas pela disponibilidade e por toda a ajuda que proporcionou ao longo destes ultimos anos, e ao Sr. Júlio Gonçalves pela ajuda no devenvolvimento das placas em PCB.

Aos meus pais e irmã por todo o apoio que me deram durante todos estes anos, e por me proporcionarem tudo de bom e de melhor.

A todos os que conheci nesta Academia e que me proporcionaram momentos fantásticos e memórias inesquéciveis.

Aos Yof's e N-Poison, mafaka! Por tudo e mais alguma coisa...! E livrai-nos do roch!

A ti... sempre! ♥

Keywords

Intelligent monitoring; vehicle pedals; driver actuation; real-time monitoring; vehicle safety.

Abstract

Nowadays, there is a huge socio-economic effort to reduce driving accidents. To do so, there are many safety methods and emphasis has been given to methods based on computer vision techniques. However, most of these techniques require high computational power and economical cost. As an alternative there are other techniques that exploit the vehicle-driver interfaces, such as the vehicle pedals. Considering the pedals as a powerful tool to monitor the driver actions, a system to monitor the driver actuations in automobile vehicle pedals was developed. In doing so, several possible conflicting requirements were taking into account, such as: low cost production, real-time monitoring and accuracy, robustness, non-invasiveness, flexibility and easy of use. To evaluate if the developed system fulfilled these requirements, tests were carried out under both stationary and real driving conditions. Overall, the system developed presents the necessary requirements and was able to successful monitor the driver actuation on automobile vehicle pedals.

Palavras-chave

Monitorização inteligente; pedais de automóvel; actuação do condutor; monitorização em tempo real; segurança automóvel.

Resumo

Atualmente, existe um enorme esforço sócio-económico com o intuito de reduzir os acidentes de viação. Para tal, existem vários métodos de segurança e especial ênfase tem sido atribuído a métodos baseados em técnicas de visão computacional. No entanto, a maior parte destas técnicas requer um elevado processamento computacional e elevados custos financeiros. Como alternativa, existem outras técnicas que exploram a interface do veículo-condutor, tal como os pedais do automóvel. Considerando os pedais uma ferramenta poderosa para monitorizar a atuação do condutor foi desenvolvido um sistema para monitorizar a actuação humana nos pedais de um veículo. Para tal, várias características essencias foram tidas em conta aquando do desenvolvimento do sistema, tais como: baixo-custo de produção, análise em tempo-real e com precisão, robustez, não invasividade, flexibilidade e facilidade de utilização. Para avaliar se o sistema desenvolvido cumpria estes requisitos, foram efetuados testes em condições estacionárias e condições reais de condução. No geral, o sistema desenvolvido revelou ter os requisitos necessários e ser capaz de monitorizar a atuação humana nos pedais de um veículo automóvel com sucesso.

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. This has led to a huge socio-economic effort to reduce driving accidents, which, according to the Eurostat [1], is one of the leading causes of death in Europe.

Safety is also one of the concerns which motivates 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 and has become 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 towards 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 positioning, an odometry system that measures the velocity and several computing units responsible for reading all the sensors and controlling the car.

Figure 1.1: *AtlasCAR [2].*

Despite the fact that there are many safety systems, 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 with several requirements such as the following [3]:

- Low cost of ownership, since an average automobile costumer may not be willing to pay for equipment which significantly increases the final cost;
- Real-time and accurate monitoring, to ensure timely detection 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;
- 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 automobile brand and model.

In order to fulfil all of these criteria, the intelligent monitoring of human actuations 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 human actuation. Because the vehicle pedals are a meaningful source of data output of the driver patterns, these were considered a powerful tool in monitoring driver activity. Hence, this project was entitled "Intelligent Monitoring of Human Actuations on Automobile 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 Actuations on Automobile Vehicle Pedals" is the creation and development of a system able to measure the driver actuation in the vehicle pedals. This system was designed to work in an interactive way, so that the interactions between the pedals and the driver would be measured and simultaneously displayed or stored and in an intelligent way, so that the calibration is smart enough to detect equipment or driving problems.

The interactions between the driver and the pedal may be defined by 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 requirements, 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 into three main parts:

– Development of an easy-to-use and non-invasive hardware, to collect data from the pedals;

– Design 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 incredibly 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 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 another 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 which emit a warning signal alerting the driver that the seat-belt is not fastened. In addition, electronic stability control, a system that reduces loss of vehicle control, can help in critical situations such as bad manoeuvres. To prevent a vehicle from being started if the driver is under the influence of alcohol, vehicles can be equipped with alcohol interlock systems [5–8].

Regarding driver 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 altered states. Seat belts are the easiest and cheapest way to avoid injury in a car crash. Regarding the speeding, which 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 1.2). Inattention deficits are predominant risk factors in driving and consist of three basic types [3]:

– Visual: factors that distract drivers causing eye deviation from the road, such as checking the mobilephone, GPS, etc;

– Cognitive: factors that make drivers mentally inattentive or absent, like stress, worries or deep thoughts;

– Manual: factors that cause drivers to take their hands off the steering wheel, for example, picking something that fell on the floor, smoking, etc.

Even though there are vehicle- and infrastructure-safety systems available aimed to diminish the severity of accident consequences, nowadays, the desired solution is to have a system that works by detecting high risk of accident and preventing accidents, instead of diminishing their consequences [4]. Monitoring driver performance has been seen as a crucial method to prevent accidents.

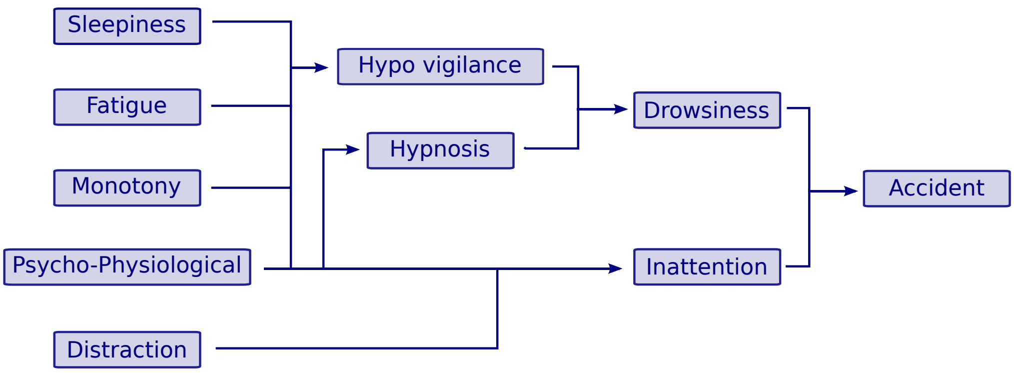


Figure 1.2: Factors that affect driver performance [3].

1.3.2 Monitoring Human Performance

Some driver behaviours that impair safe 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 risky 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 determine 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, based on previously collected data, is one of the few examples [14].

A driver performance monitoring system should be low cost, non-invasive, easy-to-use, flexible, and real-time accurate [3]. However, the few commercially available systems for monitoring the pedals usually lack modularity and flexibility. A very flexible and easy-to-use system to monitor the pedals, which 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.

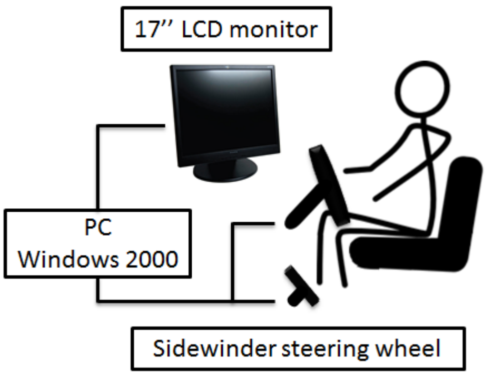


Figure 1.3: Driving simulator to collect behaviour signals [10].

1.3.3 Intelligent Monitoring

With an intelligent driver performance monitoring system, it could be possible to collect and analyse several aspects of the vehicle continuously, while driving. By analysing and recording the braking situations, braking patterns could be defined overtime. With this it would be possible to infer from a braking situation, if the system would require more or less force to be applied on the brake pedal to brake safely, and the driver could be warned accordingly. This braking assistance could lead not only to a more safely braking but also to a more efficiency braking situations. Because of the continuous degradation of the braking system and other external factors, such as road slopes, weather, etc, braking patterns change continuously. Thus, re-defining them along the driving so that the driver could always have an assisted braking would very likely increase the efficiency of the braking system of the vehicle.

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 only force sensitive resistors (FSRs) to measure the applied forces on pedals; FSRs are often used due to their small size and the applicability as sensing element 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 the mentioned requirements. Therefore, and because both sensors offer complementary characteristics, 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 relative to the car floor using another type of sensor. For the purpose of measuring the pedal displacement, infrared (IR) sensors were selected in order to work perfectly for the dimensions of the pedal travel. Here, three types of sensors were used on this system: FSRs, LCs and 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 kgf. For higher forces, the sensor saturates without getting permanently deformed [15].

The LCs are sensors composed of strain gauges, usually forming a Wheatstone bridge, applied over a structure element that is 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 require 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 kgf [16].

The IR distance sensors are built in small packages and have low current consumptions

(50 mA @ 4.5–5.5 V). There are mainly two types of IRs: analog, that provide analog information about the distance to an object and digital, that provide a digital (ON/OFF) indication if an object is below a predefined distance or further from it. The analogue version of the sensors have output values as function of the measured distances. For this system, due to the small pedal displacement, have been selected the Sharp 2D120X IR [17] with the lowest range available, that was from 4 to 30 cm.

In order to read the data from all the three types of sensors, they were connected to a board which is then connected to the computer. The board used on this system was the PhidgetInterface-kit 8/8/8 from Phidgets. An Arduino Nano v3.1 was also used in preliminary tests. The Interface-kit provides 8 digital inputs and 8 analog inputs and outputs, making it possible to connect all the sensors. However, the PhidgetInterface-kit 8/8/8 cannot provide the required signal conditioning so, in order to do that it was necessary to use signal conditioning boards, such as IR adapters and FSR adapters provided from Phidgets and custom LC adapters.

2.1.1 Force Sensitive Resistor

The force sensitive resistors (FSR) are flat, robust and polymer thick film sensors that have variable resistance between their pins depending upon the applied force (Figure 2.1). The sensing film is composed by electrical conducting and non-conducting particles suspended in matrix. When force is applied, those non-conducting particles get in contact with the conducting electrodes, thus changing the electrical resistance of sensor [15]. Their working principle is: the higher the force the lower is the resistance.

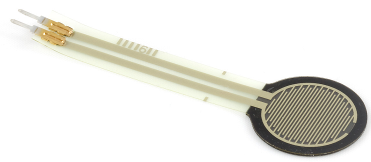


Figure 2.1: Interlink Electronics 0.5" Circular 20N FSR [15].

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, once released the FSR output value will come back to approximately 95% of its initial value almost instantly, and then drift the final 5% over a few seconds [15]. Hence, these sensors are not the best option when setting up a system that relies greatly on accuracy and repeatability. However, in combination with other sensors, such as load cells, they can be very useful. For instance, even though the FSRs used on this system have the maximum capacity of 20 N, because the driver applies more force on the pedals these sensors will quickly saturate so they can be used to detect smaller forces, such as, for example, if the driver is touching the pedal.

2.1.2 Force Load Cell Sensor

A Load cell (LC) is a sensing module that converts force into an electrical signal (Figure 2.2). The LC has strain gauges mounted in its structure and are collectively deformed when force is applied, in this case 0 to 50 Kg; the strain gauges are responsible for converting the deformation on them into an electrical signal. The electrical signal output is very low, typically in the order of 1 millivolts/volt, thus it is required the use of an instrumentation amplifier [16].

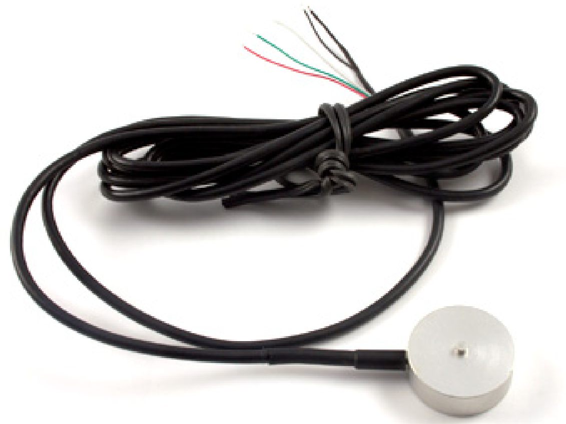


Figure 2.2: Button Load Cell (0–50kg) [16].

