INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Área Departamental de Engenharia Electrónica e Telecomunicações e de Computadores

Fatigue and drowsiness detection using inertial sensors and electrocardiogram

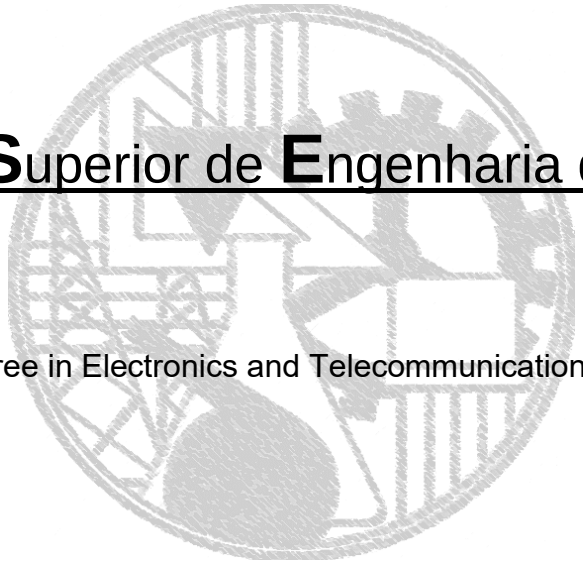
António Jorge Janicas Cerca
(Licenciado)

Dissertação realizada no âmbito de Trabalho Final de Mestrado para a obtenção de grau de Mestre em:
Engenharia Electrónica e Telecomunicações

Orientadores:
Professor Doutor André Ribeiro Lourenço
Professor Doutor Artur Jorge Ferreira

Júri:
Professor Doutor Carlos Eduardo Meneses Ribeiro
Professor Doutor Paulo Alexandre Carapinha Marques

Outubro de 2018



Instituto Superior de Engenharia de Lisboa

Master's degree in Electronics and Telecommunications Engineering

Fatigue and Drowsiness detection using inertial sensors and electrocardiogram

António Jorge Janicas Cerca

Supervisors:

André Lourenço, PhD

Artur Ferreira, PhD

September 30, 2018

Agradecimentos

A minha gratidão aos Professores André Lourenço e Artur Ferreira, por todo o conhecimento transmitido, apoio e conselhos fornecidos durante o desenvolvimento desta dissertação.

Um sincero agradecimento ao Professor Christer Ahlström, do Instituto de Pesquisa de Transportes e Estradas da Suécia, pela base de dados que tornou a análise das técnicas de classificação possível.

Um especial agradecimento à equipa da CadioID – Borja Carrillo, Ricardo Nunes, Carlos Carreiras, David Velez e Ricardo Rodrigues – pelo apoio na aquisição de dados de ECG e Acelerometria, bem como pelos conselhos de programação e conhecimento transmitido sobre BLE.

A todos os professores do Instituto Superior de Engenharia de Lisboa (ISEL) que me transmitiram o conhecimento necessário para começar e terminar o meu Mestrado em Engenharia Electrónica e Telecomunicações.

Uma consideração única aos meus pais, ao meu irmão e à minha namorada, por toda a motivação e apoio, para que eu nunca desista dos meus objectivos.

E finalmente, à minha família e aos meus amigos por me terem acompanhado durante toda a minha vida pessoal e académica.

Resumo

O interesse em monitorizar os condutores dos veículos durante a sua condução tem vindo a aumentar ao longo dos anos, com o objectivo de tornar as estradas mais seguras para condutores e peões. Com este pensamento, surgiu a ideia de desenvolver um sistema capaz de monitorizar a fadiga e a sonolência do condutor e, se necessário, alertá-lo sobre o seu estado físico e psicológico. O ADAS, conhecido como sendo um sistema de assistência avançada para os condutores, é um sistema que monitoriza o desempenho e o comportamento do automóvel, bem como as condições físicas e psicológicas do condutor.

Este sistema pode ter um comportamento passivo, alertando os condutores para situações de perigo eminente para que o condutor consiga evitar esses perigos. O LDW, ou aviso de mudança de faixa, é capaz de alertar o condutor de uma saída involuntária de faixa e o FCW, ou aviso de colisão frontal, consegue alertar o condutor de uma colisão eminente, tendo em conta o veículo frontal.

Por outro lado, o ADAS consegue concretizar acções de forma assegurar a segurança dos passageiros e dos peões. O AEB, ou travagem de emergência automática, identifica uma colisão eminente e trava sem intervenção do conduto e o LKA, ou assistente de manutenção de faixa, que movimenta o veículo para que este não saia da faixa de rodagem.

Esta dissertação é baseada no projecto CardioWheel, desenvolvido pela empresa CardioID, e consiste na monitorização do sinal cardíaco do condutor e na gravação dos movimentos realizados pelo volante do veículo durante a condução. O sinal cardíaco, conhecido como ECG, é extraído através de eléctrodos secos fixados numa capa em pele colocada no volante, que conseguem captar o sinal eléctrico provocado pelo batimento cardíaco enquanto o condutor estiver com as mãos no volante. O controlo dos movimentos do volante, ou SWA, é conseguido através de um acelerómetro de 3 eixos colocado no centro do volante que grava as variações da aceleração instantânea enquanto o condutor movimenta o volante. Através dessas acelerações é possível calcular-se o ângulo de rotação do volante durante todo o percurso.