Most designs of load cells use 4 strain gauges in a Wheatstone bridge configuration, but there are also other types that only use 2 strain gauges, often called half bridges, and one strain gauge (quarter bridge) [16].

A Wheatstone bridge is an electrical circuit originally used to determine the value of an unknown resistance by balancing two nodes of the bridge circuit. It has two input terminals and two output terminals consisting of four resistors represented in a diamond-like arrangement. In the case of the load cells, these resistors are strain gauges [18].

The load cell electrical output can be converted to force or weight using equation (2.1) [16]:

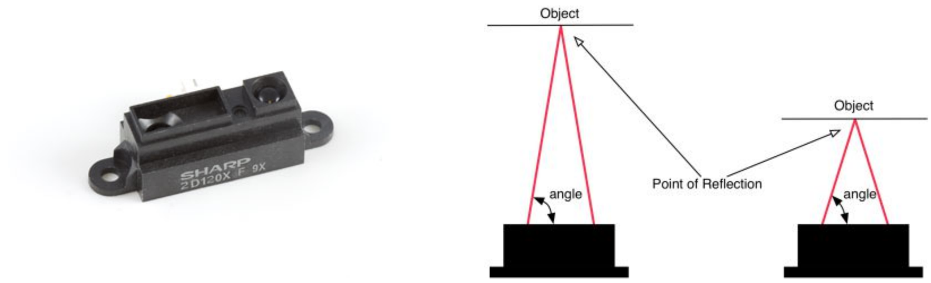
$$Force = K \times (Measurement - Offset) \tag{2.1}$$

Where K is the gain value that will change depending on which are the units being used and mounting arrangements. The offset value varies between individual load cells, so it is necessary to measure the offset for each sensor. By measuring the output of the load cell at rest on a flat surface with no force applied on it, it is possible to find the offset. Once the offset is found and, a known force is exerted on the sensor, it is possible to solve equation 2.1 for K.

This calibration is important when the real value of the force is necessary. However, due to the fact that in this monitoring system the data range from the sensors will be different for each user, instead of using equation (2.1) a minimum and a maximum value is used to calibrate the sensors, the minimum value is measured when no force is applied on the pedal and the maximum value when the pedal is pressed till the end.

2.1.3 Infrared Distance Sensor

The Sharp infrared (IR) distance sensor provides information about the distance to an object (Figure 2.3a). These sensors work by the process of triangulation. A pulse of infrared light is emitted and then reflected back, or not reflected at all. When the light returns it forms an angle that is dependent on the distance of the reflecting object. Triangulation works by detecting this reflected beam angle; by knowing the angle, the distance can then be determined (Figure 2.3b) [17].



(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 instead of in approach. This particular feature could represent a problem for distances lower than 4 cm. In order to solve this issue, the possibility to adjust the IR sensor orientation was added in the system.

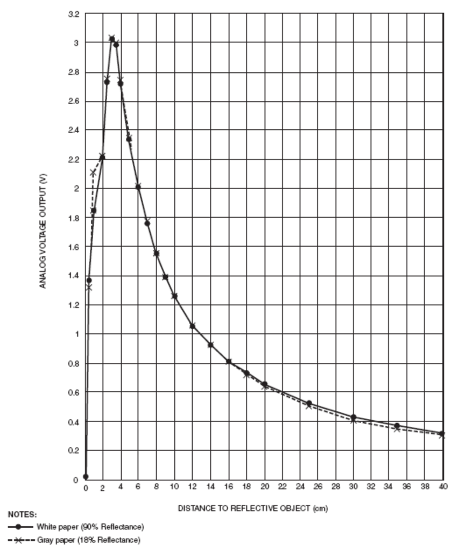


Figure 2.4: Sharp 2D120X IR (4–30 cm) output voltage vs distance [17].

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 ADC output values between 80 and 530, 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, by using the standard equation on this system, the IR sensors provide good results, 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, an enclosure able to lodge 2 LCs, 2 FSRs and 2 IR sensors was developed. When designing the enclosure, special attention was given to the dimensions, because it cannot become 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).

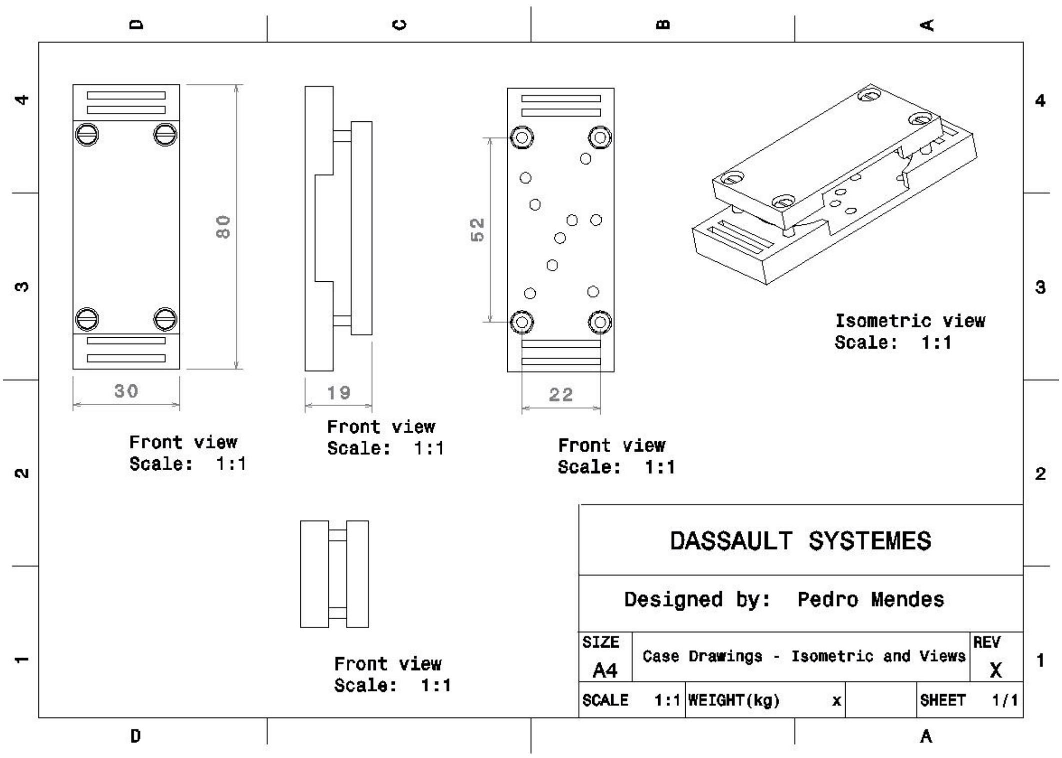
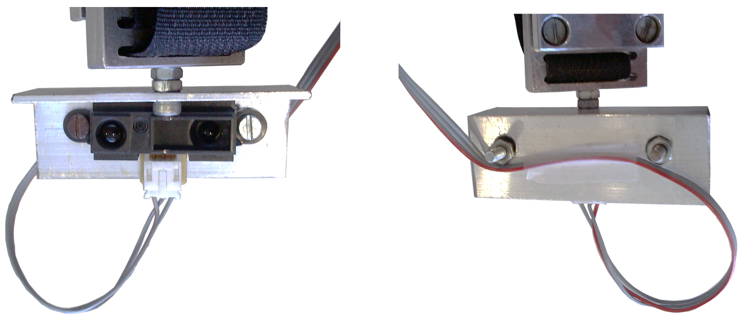


Figure 2.5: Case mechanical drawings.

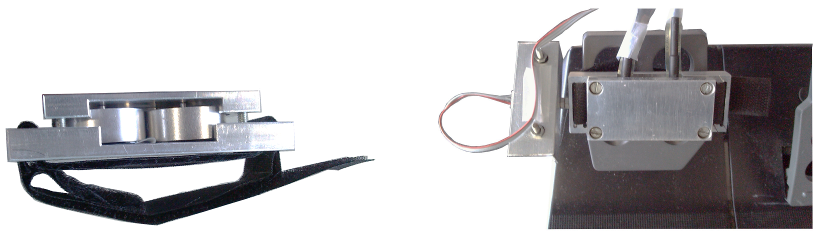
The load cells and force sensitive resistors were meant to be fixed on the case, whereas the infrared sensors were planned to be installed on the sides of the case, with the possibility of adjusting their orientation shown in Figure 2.6. The reduced dimensions of the FSRs made it possible to install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

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 the driver on the lid would be transferred through the fastening system instead of being transferred60 through the LCs and consequently the FSRs, as supposed. To accomplish this, it was necessary to use a clamping system by adding both threaded holes on the bottom and through holes on the lid. Both parts were fastened using screws so that the sensors would be just touching not saturated. In addition, for a quick installation/uninstallation of the case on top of the pedals, a fixing system based on Velcro strips was adopted and is shown in Figure 2.7. The exploded view is shown in Figure 2.8.



(a) (b)

Figure 2.6: a) Adjustment system - Front view. b) Adjustment system - Rear view.



(a) (b)

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

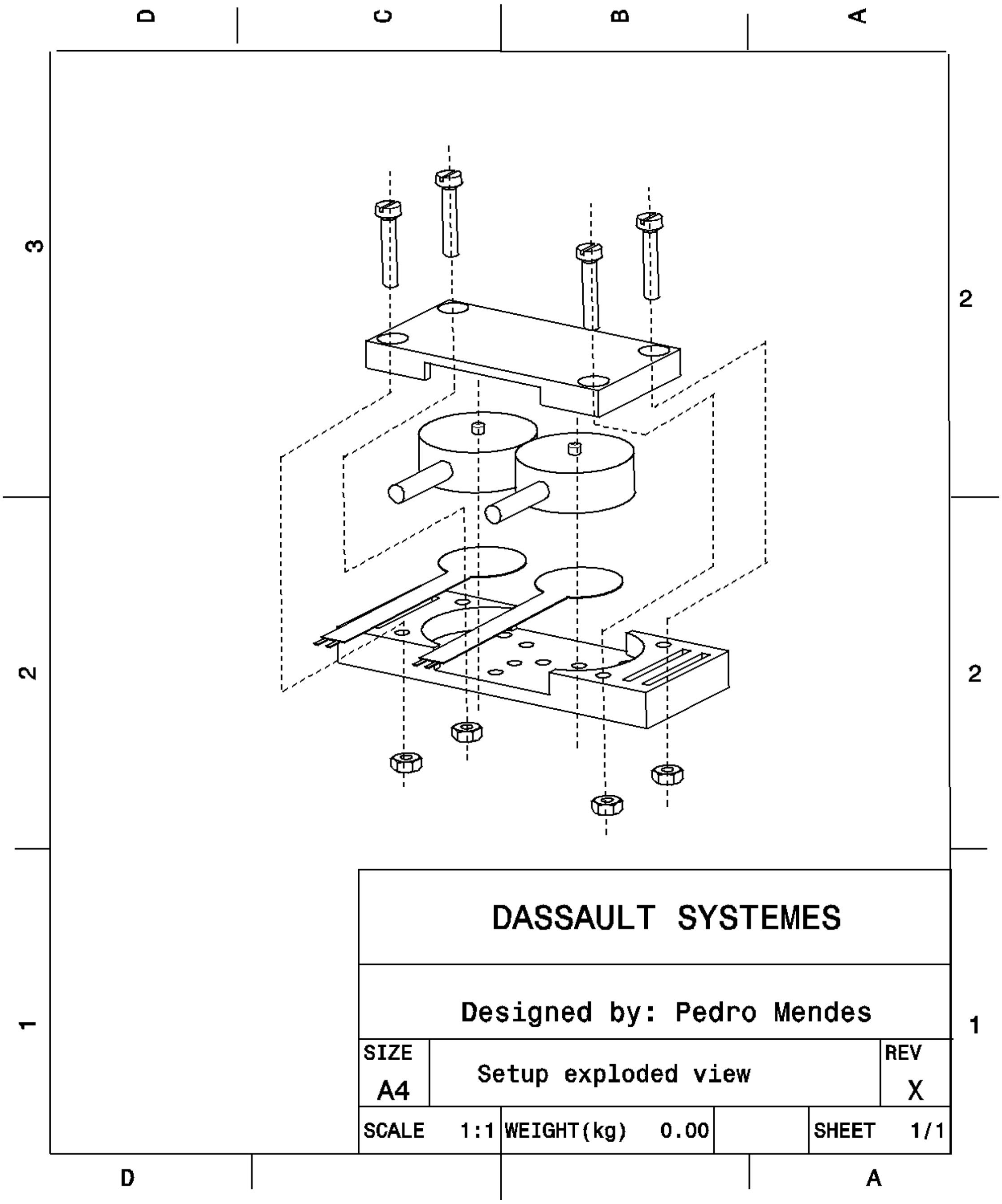
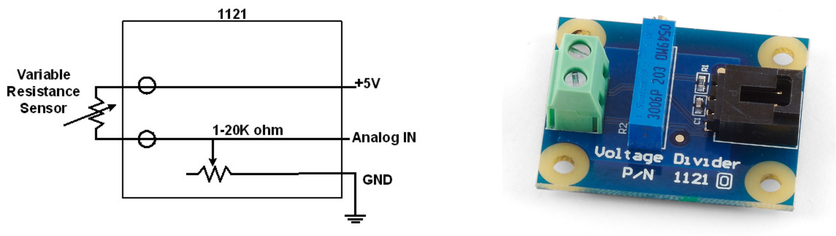


Figure 2.8: Exploded view of the case.