Os dados adquiridos de ECG e SWA geram uma enorme quantidade de informação que tem que ser codificada de forma a reduzir a largura de banda necessária à transmissão. Técnicas no domínio do tempo, como o AZTEC, TP e o CORTES, estão bem documentadas como boas técnicas para compressão de sinal ECG onde o principal objectivo é a obtenção da pulsação cardíaca. Dadas as exigências do projecto, concluiu-se que estes métodos não seriam os melhores para preservar as características principais do sinal de forma a obter-se padrões de fadiga e sonolência. Outros métodos de codificação com e sem perdas foram testados tanto para compressão de sinal ECG como para SWA e pode-se concluir que o

método híbrido de Codificação Linear Preditiva com a técnica Lempel-Ziv-Welch é o método sem perdas em que se obteve maior rácio de compressão. Por outro lado, outro método hibrido utilizando escalamento de amplitude com DWT, provou ser o método com perdas com maior rácio de compressão onde o erro quadrático médio é reduzido.

A transmissão da informação comprimida é assegurada através de um módulo BLE, presente no CardioWheel, no entanto, foi possível concluir que outras tecnologias como ZigBee ou ANT seriam igualmente compatíveis com o propósito do projecto. Foi desenvolvido especificamente para este projecto um perfil BLE com a capacidade de transmitir a informação do sinal ECG e do acelerómetro em tempo real.

Para detectar se o condutor está a apresentar sinais de fadiga ou sonolência, foram testados vários algoritmos de aprendizagem automática que, de acordo com a informação ECG e do acelerómetro enviada pelo volante, conseguem detectar esses padrões. A escala KSS, é uma escala subjectiva que identifica o nível de sonolência de uma pessoa e que permite a classificação do nível de sonolência do condutor.

Para construir um algoritmo de inteligência artificial é necessário extrair-se características dos sinais a interpretar. Essas características têm que descrever o sinal de forma precisa para que os algoritmos de aprendizagem automática consigam interpretar e classificar cada sinal da forma adequada. Características como ritmo cardíaco ou amplitude da onda R são exemplos de características utilizadas para descrever o sinal ECG. Características como tempo com o volante estático e aceleração média são exemplos de características utilizadas para descrever o sinal de SWA.

Para além das características, um algoritmo de aprendizagem automática necessita de uma base de dados que consiga cobrir todas as situações possíveis para que o algoritmo, olhando para os dados inseridos, consiga detectar os padrões nas características para cada resultado final possível.

Métodos de regressão foram implementados de forma e testar o seu desempenho para um problema de classificação, no entanto, não provaram ser os melhores métodos para essa abordagem. De todas as técnicas de classificação testadas, o método de SVM, ou máquina de vectores de suporte, provou ser o que obtém melhores resultados de classificação.

Com os resultados obtidos será possível implementar-se um sistema de alarmística que consiga avisar o condutor sobre o seu estado físico e psicológico, aumentando assim a segurança rodoviária.

Palavras-chave:

ECG, Acelerómetro, Sonolência, Fadiga, Compressão, BLE, Aprendizagem automática.

Acknowledgments

My gratitude to Professors André Lourenço and Artur Ferreira, for the knowledge transmitted, support and given suggestions during this dissertation development

Sincere thanks to Professor Christer Ahlström, from Swedish National Road and Transport Research Institute, for providing the dataset that made the classification possible.

Special thanks to the CardioID team – Borja Carrillo, Ricardo Nunes, Carlos Carreiras, David Velez and Ricardo Rodrigues – for the support in ECG and SWA data acquisition, as well for the programming advices and BLE knowledge transmitted.

To all Professors in the Engineering Institute of Lisbon (ISEL) that transmitted the necessary knowledge to start and finish my degree in Electronics and Telecommunications Engineering.

A unique remark to my parents, my little brother and my girlfriend for the motivation and support, to never give up fighting for my objectives.

Finally, to all my family and friends that have accompanied me throughout my personal life and academic life.

Abstract

The interest in monitoring a driver's performance has increased in the past years in order to make the roads safer both for drivers and pedestrians. With this thinking in mind, it arises the idea of developing a system to monitor driver's fatigue and drowsiness to alert him, if needed, about his psychological and physical states.

This dissertation is based on the CardioWheel system, developed by CardioID, and consists in monitoring the person's ECG signal and to record the motion of the steering wheel during the journey. The ECG signal is extracted with dry-electrodes placed in a conductive leather covering the steering wheel that can sense the electrical signal caused by the heartbeat of the person while having the hands on the wheel. The steering wheel movement monitoring is performed with the help of a three-axis accelerometer placed in the middle of the steering wheel that records the proper acceleration variations while moving the steering wheel. With those accelerations it is possible to calculate the steering wheel rotation angle during all the journey.

The amount of data acquired with this system undergoes a compression stage for transmission with the goal of reducing the necessary bandwidth. From the evaluated techniques for data compression, it was possible to conclude that the hybrid method using Linear Predictive Coding and Lempel-Ziv-Welch is the lossless technique with the highest Compression Ratio. However, the hybrid technique using amplitude scaling e DWT is the lossy method with the highest Compression Ratio and a reduced RMSE.

The transmission of the compressed data is done via Bluetooth® Low Energy, available in the CardioWheel system, with an exclusive profile developed for this dissertation. This profile has the ability to transmit the ECG and accelerometer data in real time.

To detect if the driver is becoming drowsy, were evaluated machine learning algorithms to detect fatigue and drowsiness patterns according to the received ECG and accelerometer data from the steering wheel. Many features were extracted to describe the main characteristics from both signals and, from all the tested techniques, the Support Vector Machine technique proved to be the best classification method with the higher accuracy in classification.

With these tested results, it could be possible to implement an alarmistic system, to warn the driver about his psychological and physical states, increasing the safety in the roads.

Keywords:

ECG, Accelerometer, Drowsiness, Fatigue, Compression, BLE, Machine learning.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADAS – Advanced Driver Assistance System

ADC – Analogue-to-Digital Converter

AEB – Automatic Emergency Braking

ANN – Artificial Neural Network

AR – Autoregressive

AZTEC – Amplitude Zone Time Epoch Coding

BLE – Bluetooth Low Energy

BMP – Bitmap

BPF – Band-Pass Filter

BRF – Band-Reject Filter

CORTES – Coordinate Reduction Time Encoding Scheme

CR – Compression Ratio

CRC – Cyclic Redundancy Check

CSMA/CA – Carrier Sense Multiple Access/Collision Avoidance

DCT – Discrete Cosine Transform

DFT – Discrete Fourier Transform

DPCM – Differential Pulse Code Modulation

DWT – Discrete Wavelet Transform

ECG – Electrocardiogram

EEG – Electroencephalogram

EOG - Electrooculogram

FCW – Forwarding Collision Warning

FFT – Fast Fourier Transform

FIR – Finite Impulse Response

GAP – Generic Access Profile

GATT – Generic Attribute

GIF – Graphics Interchange Format

HPF – High-Pass Filter

HU – Hidden Units

I^2C – Inter-Integrated Circuit

IEEE – Institute of Electrical and Electronics Engineers

IIR – Infinite Impulse Response

IoT – Internet of Things

JPEG – Joint Photographic Experts Group

KSS – Karolinska Sleepiness Scale
LDW – Lane Departure Warning
LinReg – Linear Regression
LKA – Lane Keeping Assist
LogReg – Logistic Regression
LPC – Linear Predictive Coding
LPF – Low-Pass Filter
LZ77 – Lempel-Ziv 77
LZ78 – Lempel-Ziv 78
LZW – Lempel-Ziv-Welch
MAC – Medium Access Control
MP3 – MPEG Layer 3
MPEG – Moving Picture Experts Group
MSE – Mean-Squared Error
ODR – Output Data Rate
P2P – Peer-to-Peer
PCA – Principal Component Analysis
PCM – Pulse Code Modulation
RF – Random Forest
RLE – Run-Length Encoding
RMSE – Root-Mean-Squared Error
SMOTE – Synthetic Minority Over-sampling Technique
SNR – Signal-to-Noise Ratio
SPI – Serial-to-Peripheral Interface
SVM – Support Vector Machine
SWA – Steering Wheel Angle
TDM – Time-Division Multiplexing
TP – Turning Point
WPAN – Wireless Personal Area Network
ZCR – Zero-Crossing Rate
ZOI – Zero-Order Interpolator

Chapter 1

Introduction

1.1 Motivation

Fatigue and drowsiness are two factors that affect the driving abilities of each person. There is an increasing interest in the development of Advanced Driver Assistance Systems (ADAS) [1], which monitors the vehicle performance and behaviour, as well as the physical and psychological conditions of the driver.

Acting as a passive system, ADAS alerts drivers of a potentially dangerous situation so that a driver can take action to correct it. For example, Lane Departure Warning (LDW) alerts the driver of unintended lane departure, and Forward Collision Warning (FCW) indicates that under the current dynamics relative to the vehicle ahead, a collision is imminent.

In contrast, ADAS can take action to ensure the safety of the passengers and pedestrians. For example, Automatic Emergency Braking (AEB) identifies the imminent collision and brakes without any driver intervention, and Lane Keeping Assist (LKA) automatically steers the vehicle to stay within the lane boundaries.

To find patterns in the driving style of each person, there are several sensors in the market that enable the monitorisation of the driver's condition. Accelerometers, for example, are inertial sensors that measure the proper acceleration applied to an object, called g force. They can be placed on the automobile's steering wheel to monitor their movements.

In addition to the use of inertial signals, physiological signals can be monitored, such as, the electrocardiogram (ECG) signal. The cardiac signal can be obtained with the aid of dry-electrodes placed on the vehicle's steering wheel, such that, in contact with the human skin, it detects the electrical signals caused by the heartbeat.

The fatigue and drowsiness detection could be achieved with techniques from a subset of artificial intelligence, named machine learning. Machine learning algorithms use statistical techniques to give computers the ability to learn data, without being explicitly programmed for that purpose. With these methods, it is possible to identify the sleepiness characteristics in both ECG and accelerometer data and to predict if the driver is having some sights of sleepiness.

1.2 The proposed approach

The global approach for this dissertation is composed by two main parts: the acquisition system and the gateway solution. The acquisition system is responsible for all data collection, pre-processing, and transmission. The gateway solution is responsible for data reception, classification, and alarm activation.