2.3 Electronic Devices

As mentioned, the output values of the sensors need to be conditioned, in this case filtered 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 another custom made board for the force load cell (LC).

The board used for the FSR signal conditioning was the Voltage Divider (Figure 2.9), which provides an interface to resistance sensors such as: FSRs, light sensors, heat sensors and bend sensors [20]. This board has a high precision potentiometer that allows the adjustment of the resistors in the divider. The two resistors form a voltage divider who's output is then connected to the analog input of the interface board. In order to make sure that the sensor value stays within an expectable range, it is possible to optimize the voltage divider; the voltage divider resistance can be adjusted by simply turning the screw on the potentiometer; clockwise to decrease the resistance, and counterclockwise to increase it [21].



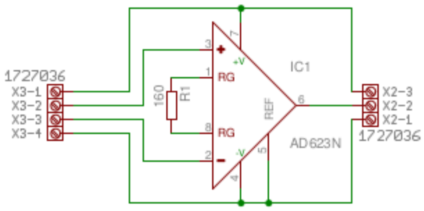
(a) (b)

Figure 2.9: a) Force Load Cell adapter circuit scheme. b) Force Sensitive Resistor Adapter [20].

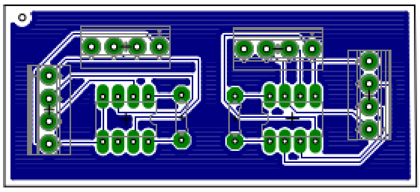
The LCs are sensors that require an instrumentation amplifier (in-amp) to amplify their output value, which is in the range of millivolts. The reason to use an in-amp is for improved common-mode rejection. Therefore, a circuit board was developed to mount the sensors, amplify the signals and connect to the PhidgetInterfaceKit 8/8/8 responsible for the data acquisition (Figure 2.10). 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):

$$G = 1 + \frac{100000}{Rg} \tag{2.3}$$

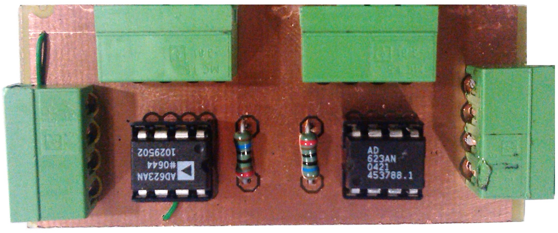
Where the Rg is the resistor connected between pins 1 and 8 of the in-amp. The resistor installed in the board has 160 Ω; placing this value on the equation it is possible to calculate the gain of the board that is approximately 600.



(a)



(b)



(c)

Figure 2.10: a) Force Load Cell adapter circuit scheme. b) Force Load Cell adapter PCB circuit. c) Force Load Cell Adapter.

For the IR distance sensors signal conditioning, the IR Distance Adapter board was used (Figure 2.11). This board provides an interface to an analog input on the PhidgetInterface-kit 8/8/8 board, mentioned before, and provides a conditioned power source for the connected sensor. Although the specified current consumption of the sensor 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 [22].

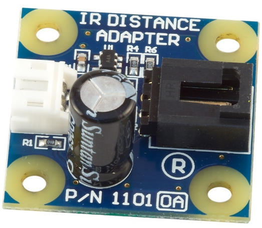


Figure 2.11: Infrared Distance Sensor Adapter [22].

2.4 Acquisition System

As mentioned, for the data acquisition, the PhidgetInterfaceKit 8/8/8 board was used and is shown in Figure 2.12 [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 corresponding register and then all registers can be read by the system master [24]. The board has a MCP3008 which is an 8 channel 10-bit analogic to digital converter (ADC) [25]. Finally, this board has a CY7C64215 that is a full-speed USB controller. This device is responsible for interfacing between SPI and USB [26], which is the protocol used to connect the board to the computer.

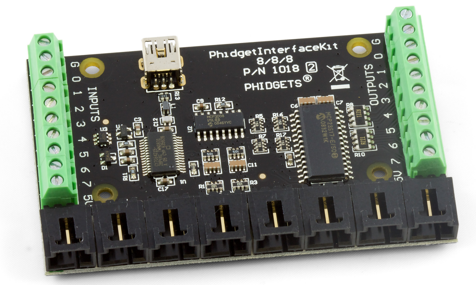


Figure 2.12: PhidgetInterfaceKit 8/8/8 Acquisition Board from Phidgets [23].

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 2.13) [27]. The Arduino Nano v3.1 is a single-board microcontroller, with smaller dimensions, more flexible and less expensive than the PhidgetInterfaceKit 8/8/8. Actually, as mentioned above the Arduino Nano was used on preliminary tests to read the data from the sensors, however, due to convenience reasons, the PhidgetInterfaceKit 8/8/8 was used; because the signal conditioning boards were also from Phidgets and so the assembly was easier and the system more robust for a first prototype.

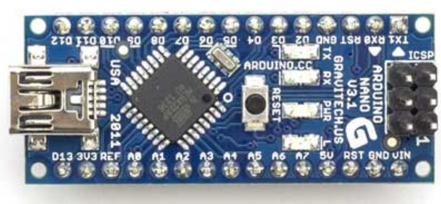


Figure 2.13: Arduino Nano v3.1 [27].

However, for a final and large scale production approach, the use of custom electronic devices or existing commercial off-the-shelf, such as Arduino or a single-board computer

(SBC) like the Raspberry Pi [28], would represent a more feasible solution due to their lower costs and firmware flexibility. Nonetheless, to use these solutions, it would be required the use of additional electronic devices to assure proper signal conditioning.

For instance, when the electronic devices are far away from the sensors, to receive the output from the sensors without any losses it is necessary to convert the output signal of the sensors from voltage to current (V-to-I). Because the current signals are exactly equal in magnitude throughout the circuit a V-to-I converter can be used to prevent signal losses. To do so, several aspects of the setup should be modified: 1) the power source that supplies the sensors must be isolated or different from the power source that supplies the acquisition board, otherwise it will not be possible to read the signal from the sensors, 2) the output signal from the sensors must be connected to the V-to-I converter, 3) after the V-to-I conversion and before the signal reaches the acquisition board, the current must be converted back to voltage by adding a resistor to the circuit.

In the case of using a Raspberry Pi, it would not be necessary to use a computer to connect all the system. Also, by using the Raspberry Pi, it is also possible to create a network connection, such as Ethernet or Wifi. With this network connection it would be possible to communicate with the system via remote, which could represent a great improvement in the system.

2.5 Setup

After the conclusion of the hardware and electronic devices, it was possible to assemble all of the components and create a setup to start monitoring the pedals (Figure 2.14). To do so, it was 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, responsible for supplying and for the communication with the board, launch the software interface and start monitoring.

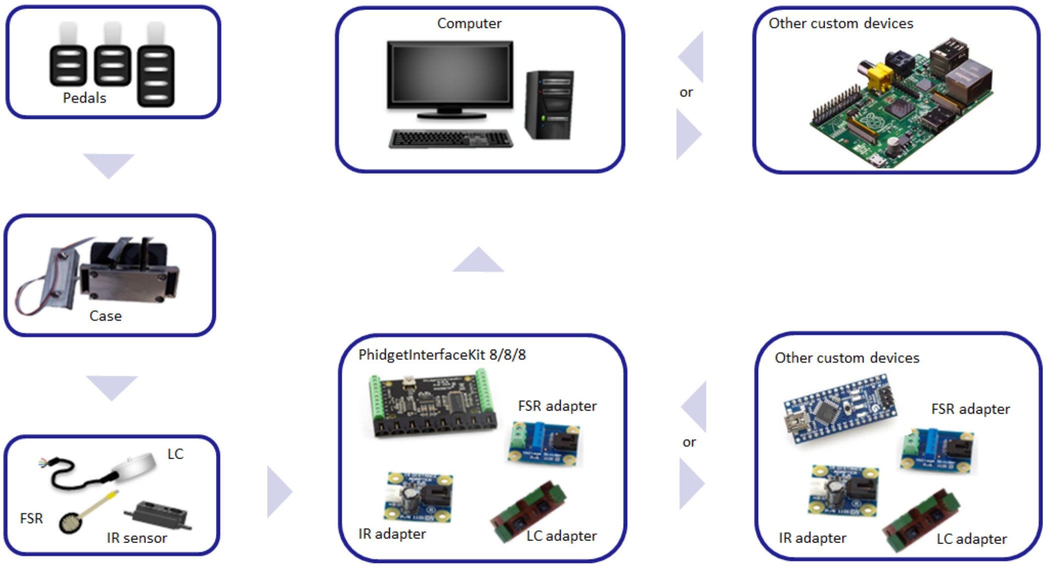


Figure 2.14: Setup scheme. In order to monitor the pedals, the case is mounted on them. The case has LCs,FSRs and IR sensors installed and these are connected to their adapters. The adapters connect to the PhidgetInterfacekit and than to the computer. It is possible to replace these adapters and the PhidgetInterfacekit for other custom devices. Also, it is possible to replace the computer for other device, such as a single board computer (SBC).

Chapter 3

Software Architecture and Applications

The second main objective of this project was the design 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

The software was developed under a Linux platform using C/C++ programming and 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 a useful software for designing interfaces. In addition, it was necessary to use proprietary 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 improves comprehension [29].

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 starts to receive messages from the node that is publishing. While communicating, both nodes are connected to a topic (Figure 3.1). This communication structure allows to publish simultaneous topics and these topics can be subscribed by multiple nodes. In this way, a module can fulfil multiple tasks and be easily integrated in a simple or multiple research project [30].

Another important aspect provided by ROS is the ability to store ROS data messages into *bags*. *Bags* are files which store all the requested published data, being considered

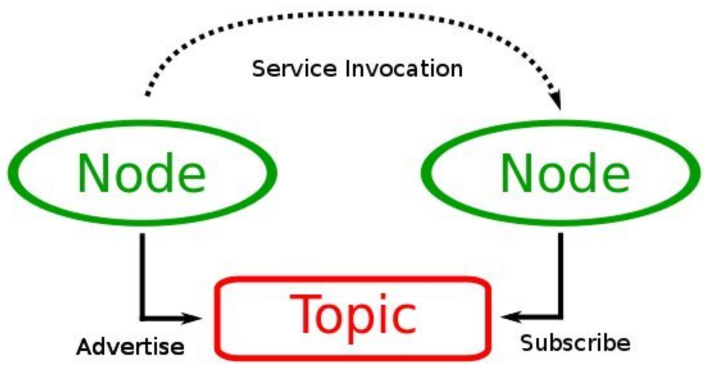


Figure 3.1: Basic ROS communication concept [30].

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* [30].

Due to the important aspects of ROS, the *Atlas Project* has been migrating every application to this environment.

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 [31]:

- 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+ widgets use the GObject hierarchy system, which is one of the most important aspects of GTK+ (Figure 3.2) [31].