Figure 1 illustrates the proposed system approach.

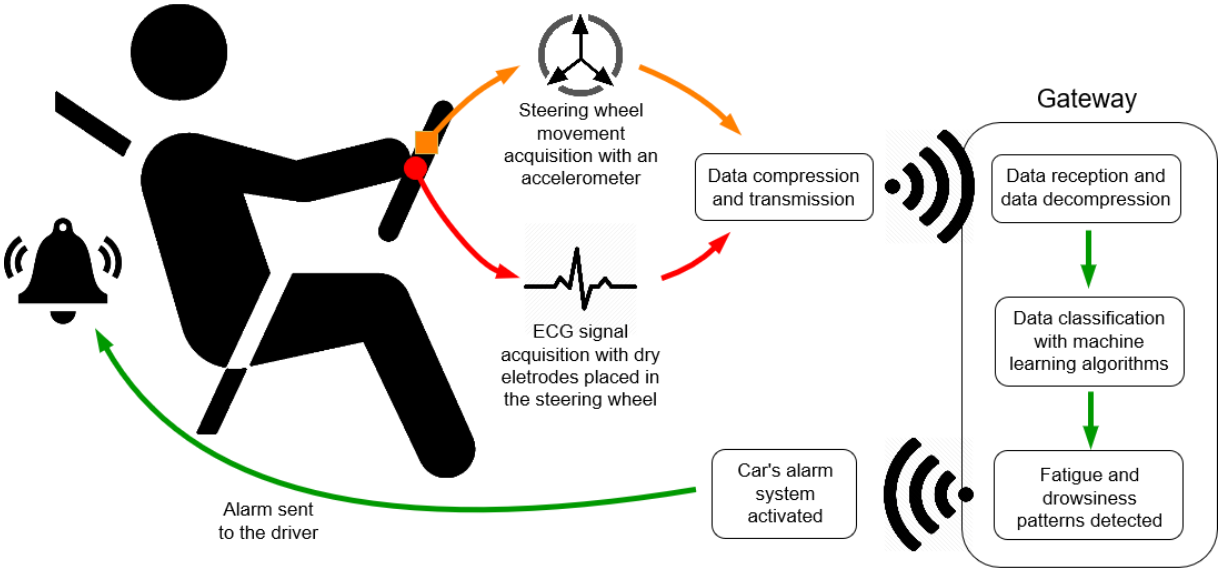


Figure 1 – Illustration of the system to be implemented.

In the acquisition system, the accelerometer and the ECG system will work for the entire driving period, which can last several hours. So, with the amount of acquired data, it is expected a significant volume of information to store and thereby, it is necessary to compress the data in order to require less storage space.

There is a lot of research in this area in which different compression methods such as Amplitude Zone Time Epoch Coding (AZTEC), Turning Point (TP), Coordinate Reduction Time Encoding Scheme (CORTES), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Huffman, Lempel-Ziv-Welch (LZW), Differential Pulse Code Modulation (DPCM), Linear Predictive Coding (LPC) or Run-Length Encoding (RLE), were tested for ECG signal compression, in the way to find the algorithm that gets the best compression ratio without introducing a significant error. It will be necessary to evaluate whether the methods used for ECG signal compression are equally valid for the compression of the inertial signals obtained with the aid of the accelerometer.

In addition to the compression, a way of not having to store large amounts of bits is to transmit that information via wireless to databases not physically attached to the acquisition system. Technologies such as Bluetooth® Low Energy (BLE), ZigBee or ANT may be the solution to this problem and it is necessary to know which of these wireless technologies best suits to the problem of transmitting the compressed data to a processing unit outside the acquisition system.

The CardioWheel [2] system, developed by CardioID, allows the acquisition of ECG and accelerometer signals in a non-intrusive way, with a BLE module for wireless transmission purposes.

In the gateway solution, upon receiving the transmitted data it is required a trained machine learning algorithm to classify the incoming data and to predict if the driver is in a capable state to keep driving. Random Forest (RF), Artificial Neural Networks (ANN) and Support Vector Machines (SVM) were tested for ECG and accelerometer data classification in a way to identify the patterns of sleepiness in both signals to get a high accuracy in classification.

The gateway solution will be responsible for the activation of an alarmistic system that can warn the driver in case it was detected fatigue and drowsiness patterns in the extracted signals.

1.3 Document organisation

The remainder of this dissertation is composed by seven chapters, described as follows:

- Chapter 2, Monitoring Systems, Sensors and Biological Signals, where the theoretical concepts of the accelerometer and ECG signals are discussed, giving an overview of the techniques to acquire and to measure these signals;

- Chapter 3, Data Pre-processing and Compression, that describes the basic concepts for data processing, and also explains some of the data compression algorithms for lossy and lossless encoding.

- Chapter 4, Wireless Technologies, where some of the low-power wireless transmission methods are introduced and compared to assess which one best suits for transmitting compressed data at short distances.

- Chapter 5, Learning from Data, outlines the process for data classification, including the feature extraction, the operation of some machine learning algorithms and their performance assessment.

- Chapter 6, Proposed Solution, defines the practical problem addressed in this dissertation and the possible solutions, with an initial analysis of what will be tested and could be implemented.