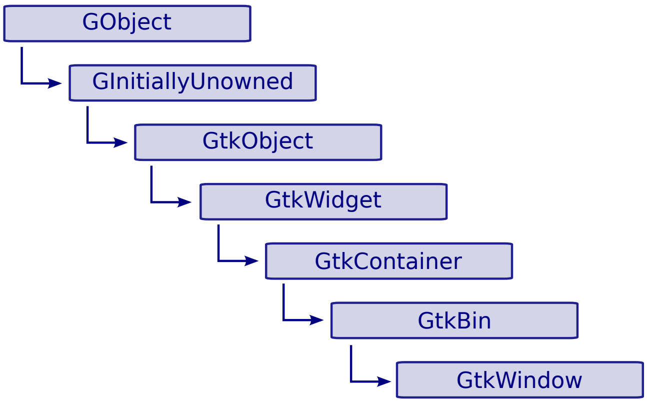


Figure 3.2: The widget hierarchy of GTK+ [31].

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 [32].

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, by defining a class on a source file named *PhidgetClass* and adding these functions to it, when the *start()* function is called, the connected PhidgetInterfaceKit 8/8/8 will be attached, becoming ready to communicate with the computer. After the board is attached it is possible to read the data from the sensors using the function *CPhidgetInterfaceKitget_getSensorValue(ID, sensor_index, &Sensor1)*. This function requires three parameters: 1) the ID of the board, which is known when the board is attached, 2) the sensor index (which ranges from 0 to 7), which are the ports where the sensors are connected in the board, and 3) a variable, for example *Sensor1*, where the sensor value is saved.

3.2 Interface

As mentioned, the program responsible for the interface was written in C/C++ and uses Glade and GTK+. It is responsible for the communication between the user and the system. This interface was intended to calibrate the sensors as well as to monitor the data received from them. To accomplish this, it was necessary to create a main source file responsible for two tasks: the interface and the communication between the program and the acquisition system (Figure 3.8).

The interface was designed using the Glade software and then loaded using the GTK+ in the main file. The interface is composed by three menus: 1) the *Main menu* shown in Figure 3.3, where the user an choose either to create a new calibration or to load an existing one, 2) the *New calibration menu* shown in Figure 3.4, where the user can create a new calibration and 3) the *Load calibration menu* shown in Figure 3.7, where the user can load an existing calibration and monitor the pedals.

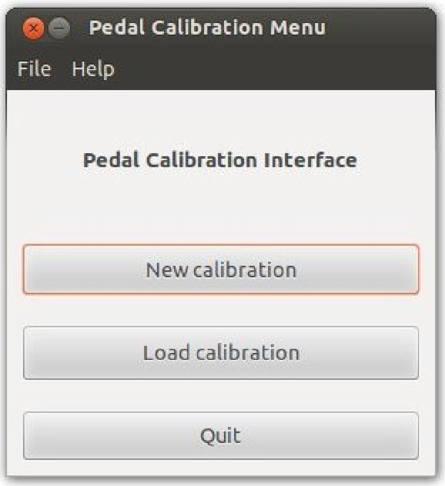
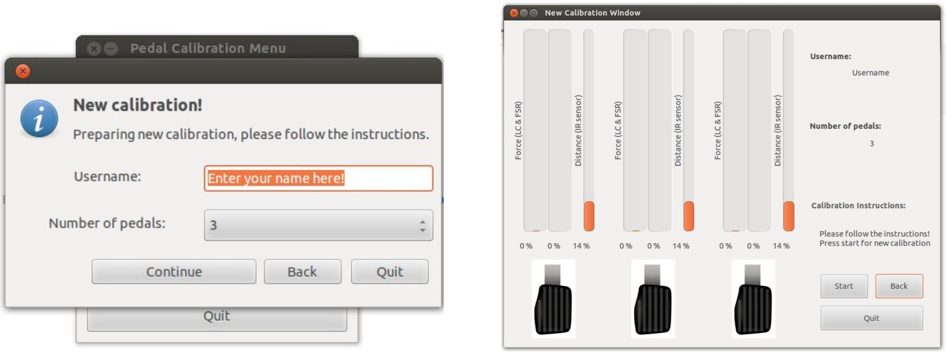


Figure 3.3: Interface main menu.

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 window emerges and the system is ready to create a new calibration. In order to do that, it is necessary to press the *Start* button and simply follow the instructions (Figure 3.5). In case of failure, mostly due to the wrong orientation of the infrared (IR) sensors, a function was added to test that condition and issue a warning if is detected. After completing the calibration process, a XML file is created and a save file dialog is open so the user can choose where to save the calibration file. This XML file is organized into categories: the main category is 1) the root element, which is the *calibration*, 2) the *calibration* has two children named *pedal*, 3) the *pedal* has 2 children, which are the *name* and the *sensor*, and finally 4) the *sensor* has 4 children that are *name*, *type*, *min* and *max*, as shown in Figure 3.6.



(a) Selection menu for user and pedals. (b) New calibration menu

Figure 3.4: Interface new calibration menu.

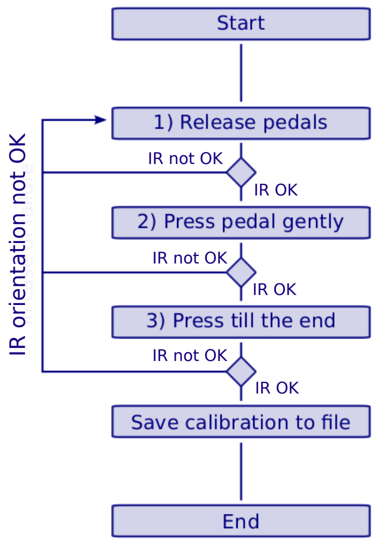


Figure 3.5: Calibration instructions scheme.

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. 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 was created inside this thread. Among this set of functions, two were responsible for the

```xml
<?xml version="1.0" encoding="utf-8"?>
<calibration user_name="calibration test">
    <pedal>
        <name>clutch</name>
        <sensor>
            <name>LC1</name>
            <type>LC</type>
            <min>5</min>
            <max>254</max>
        </sensor>
        <sensor>
            <name>FSR1</name>
            <type>FSR</type>
            <min>363</min>
            <max>954</max>
        </sensor>
        <sensor>
            <name>IR1</name>
            <type>IR</type>
            <min>0.2356472795497186</min>
            <max>0.4603550295857989</max>
        </sensor>
    </pedal>
    <pedal>
        <name>brake</name>
        <sensor>
            <name>LC2</name>
            <type>LC</type>
            <min>74</min>
            <max>114</max>
        </sensor>
        <sensor>
            <name>FSR2</name>
            <type>FSR</type>
            <min>0</min>
            <max>635</max>
        </sensor>
        <sensor>
            <name>IR2</name>
            <type>IR</type>
            <min>0</min>
            <max>0.01823607427055704</max>
        </sensor>
    </pedal>
</calibration>
```

Figure 3.6: Example of a XML calibration file.

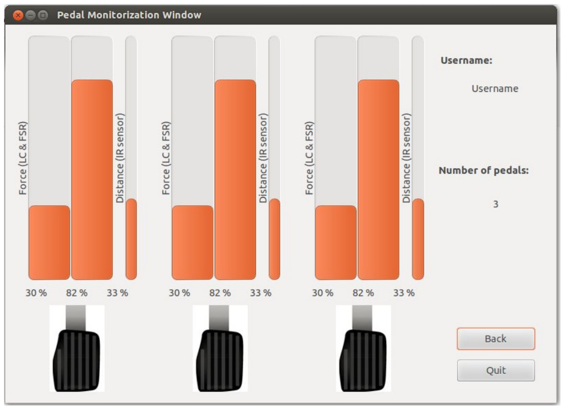


Figure 3.7: Interface load calibration menu.

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*.

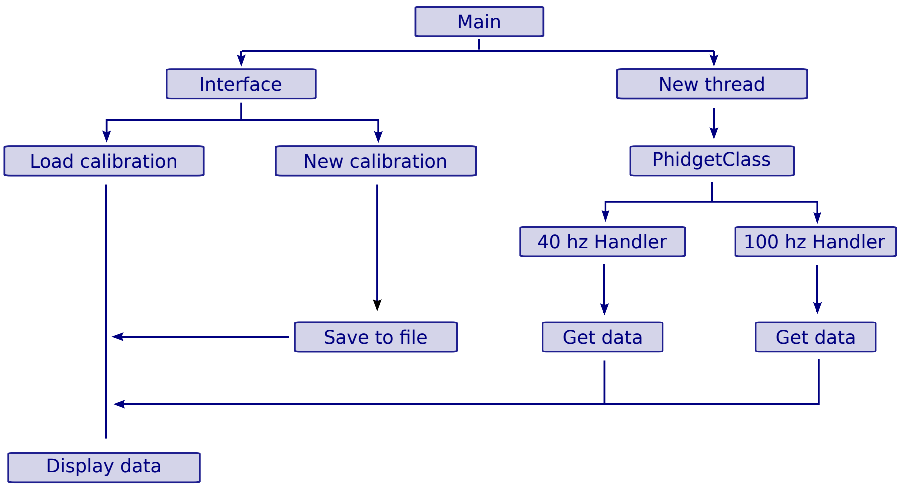


Figure 3.8: Code organization diagram.

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 wants 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 3.9).

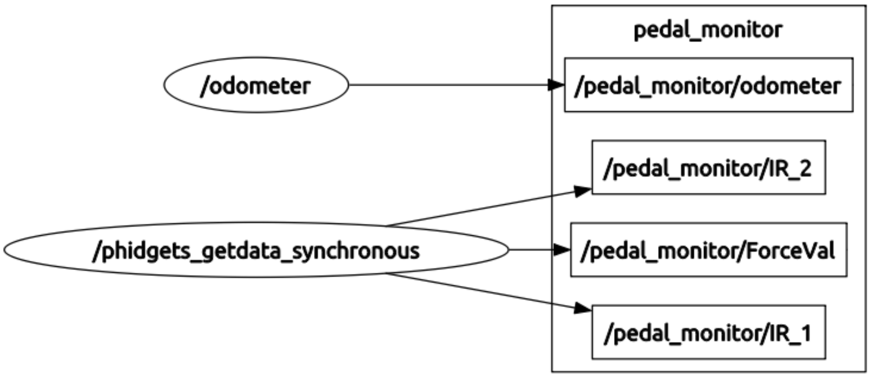


Figure 3.9: Published messages on ROS.

These messages can be displayed using the ROS rviz, which is a ROS package that allows the visualization of specific messages: range type for the IR messages and markers type for the SenVal messages. For instance, in Figure 3.10) displayed messages are presented: the green color represents the LC values, the blue color the FSR values and the red color the IR sensor values.

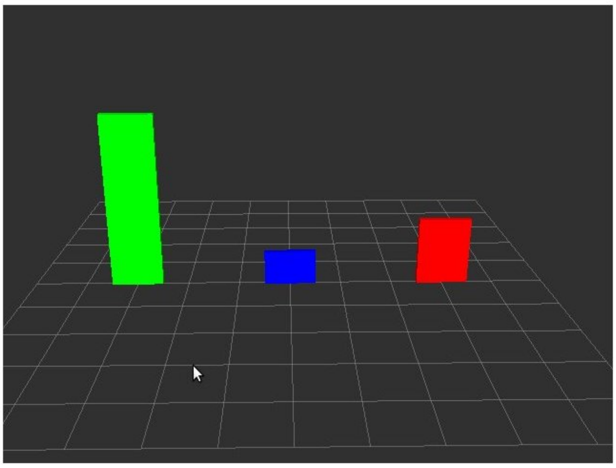


Figure 3.10: Published data displayed on Rviz.