- Chapter 7, Experimental Evaluation, describes the implementations and all the simulated tests that were carried out, for compression and classification, with a comparison based on metrics.

- Chapter 8, Conclusion, summarises all the conclusions taken from the experiments and what could be done to improve and to finish the overall project's global approach.

Chapter 2

Monitoring Systems, Sensors and Biological Signals

Monitoring systems are sensors or devices that measure parameters for a given purpose. There are two main types of monitoring: direct monitoring and indirect monitoring. Fatigue and drowsiness lead to a modification in the person's biological signals, and these types of monitoring are used to get information about the person's physical and psychological behaviour for medical, fitness or safety purposes.

This chapter is composed by three main folds. The monitoring systems fold contains section 2.1, Direct and indirect monitoring systems, where it is described the advantages and disadvantages of those two types of monitoring. Section 2.1.1, Integrated monitoring systems, addresses a specific type of indirect monitoring that is implemented in the vehicles nowadays.

The sensors fold, that includes section 2.2, Accelerometers, explains how an accelerometer operates and the importance of this sensor on the daily basis. The last main fold, the biological signals fold, includes section 2.3, ECG signal, that explains the theoretical aspects of an ECG signal and how its analysis is carried out, section 2.3.1, ECG signal acquisition methods, that shows two different ways to collect ECG data, and section 2.4, Fatigue and drowsiness, that explains the differences between these two words and how it is classified.

2.1 Direct and indirect monitoring systems

Nowadays, the monitorisation of human behaviour is increasing, whether for health care purposes in a medical basis or for the person's well-being, like in fitness or safety. It is possible to perform these monitoring with direct or indirect monitoring systems.

Direct monitoring systems deal with physiological signals or with a person's behaviour, for example, facial expressions, yawning, eye tracking and blinking, electrooculogram (EOG) electroencephalogram (EEG), electrocardiogram (ECG) and heart rate, body temperature, among others. The main advantages of these methods are [3]:

- Accuracy – measurements are under medical investigation and supervision;
- Universality – the results are valid or are directly linked with scientific or commercial domains;

- Versatility – the experiments can be tested in a laboratory environment since it is simple to reproduce good real conditions for the task of interest.

However, using this kind of monitoring, there are some disadvantages such as [3]:

- Privacy invasion – measurements can describe a lot of physical and psychological conditions of the individual;
- High sensitive – light, weather, dress or accessories, actual health conditions can decrease the precision of the measures.

Indirect monitoring systems interact with the objects controlled by the individual, for example, in an automobile, it is possible to monitor the steering wheel movements, pedal acceleration (gas or break), sitting position, as well as other indicators. Unlike direct monitoring systems, the main advantages of this kind of monitoring are [3]:

- Robustness – the influence caused by external sources like weather, cannot nullify the measurement;
- Privacy – the methods are non-intrusive to the individual;

On the other hand, these systems have the following disadvantages [3]:

- Experimental rigorous – to achieve significant results, the tests should be done using real conditions to best suit the measurements to the real environment;
- Low applicability – even the promising results usually cannot be reused in other research domains and are focused on a specific problem.

The best choice between these methods depends on the application.

2.1.1 Integrated monitoring systems

There are several electrical systems implemented in vehicles nowadays that make possible the motion control of the steering wheel, in an indirect way. One of these systems is the rotary encoder and it can be built in different ways [4].

An optical rotary encoder has a glass disc, with some opaque concentric rings with gaps in each other. In one side of the disc there is a light source that illuminates the disc and the light that passes through the glass disc is caught by an array of photodetectors that reads the combination of non-opaque areas.

Figure 2 illustrates how an optical rotary encoder system operates.

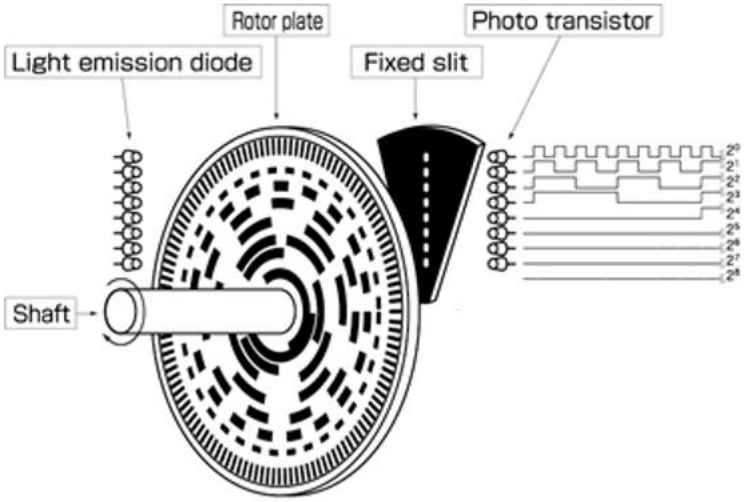


Figure 2 – Illustration of an optical rotary encoder.

A mechanical rotary encoder is composed by a metal disc, containing a set of concentric rings with gaps in each other. Attached to this metal ring, there is a stationary object with a set of electrical sensors holding as much sensors as the number of concentric rings. As the metal disc rotates, the electrical sensors will read a combination of electrical current. There is other interesting technique under the mechanical rotary encoder that allows the electrical connection between a stationary object with a rotational one – the slip ring.

The slip ring is composed by a set of brushes for each concentric ring which rubs on the rotating metal rings. As the disc rotates, the electric current is conducted through the stationary brushes.

Figure 3 represents an example of a pancake slip ring.

Figure 3 – Example of a pancake slip ring that can be used in a mechanical rotary encoder.

It is considered an absolute rotary technique when the output of these two encoding methods is a binary value that represents a unique steering wheel angle. However, it is possible to create an incremental rotary encoder where the disc has interpolated stripes and the angle calculation is done by the number of stripes counted during the motion. It increments or decrements the angle value with the number of stripes counted, depending if it performs a clockwise or counter clockwise move.

There are several other ways to implement this rotary encoder, but all are based in these main principles. Figure 4 illustrates an example of metal/glass disc of an absolute combination and an example of a two-track incremental disc.

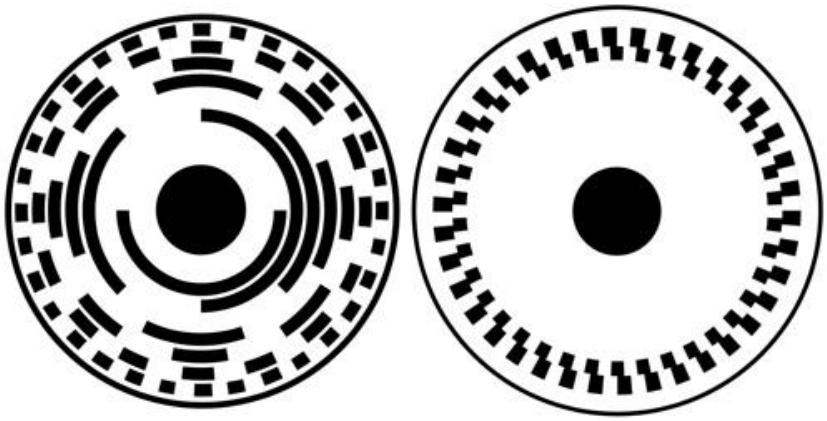


Figure 4 – Example of an absolute disc combination (left) and a two-track incremental disc (right).

2.2 Accelerometers

An accelerometer is an inertial sensor that measures the proper acceleration of an object, named as g force [5]. This acceleration is different from the acceleration as the time rate of the speed variation and is measured according to an axial complex present in the device.

There are accelerometers with only three axes (x, y, and z) and others with six axes that are designated as gyroscopes. These more complex accelerometers have the ability to detect rotations on each of the three axes (x, y, and z), and it is possible to monitor rotational movements on the accelerometer besides the axial acceleration.

Based on Newton's laws, when an object suffers acceleration, the mass, by the effect of inertia, tends to conserve its velocity, moving in the direction of an axis. This situation can be exemplified through a half-full glass of water. As we push the glass, causing acceleration, the water will move relative to the glass. The intensity of this movement gives a measure of acceleration that will be read by the accelerometer. The mechanical accelerometers operate

in a similar way. They are composed by a moving mass between fixed masses. As the moving mass comes near or moves away from the fixed masses, the capacitance measured in each fixed mass changes with the distance to the moving mass, making possible the measurement of the proper acceleration.

Figure 5 represents a model for the measurement of the proper acceleration in one axis.

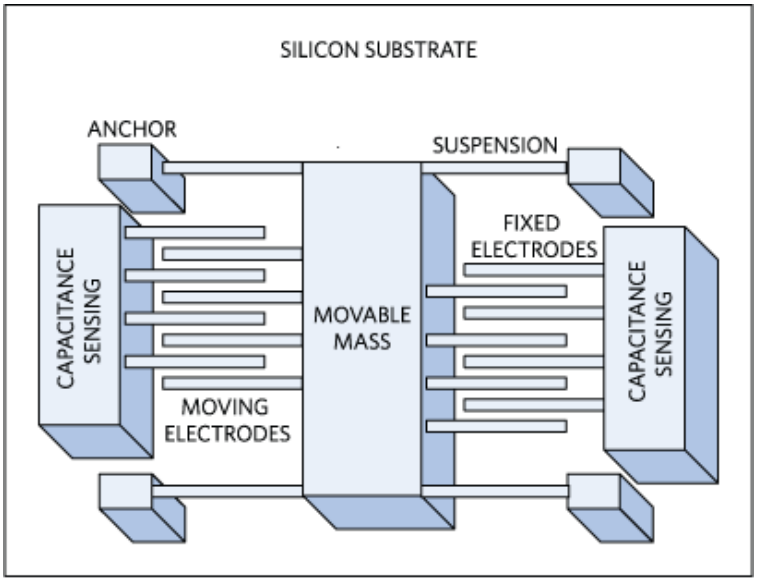


Figure 5 – Mechanical model of an accelerometer.

A free fall accelerometer cannot get any reading, however it is known that the acceleration caused by gravity is $1\,g$, meaning 9.81 m/s^2. On the other hand, if the accelerometer is in a flat surface it will measure 1 g of acceleration on an axis that is parallel to the vector of the gravitational acceleration.