In addition, it is also possible to launch several packages together with the one developed in this system. For instance, it would be interesting to also know the car speed in this system, hence, it is possible to launch an odometry package, in order to measure the car speed and rpm. In order to launch a package or several packages, it is necessary to create a *launch file*, which is XML configuration file where is it possible to specify which packages the user want to launch, among other options. An example of a launch file is shown in Figure 3.11.

```
<?xml version="1.0"?>
<launch>

    <remap from="phidgets_getdata_synchronous/IR_1" to="/pedal_monitor/IR_1"/>
    <remap from="phidgets_getdata_synchronous/IR_2" to="/pedal_monitor/IR_2"/>
    <remap from="phidgets_getdata_synchronous/ForceVal" to="/pedal_monitor/ForceVal"/>

    <node pkg="pedal_monitor" name="phidgets_getdata_synchronous" type="phidgets_getdata_synchronous">
        <param name="file_name" value="$(find pedal_monitor)/src/calibration_test.xml"/>
    </node>


    <!-- Launch the odometer node -->
    <remap from="odometer/status" to="/pedal_monitor/odometer"/>

    <node pkg="odometer" name="odometer" type="odometer_node">
        <param name="server_ip" value="10.0.0.31"/>
        <param name="server_port" value="120" type="string"/>
    </node>


    <!-- This is a launch file for recording the topics for my thesys  -->

    <node pkg="rosbag" type="record" name="rosbag_record"
    args="record -o /media/Data/Data_Record/newtests/bag1

    /pedal_monitor/ForceVal
    /pedal_monitor/IR_1
    /pedal_monitor/IR_2
    /pedal_monitor/odometer
    "/>

</launch>
```

Figure 3.11: Launch file example used to record the data from the monitoring system.

3.4 Future developments

As mentioned before, it is possible to use other electronic devices as acquisition boards, such as an Arduino or a Raspberry Pi, for a final solution approach.

In case of using an Arduino, instead of Phidgets libraries, it would be necessary to develop a code to read the data from the sensors. In order to do that, the code should be able to read the Arduino analogical pins where the sensors are connected to. The Arduino code is implemented in C/C++ and there are many libraries available that facilitate the code development; thus, to read an analogical pin it is just required to use a simple function — *analogRead()*.

Also, both the developed programs, the interface and the monitoring and publishing, would require some changes in order to communicate with the Arduino instead of the Phidgets board. The communication with the Arduino is much easier, due to the fact that the Arduino communicates by serial port. To access the data from the Arduino, it is necessary to replace the Phidget class to a serial read function. To implement this serial read function, it is possible to use the boost/asio library that provides the *asio::read()* function, which reads the data from an established serial connection.

In addition, by adding a bluetooth module to the Arduino, it would be possible to communicate with the system using a device with bluetooth, such as a smartphone or tablet. With this module and an user interface application for Android, developed using the Android software development kit, it would be possible to interact with the system using a smartphone or a tablet, instead of a computer.

In case of using a Raspberry Pi, it would be possible to create a WiFi network to communicate with the user gadgets, such as smartphone, tablet, laptop or other devices supporting WiFi communication. With the WiFi connection, and developing an user interface using HTML and PHP languages, the user interface would be accessible from every platform that has a web browser. This approach would be an asset to make the system even more flexible and robust.

Chapter 4

Results and Discussion

As mentioned on the first chapter, this project aims the development of a system able to comply several issues of concern, such as: flexibility, easy-to-use, non-invasiveness, robustness and real-time monitoring. To understand if the developed system fulfils these requirements, several experiments were performed. These experiments were conducted in two states: stationary driving and real driving situations.

4.1 Stationary Driving Situations

In stationary driving conditions, it was possible to evaluate the fastening system in the pedals as well as the calibration process. To understand if the fastening system was working properly, the system was installed in a variety of pedals of several vehicles while they were turned off.

In one of the tests, the system was installed in the clutch and brake pedals of a *Peugeot 207* (Figure 4.1). The system was easily installed and the calibrations created for each pedal were intuitive and fast. The few errors while calibrating the brake pedal, were simply corrected by readjusting the orientation of the infrared (IR) sensors.

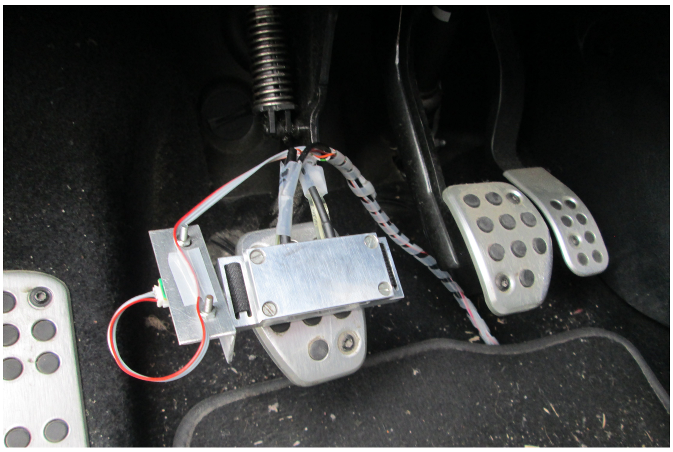


Figure 4.1: Setup installed on a Peugeot 207.

The system was also installed in the clutch and brake pedals of the *AtlasCAR* (Figure 4.2). In this case the installation was easier than in the *Peugeot 207*, mainly due to the different materials of the pedals: metal for the *Peugeot 207* and rubber for the *AtlasCAR*.

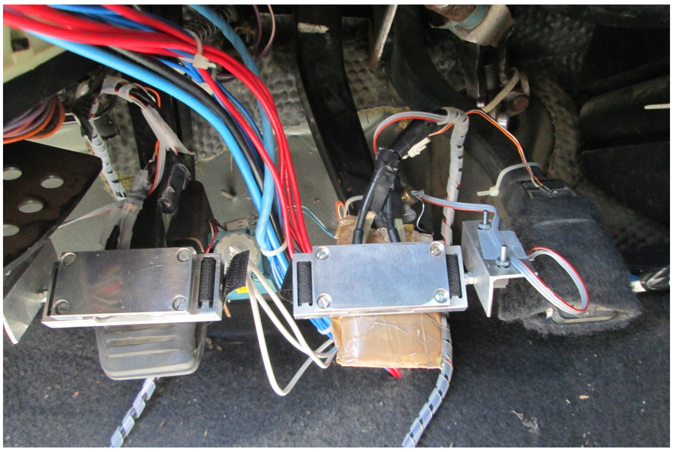


Figure 4.2: Setup installed on the AtlasCAR.

For all the tested pedals of vehicles, the system generally revealed an effortless and quick installation. Thus, it was possible to validate the flexibility and ease of use of the developed system to monitor the vehicle pedals. Regarding the calibration, the process was relatively fast and easy to follow. There were a few errors while calibrating due to the wrong orientation of the IR sensors, however, they were automatically identified by the interface and the driver was immediately warned to correct it.

4.2 Real Driving Situations

In real driving conditions, it was possible to test the non-invasiveness and robustness of the system. Importantly, it was also possible to test if the pedals could be monitored in real-time. As for the stationary driving conditions, these tests were also performed in several vehicles.

In order to validate those conditions, the system was installed and tested in two different driving situations:

– The same driving route and vehicle for different drivers;

– Different velocities, stopping the car only using the brake pedal.

4.2.1 Preliminary test

Before performing the tests mentioned above, a preliminary test was made in order to observe the system capabilities. For this test, the *AtlasCAR* was driven through the route shown in Figure 4.3. Because the *AtlasCAR* has an odometer system installed, it was also possible to obtain the velocity overtime. This is an advantage because it is possible to correlate the action on the pedals with its consequences in the velocity.

With this preliminary test, it was possible to conclude about several characteristics of the system. Firstly, due to the fact that the system was not a nuisance to the driver, it was possible to confirm its non-invasiveness. Moreover, while driving, it was possible

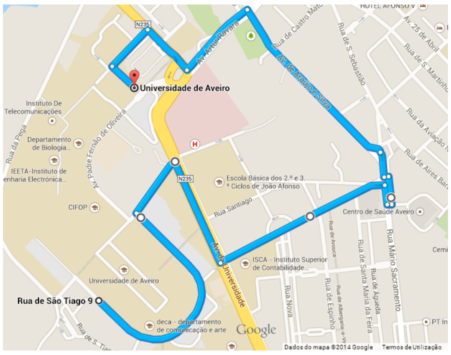


Figure 4.3: Route for the preliminary test [33].

to test the robustness of the system. The system showed to be very robust in terms of collecting data, so that it was possible to infer the real actuation on the pedals just by looking at the displayed data on the monitoring interface. However, in terms of the fixation on the pedals it was not much effective. This lack of effectiveness was partially due to the non linear geometry of the pedal shaft, which makes it quite difficult to secure the system so it is firmly fixed. In order to improve this, it is possible to use other fixing systems, such as a belt.

Finally, the velocity and force values were normalized to their maximum values. By comparing the data collected from the brake with data from the clutch (Figure 4.4) it was possible to analyse the behaviour of both pedals.

Also, to understand if there was a correlation between the velocity and the pedal forces, the kinetic energy of the car and the braking energy were taking into account. These energies were calculated as shown in Equation 4.1.

$$\Delta E_{braking} = |\Delta E_c| \Leftrightarrow \int_0^t F_{brake} \cdot v_{car} \cdot dt = \frac{1}{2} \cdot m_{car} \cdot \left|(v_f^2 - v_i^2)\right| \tag{4.1}$$

By assuming that the force applied on the brake pedal is proportional to the braking force, as shown in Equation 4.2, it was possible to find the ratio between the kinetic energy of the car and the braking energy using the force applied on the brake pedal, shown in Equation 4.3.

$$F_{braking} = K \cdot F_{pedal} \tag{4.2}$$

$$K = \frac{\frac{1}{2} \cdot m_{car} \cdot \left|(v_f^2 - v_i^2)\right|}{\int_0^t F_{pedal} \cdot v_{car} \cdot dt} \tag{4.3}$$

Due to the high amount of information collected in this test, only a selected part was analysed, between 100 and 250 seconds, shown in Table 4.1.

By observing the spikes on the graph, it seems that generally the clutch is pressed much more times than the brake. Moreover, it was possible to find that the forces on the brake pedal and the velocity of the car are on average inversely correlated, meaning that generally when the force on the brake pedal increases, the velocity decreases. These values are not significantly correlated, maybe due to signal noise and other external variables,

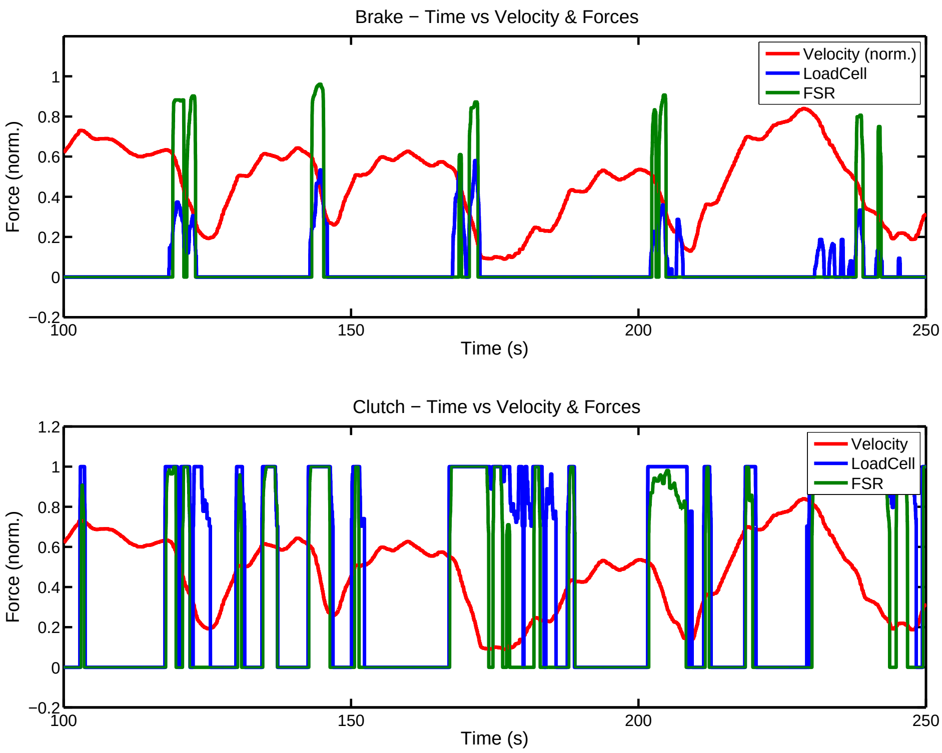


Figure 4.4: Data of the preliminary test from brake (top) and clutch (bottom) pedals, with a median filter applied, between 100s and 250s.

v_i	v_f	$v_f^2 - v_i^2$	$\int_0^t F_{pedal}.v_{car}.dt$	K
0.629	0.257	0.330	0.530	0.622
0.604	0.286	0.283	0.491	0.577
0.542	0.345	0.175	0.365	0.480
0.326	0.116	0.093	0.186	0.501
0.515	0.283	0.185	0.263	0.705
0.255	0.240	0.007	0.003	2.8717
0.223	0.154	0.026	0.048	0.544
0.805	0.714	0.138	0.161	0.857
0.681	0.624	0.075	0.084	0.891
0.573	0.549	0.027	0.041	0.652
0.522	0.502	0.020	0.020	1.016
0.485	0.362	0.104	0.156	0.669
0.315	0.269	0.027	0.031	0.871

Table 4.1: Data of the preliminary test from brake pedal, between 100s and 250s. For each braking situation: initial and final velocities, the kinetic energy, the energy of the pedals and the ratio K between the kinetic energy and the energy of the pedals.

such as road slopes, engine braking, etc. By looking at the graph, it is possible to see that the Force Sensitive Resistors (FSRs) values reach saturation levels much faster than the Load Cells (LCs). Thus, for a more accurate analysis, only data from the LCs was taken into account. To attenuate the noise-to-signal ratio, a median filter was applied. A median filter is a non-linear digital filtering technique, which consists in replacing each entry with the median of neighbouring entries.