Nowadays, accelerometers are used in a wide range of applications, such as in seismographs, impact measurement systems, motion sensors used in some gaming controllers, tilt sensors found in almost all smartphones as well as in automobile steering wheel for motion monitoring.

2.3 ECG signal

The electrocardiogram (ECG) signal is the electrical signal that the heart emits through successive contractions and distensions of the heart muscle, named myocardium [6] [7]. This signal is easily distinguished from other electrical signals by having a distinctive format where it is possible to identify five types of wave – P, Q, R, S, and T. In some cases, it is possible to identify a sixth wave called U.

Figure 6 shows the typical form of an ECG signal and its waves.

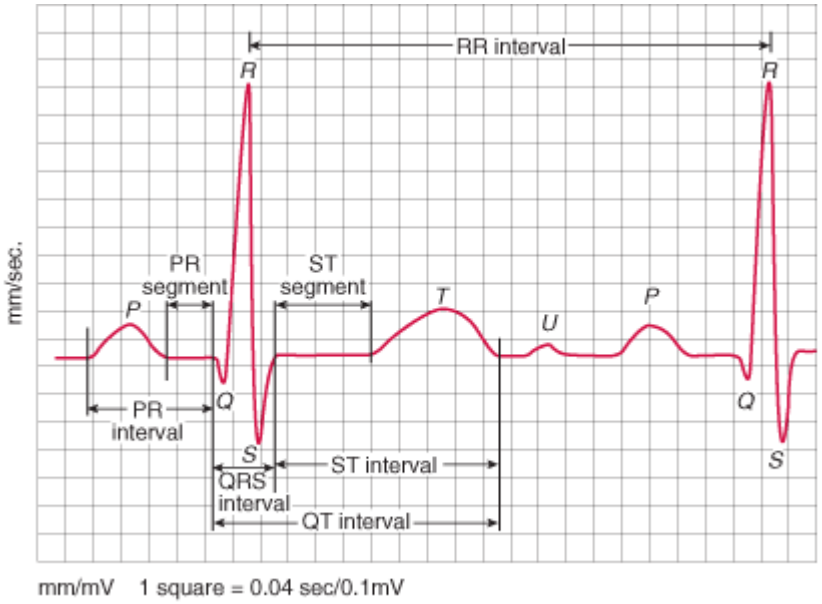


Figure 6 – Representation of an ECG signal and its waves [5].

The medical analysis of an ECG signal focuses mainly on the QRS wave complex. However, the P and T waves also have a high clinical value. Cardiac abnormalities are detected by considering the mean amplitude of each wave as well as the time intervals between them. Typically, the signal voltage values may range from 1 to 10 mV, with signal frequency values ranging from 0.05 to 100 Hz and a heart rate oscillating from 60 to 100 beats per minute [7].

Table 1 shows the mean voltage and duration values for each wave.

Table 1 – Amplitude and duration of the various ECG waves [7].

Amplitude [mV]	P wave	0.25
	R wave	1.60
	Q wave	25% of the R wave
	T wave	Between 0.1 and 0.5
Duration [s]	P-R interval	Between 0.12 and 0.2
	Q-T interval	Between 0.35 and 0.44
	S-T interval	Between 0.05 and 0.15
	P wave	0.11
	QRS complex	0.09

2.3.1 ECG signal acquisition methods

The acquisition of ECG signals can be done in two different ways: using intrusive or non-intrusive methods [8]. These acquisition methods have their own advantages/disadvantages and proper situations to use.

Intrusive methods are used in clinical settings where biological signals are extracted using devices placed in the human skin. These components are placed on the surface of the human body using a gel or a conductive paste that enables a better contact with the skin and, consequently, a better capture of the cardiac signals. These clinical methods may require the placement of, for example, up to twelve electrodes on the surface of the body to extract a good ECG signal and are limited to a small physical space of use, such as an ambulance, or a treatment room.

Non-intrusive methods allow the acquisition of signals where sensors do not have to be placed in the body, but in objects of everyday use. The purpose of these methods is to make the acquisition of signals almost involuntarily, without having an impact on the person's daily actions. The components used in this method are called dry-electrodes as they do not require the use of any conductive gels or pastes, taking advantage of human perspiration to improve contact with the individual's skin. These electrodes can be placed on any equipment, such as, for example, computer mice, keyboards, mobile phones, watches and cars' steering wheels. To obtain an acceptable biometric signal using this method only two electrodes are required, however this method becomes much less resistant to noise, making the signal processing more complex and harder to implement. After signal processing it is possible to achieve the very same performance as with hospital systems.

Figure 7 illustrates the differences between these two ways for acquiring an ECG signal.