4.2.2 Tests performed with different drivers for the same driving route and vehicle

The tests regarding the validation of the system by collecting data on the same driving route (Figure 4.5) and the same vehicle but with different drivers were performed in the *AtlasCAR*. Eleven subjects tested the system in these conditions.

Figure 4.5: Route for the first test, for the same car and different drivers [33].

By analysing the data collected from these tests, it was possible to evaluate the drivers behaviour. Three representative examples are shown in Figure 4.6. It is possible to see that, for instance, Subject A uses much less the brake than Subject B or C. The brake intensity of Subject A is also inferior to the brake intensity of the other subjects, that reach high levels most of the times thus leading to the saturation point of the force sensors (Table 4.2). Thus, overall Subject A seems to have a sporty driving, while Subject B and C appear to have a more normal driving. With these tests it was possible to validate the robustness of the system in detecting different driving patterns from different subjects.

After these tests each driver filled a form, in Figure 4.9, in order to evaluate some features of the system, such as: 1) the ease of calibration and habituation (Figure 4.7), 2) its interference with the normal driving (Figure 4.8a), 3) its contribution for road safety (Figure 4.8b) and 4) its use on a daily basis (Figure 4.8c). The results of the inquiries confirm the ease of calibration and habituation to the system, seen before in the preliminary test. Also, it was possible to confirm its non-invasiveness, as for all subjects the system did not interfere with the normal driving. Finally, most of the subjects agree on the monitoring pedal system contribution for road safety and probably could use it on a daily basis.

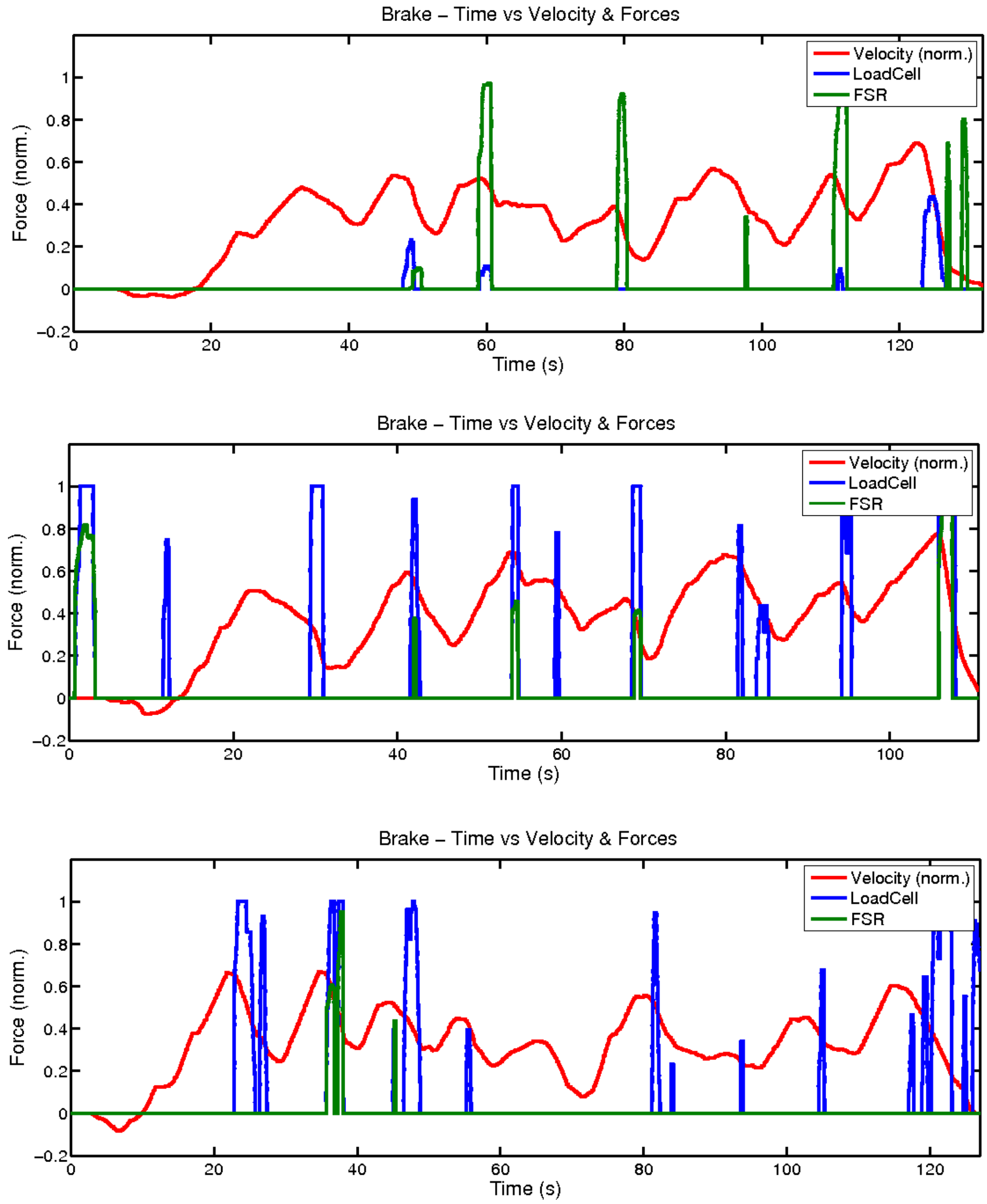


Figure 4.6: Data collected in the first test, for the same route, same car and different drivers. Top: Subject A; Middle: Subject B; Bottom: Subject C.

Subject	Pressed [Times]	Average [%]	Max [%]
A	4	1	43
B	10	9	100
C	15	9.6	100
D	11	11.6	100
E	8	9.8	100
F	5	0.8	28.4
G	7	1.8	50
H	14	8.4	100
I	7	3.1	55
J	21	3.4	58

Table 4.2: Data collected from the LoadCell for the subjects. Number of times that the brake was pressed, the average and the maximum value of the applied force are presented.

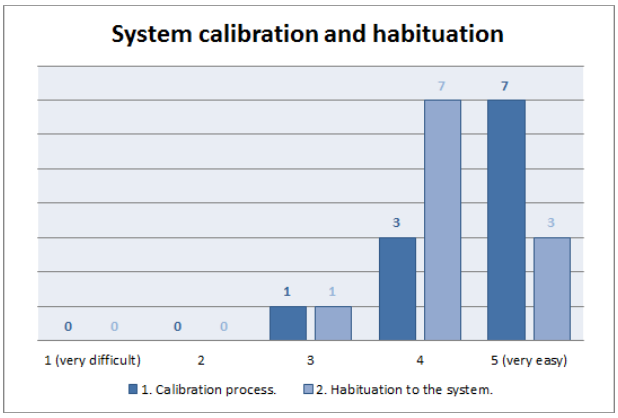


Figure 4.7: Subjects evaluation of the calibration and habituation processes.

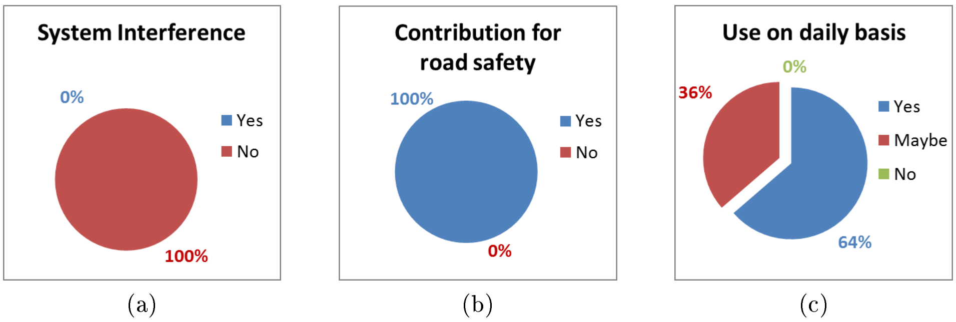
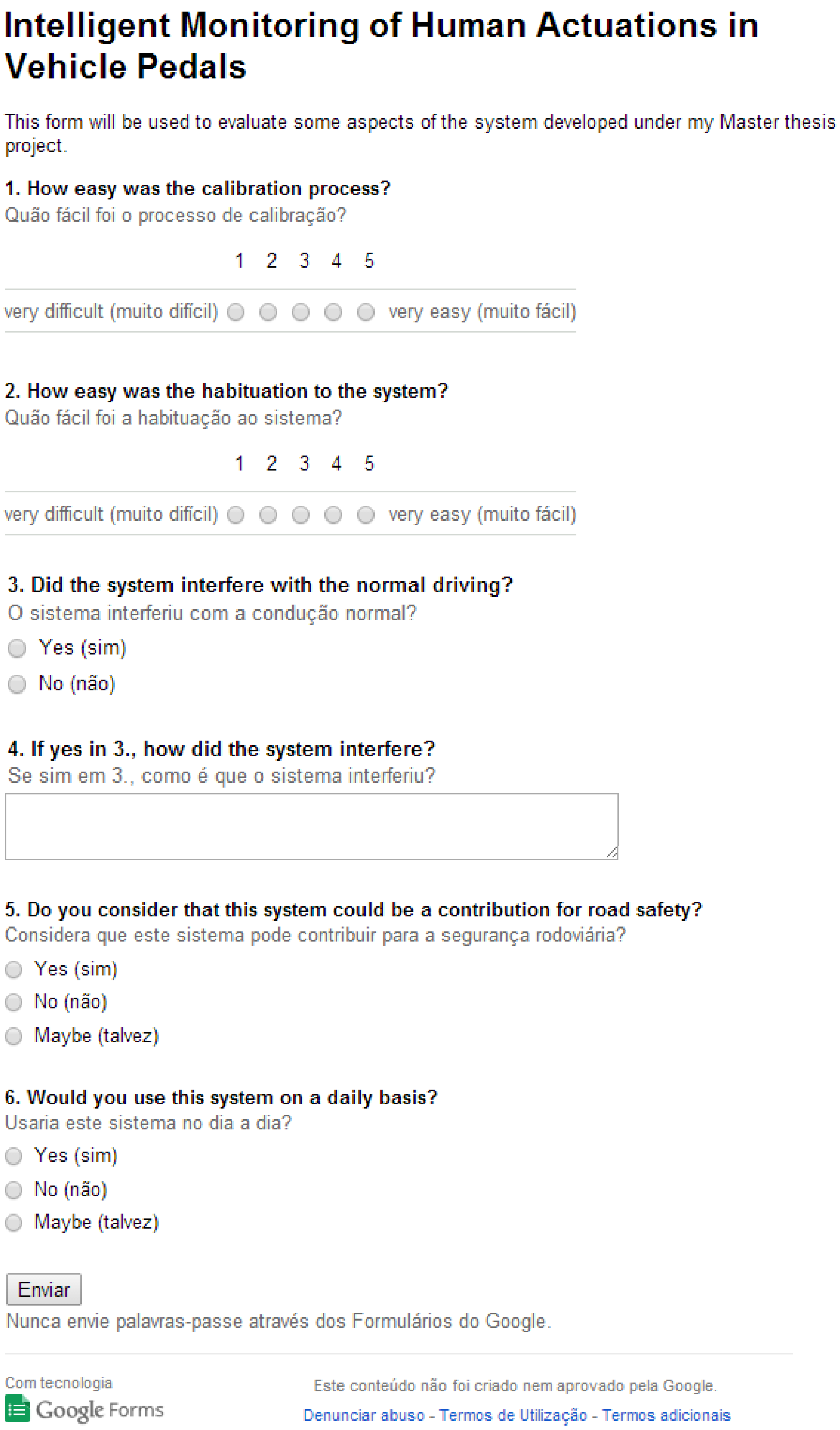


Figure 4.8: Subjects opinion of the system regarding its: a) interference with the normal driving, b) contribution for road safety and c) usage on daily basis.



Intelligent Monitoring of Human Actuations in Vehicle Pedals

This form will be used to evaluate some aspects of the system developed under my Master thesis project.

1. How easy was the calibration process?
Quão fácil foi o processo de calibração?

	1	2	3	4	5	
very difficult (muito difícil)	○	○	○	○	○	very easy (muito fácil)

2. How easy was the habituation to the system?
Quão fácil foi a habituação ao sistema?

	1	2	3	4	5	
very difficult (muito difícil)	○	○	○	○	○	very easy (muito fácil)

3. Did the system interfere with the normal driving?
O sistema interferiu com a condução normal?

- Yes (sim)
- No (não)

4. If yes in 3., how did the system interfere?
Se sim em 3., como é que o sistema interferiu?

5. Do you consider that this system could be a contribution for road safety?
Considera que este sistema pode contribuir para a segurança rodoviária?

- Yes (sim)
- No (não)
- Maybe (talvez)

6. Would you use this system on a daily basis?
Usaria este sistema no dia a dia?

- Yes (sim)
- No (não)
- Maybe (talvez)

Enviar

Nunca envie palavras-passe através dos Formulários do Google.

Com tecnologia Google Forms

Este conteúdo não foi criado nem aprovado pela Google.

Denunciar abuso - Termos de Utilização - Termos adicionais

Figure 4.9: Form to evaluate the monitoring of vehicle pedals system.

4.2.3 Tests done at different velocities, using only the brake pedal to stop the car

The tests using different velocities and only the brake pedal to stop the car, were performed in order to test if the force used on the brake pedal is actually proportional to the braking force and which would be the proportionality constant, K, as mentioned before. These tests were conducted under a velocity up to 60km/h and the brake was applied with the vehicle ungeared (with the clutch pedal fully pressed). This ungeared brake condition was necessary to reduce the influence of other variables for stopping the car, such as the engine braking.

For the analysis of the collected data, a median filter was also applied to reduce the noite-to-signal ratio. In this case the velocity was normalized to 60km/h, which was the maximum velocity of these tests, as shown in Figure 4.10.

First, the braking force periods were separated from the non-braking ones; for a braking force period to be considered for analysis the force applied on the pedal had to be higher than 1%. To understand the correlation between the energy on the brake pedal and the energy to stop the car, the areas of the braking periods were calculated by integrating the product of the force with the velocity, for that time period. To look only to the relevant braking areas, the area values lower than the mean were discarded. With this, it was possible to find a proportionality constant that represents the average of the proportionality constants for each braking period. This constant was calculated by dividing the energy used to stop the car by the energy used on the brake pedal, as shown before in equation 4.3. For these tests, the average of the proportionality constant value was ~615.

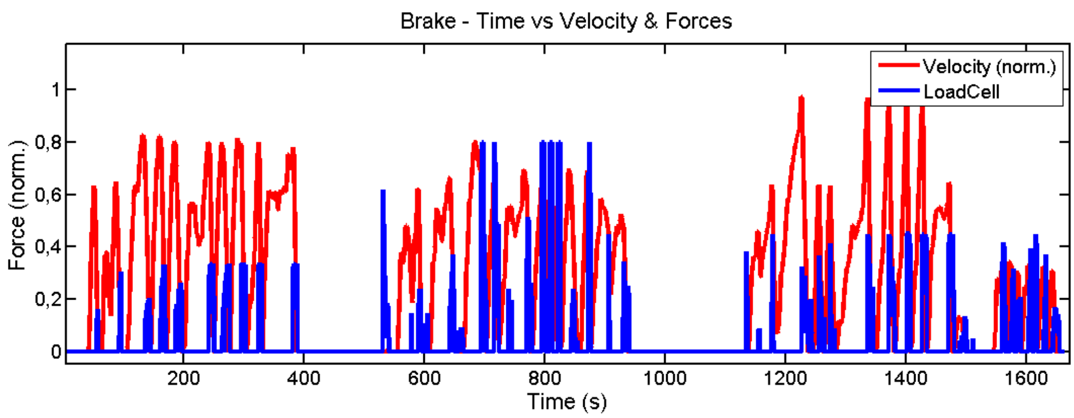


Figure 4.10: Data collected for different velocities using only the brake pedal to stop the car.

To diminish the variability associated with different velocities, the braking force periods were further sorted by a specific velocity, so that the starting point of each brake had to be comprised between ±5km/h of the selected velocity. When analysing the group of braking force periods for 60±5km/h, as shown in Figure 4.11, the average of the proportionality constant found was $K =\sim 677$ (Table 4.3). The average of the proportionality constant for the braking force periods starting in 50±5km/h, as shown in Figure 4.11, was $K =\sim 633$ (Table 4.4). For the 40±5km/h and for the the 30±5km/h, as shown in Figures 4.13 and 4.14, respectively, the average of the proportionality constant continues to decrease ($K =\sim 526.50$ and $K =\sim 510.40$), as shown in Table 4.5 and 4.6. Unexpectedly, for the 20±5km/h velocity, as shown in Figure 4.15, the average of the proportionality constant increases ($K =\sim 728.45$), as shown in Table 4.7.

For the velocity of 40±5km/h the K values range considerably. This can be partially explained by the saturation of the load cells at certain time periods (~750-900s; Figure 4.13), which can be related to pressing the pedal over the calibration limits. Also, for the 30±5km/h and 20±5km/h velocities the K values continue to range considerably. This variations may be due to other variables that weren't taken into account, such as the energy transferred to the brakes and dampers, the kinetic friction between the wheels and the road. Moreover, it seems that for low velocities this kinetic friction has more impact in braking.

Overall, it was possible to find a proportionality constant that appears to decrease with the velocity (Figure 4.16). Meaning that perhaps the energy to stop the car decreases at the same rate as the velocity decreases.

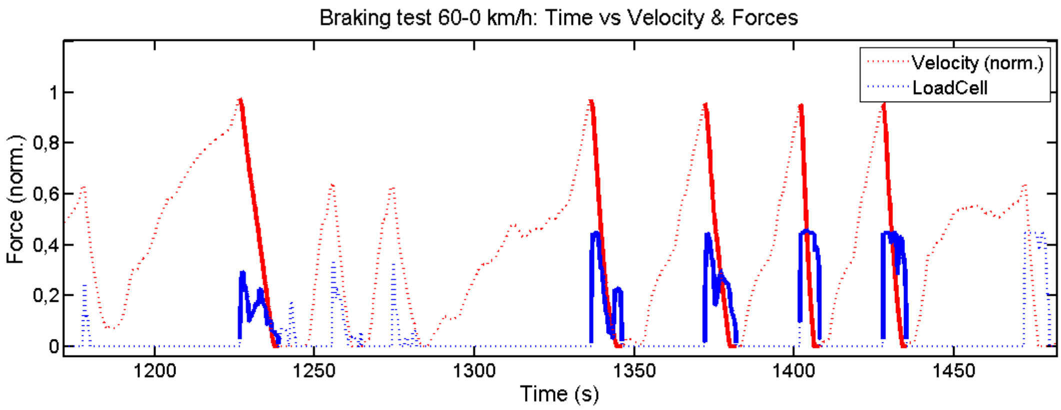


Figure 4.11: Data analysis for 60 km/h using only the brake pedal to stop the car. Only the periods where there were relevant braking forces are shown, the dotted lines represent the excluded braking periods.

	60 km/h
K values	0.776
	0.678
	0.647
	0.827
	0.627
$\overline{K}$	**0.678**

Table 4.3: K values for 60 km/h.

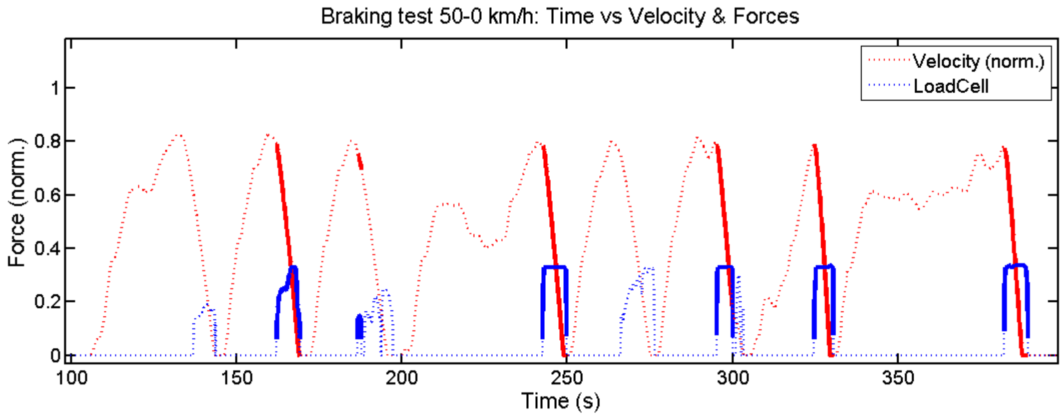


Figure 4.12: Data analysis for 50 km/h using only the brake pedal to stop the car. Only the periods where there were relevant braking forces are shown, the dotted lines represent the excluded braking periods.

	50 km/h
K values	0.671
	0.648
	0.547
	0.577
	0.711
	0.619
$\overline{K}$	**0.634**

Table 4.4: K values for 50 km/h.

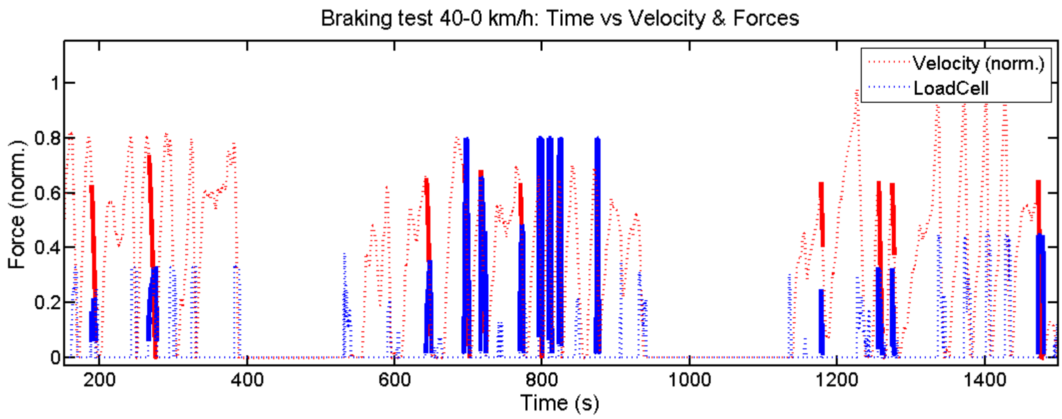


Figure 4.13: Data analysis for 40 km/h using only the brake pedal to stop the car. Only the periods where there were relevant braking forces are shown, the dotted lines represent the excluded braking periods.

	40 km/h
K values	0.843
	0.524
	0.646
	0.284
	0.338
	0.529
	0.332
	0.377
	0.419
	0.456
	1.299
	0.842
	0.959
	0.544
$\overline{K}$	**0.526**

Table 4.5: K values for 40 km/h.

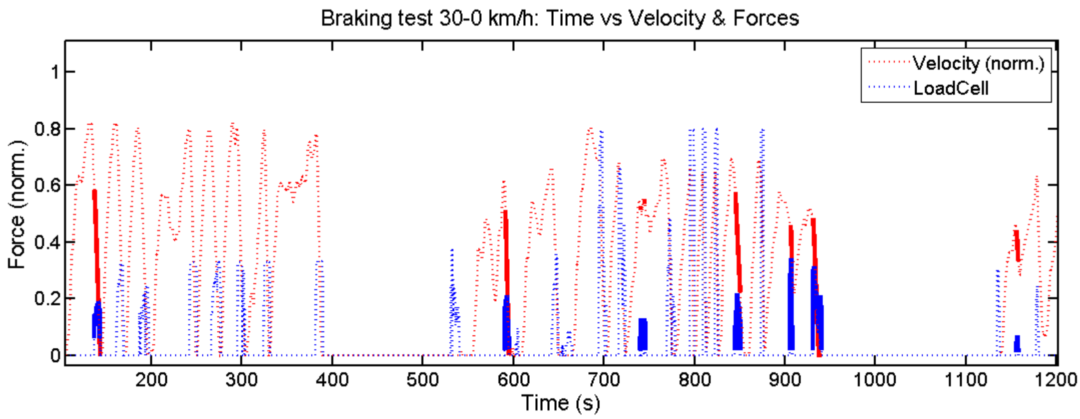


Figure 4.14: Data analysis for 30 km/h using only the brake pedal to stop the car. Only the periods where there were relevant braking forces are shown, the dotted lines represent the excluded braking periods.

	30 km/h
K values	0.888
	1.181
	0.389
	0.298
	0.670
	0.510
	0.462
$\overline{K}$	**0.510**

Table 4.6: K values for 30 km/h.

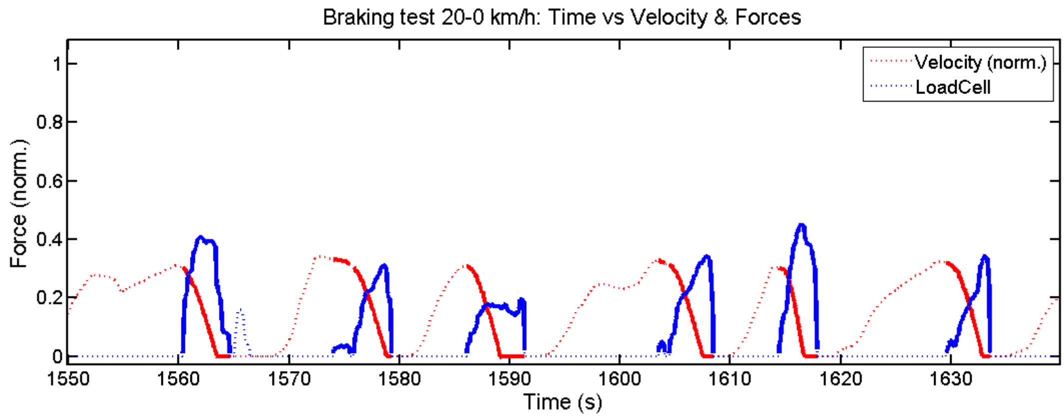


Figure 4.15: Data analysis for 20 km/h using only the brake pedal to stop the car. Only the periods where there were relevant braking forces are shown, the dotted lines represent the excluded braking periods.

	20 km/h
K values	0.411
	0.575
	0.781
	1.065
	0.572
	0.782
	0.676
	1.156
$\overline{K}$	**0.728**

Table 4.7: K values for 20 km/h.

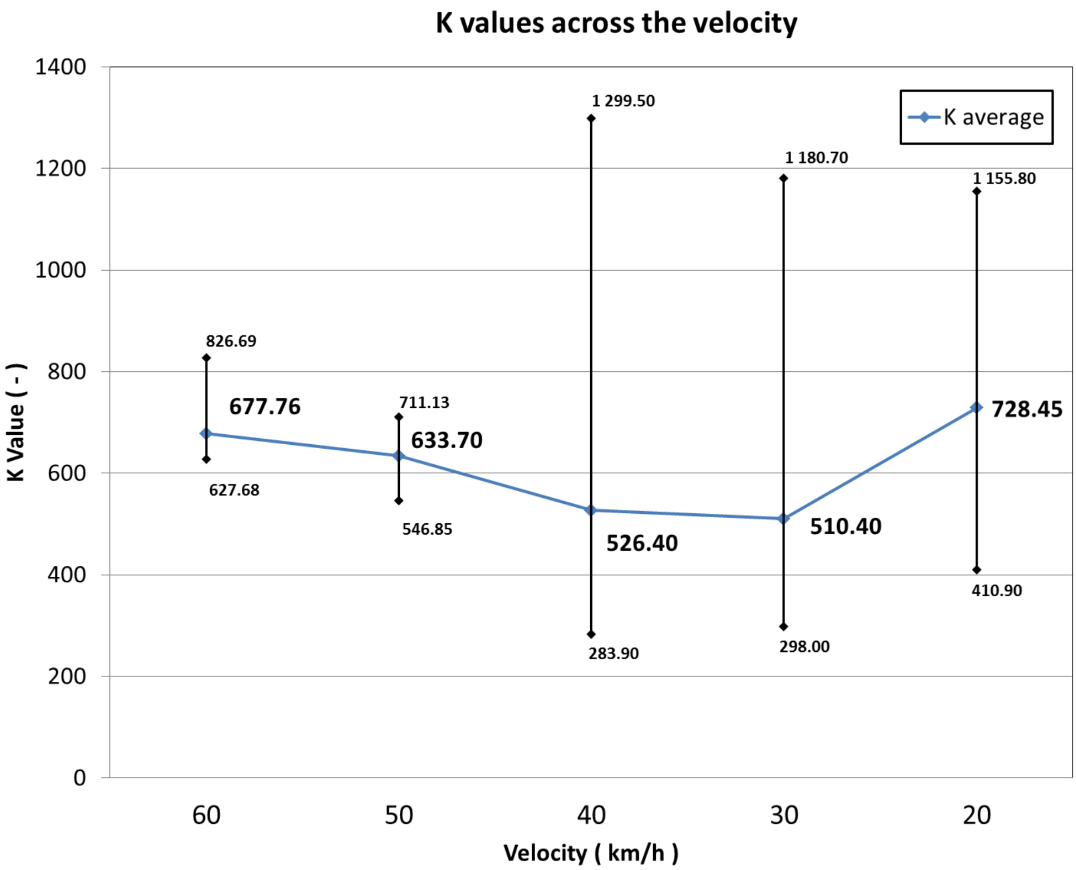


Figure 4.16: Proportionality constant, K value, across the velocity.

4.3 Application in Intelligent Monitoring context

As mentioned before, there are several assistance driving systems. For example, a visual system to detect pedestrians can detect if an object is crossing the road and calculate and warn the driver about the estimated time to collision (ETC). These calculations are generally based on the velocity of the car, on the velocity of the object and on the distance to the object. Independently of the vehicle, if these variables are the same the ETC would also be the same. Nonetheless, the braking response is different in each vehicle, which could lead to a not sufficient and efficient braking.

By integrating the monitoring system with this visual system, it would be possible to improve these braking situations. For instance, by knowing the necessary time to brake in safety and replacing it by t on the Equation 4.4 and assuming that the final velocity would be 0, $v_f = 0$, it would be possible to find the required force to be applied on the brake pedal.

$$K = \frac{\frac{1}{2} \cdot m_{car} \cdot \left|(v_f^2 - v_i^2)\right|}{\int_0^t F_{pedal} \cdot v_{car} \cdot dt} \Leftrightarrow K = \frac{\frac{1}{2} \cdot m_{car} \cdot \left|-v_i^2\right|}{\int_0^t F_{pedal} \cdot v_{car} \cdot dt} \tag{4.4}$$

Also, assuming that while driving the mass of car can be considered constant, it is possible to simplify the Equation 4.4:

$$K = \frac{\frac{1}{2} \cdot m_{car} \cdot \left|-v_i^2\right|}{\int_0^t F_{pedal} \cdot v_{car} \cdot dt} \Leftrightarrow K_2 = \frac{\left|-v_i^2\right|}{\int_0^t F_{pedal} \cdot v_{car} \cdot dt} \tag{4.5}$$

With this required force to brake safely and by knowing the force applied on the brake pedal in real time, this intelligent monitoring system could warn the driver accordingly if the force applied is more or less than the required to stop the vehicle safely.

Chapter 5

Conclusion and Future work

This project concerned the development of a system to measure the driver actuations on vehicle pedals. The system was planned to fulfil important features, such as: 1) being easy-to-use and flexible, 2) non-invasiveness, 3) robustness, 4) real-time monitoring and 5) low cost unit and operation. In order to develop the monitoring system, this project was subdivided in three parts: the hardware development, the software development and data analysis. After the completion of the hardware and software it was possible to start the data collection. Data from the vehicle pedals was collected in two states: stationary driving and real driving conditions.

The results obtained on the stationary conditions indicated that generally the system was easy-to-use, mainly due to its quick and effortless installation, and flexible enough to install in different pedals and vehicles. Thus, it was possible to validate the flexibility and ease of use of the system.

On real driving conditions other aspects were tested, such as: non-invasiveness, robustness and the capacity to monitor the pedals in real-time. While testing the system, it was generally not an obstacle to the driver, which indicates its non-invasiveness. Nonetheless, the system to fix the case on the pedals was occasionally not very effective, leading to some oscillations and dismounts; this could be fixed by using a belt as the fastening system. When collecting data the system generally showed robustness as it was also possible to monitor the pedals in real-time; the data displayed on the user interface correlated well with the actuations on the pedal. Considering the data from real driving situations, generally the clutch pedal activity is higher than the brake pedal, which indicates that a major part of the driving is actually controlled by the clutch. Also, as expected, the force values on the brake pedal appear to be inversely correlated with the velocity. In addition, it was possible to differentiate the behaviour of the driver actuation on the monitored pedals, distinguishing driving patterns related to different drivers.

It was also possible to find a proportionality constant between the energy used on the brake pedal and the energy used to stop the car. This constant allows the definition of a braking model of the vehicle based on the pedals actuation, where the energy necessary to stop the vehicle can be found just by knowing the force applied on the pedals and the car velocity. This model could be an advantage when integrated with other systems. For example, there are visual systems to detect pedestrians and warn the driver if a pedestrian is crossing the road. These systems calculate the necessary distance to brake safely based mainly in the car velocity and the distance to the pedestrian. However, there are situations where the braking system of the car is debilitated, increasing the time necessary to brake safely and leading to possible failures of the system. By integrating the braking model based on the pedals actuation with other variables of the car, a more accurately system to calculate the distance necessary to stop the car safely could be developed.

Even though the system is operational and fulfils important established criteria, several improvements could be done. For instance, if the acquisition board was replaced by a custom electronic device with bluetooth communication, an Android or iOS interface application could be developed. Because these interface applications would be able communicate with the system, the flexibility and convenience of the system would be improved. This together with the replacement of other elements by custom electronic devices, would make the system more accessible for future users by reducing unit and operation costs. Furthermore, for the system to become even more flexible and convenient, a device with Ethernet or Wifi connection could be used. With the possibility to create a local network and with an user interface developed in HTML and PHP languages, it would be possible to communicate with the system from any electronic device that has a browser. In addition, it would be important to improve not only the system to securely fix the sensors case on the pedals but also the fastening system of the infrared (IR) sensors, in order to make the orientation adjustment mechanism more efficient.

Overall, it was developed a easy-to-use, flexible, non-invasive and robust system capable of monitoring the human actuation on the vehicle pedals in real time. By connecting this monitoring system to a more complete information system it would be possible to correlate the data collected from the pedals with other variables of the car. With this, the system could not only evaluate the driver actuations in order to detect inefficient or risky usage of the pedals, but also to predict some dangerous situations by warning the driver in time so that he could act and ultimately prevent them.

Finally, as a prototype, the system can not bet considered low-cost. By replacing the acquisition board for other custom electronics devices, the system would become more flexible and more accessible for future users. The sensing unit prices were approximately: ~45€ for the Loadcells, ~6€ for the Force Sensitive Resistors and ~13€ for the IR sensors.

References

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

[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, University of Aveiro*, 2012.

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

[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_id=1101_0`.

[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] "Raspberry pi." `http://www.raspberrypi.org/`.

[29] 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.

[30] ROS.org, "ROS/Concepts - ROS wiki." `http://wiki.ros.org/ROS/Concepts`.

[31] A. Krause, "Foundations of GTK+ development," *Apress*, 2007.

[32] G. Software, "Glade - a user interface designer." https://glade.gnome.org/.

[33] "Google maps." https://maps.google.pt/.