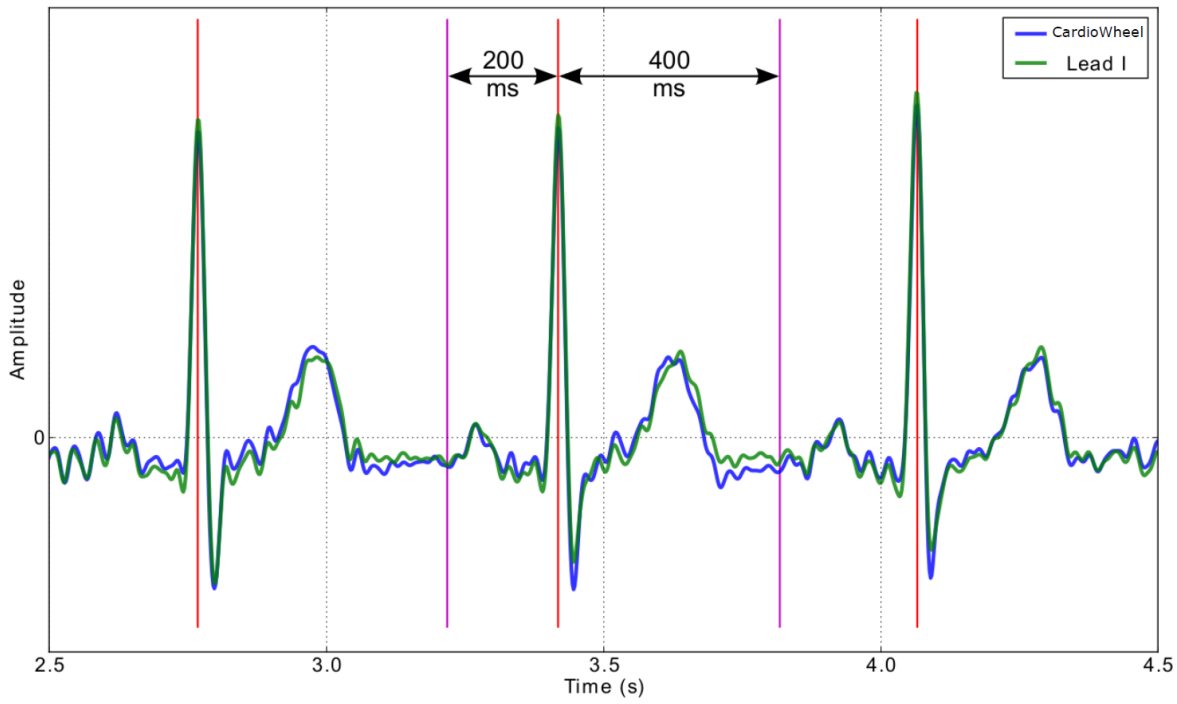


Figure 7 – Difference between an intrusive (green) and non-intrusive (blue) ECG signal acquisition.

2.4 Fatigue and drowsiness

Sometimes, fatigue and drowsiness are used to describe the same situation. These two words are quite related however they have a distinctive meaning [9].

Fatigue is a physical or psychological exhaustion. A person feels fatigued when, for example, goes to gyms and have worked his muscles and heart rate for a reasonable amount of time or when it has solved a large amount of complex mathematical problems. Fatigue, usually outcomes from doing the same task repeatedly or in an exhaustive way and it's the feeling of "I don't want to do this any longer". When the fatigue requires a rest, it could cause a person to fall in a drowsiness state.

Drowsiness is defined by the state before sleep. When a person is drowsy, he requires to sleep, and his body is fighting to stay awake. Drowsiness can interfere more actively than fatigue in the daily basis affecting concentration, reaction time, productivity and safety. Some medications induce drowsiness, but it is mostly related with sleeping habits, as people that have a good quality and a good quantity of sleep have more resistance of being in drowsiness state, for a longer period.

To classify the drowsiness state, there is a metric named Karolinska Sleepiness Scale (KSS) [10]. This is a subjective method, using a 10-point Likert scale [11], where the person classifies his sleepiness in periods of 5 minutes. Table 2 describes the KSS scale.

Table 2 – KSS scale value description [10].

Value	Description
1	Extremely alert
2	Very alert
3	Alert
4	Rather alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy, but no effort to keep awake
8	Sleepy, but some effort to keep awake
9	Very sleepy, great effort to keep awake, fighting sleep
10	Extremely sleepy, can’t keep awake

Chapter 3

Data Pre-processing and Compression

Acquiring data from various sensors usually implies that a large amount of data needs to be stored locally or sent to other devices. A good compression technique can save a large amount of disk space or bandwidth, depending on the purpose.

This chapter describes pre-processing and compression techniques. First, section 3.1, Pre-processing the data, addresses simple techniques to modify data according the needs, such as filtering and amplitude scaling. Section 3.2, ECG data direct time-domain techniques, gives an overview of direct time-domain methods that were tested for ECG data compression, such as Amplitude Zone Time Epoch Coding (AZTEC), Turning Point (TP) and Coordinate Reduction Time Encoding Scheme (CORTES). Section 3.3, Lossless encoding techniques, mentions entropy coding methods, such as Huffman coding, dictionary coding methods, such as Lempel-Ziv-Welch (LZW) coding and the DEFLATE algorithm, basic modulation techniques, such as Differential Pulse Code Modulation (DPCM), predictive coding techniques, such as Linear Predictive Coding (LPC) and “AD-HOC” techniques, such as Run-Length Encoding (RLE). Section 3.4, Lossy encoding techniques, describes transform-based methods are discussed, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). Section 3.5, Compression and distortion metrics, enumerates the metrics used to evaluate the compression techniques in terms of compression and distortion. Finally, with the existence of two types of encoding, lossless and lossy, comes the need to select the appropriate situations to apply each technique, as explained, in section 3.6, Comparison between lossless and lossy compression.

3.1 Pre-processing the data

Digital signals, right after being acquired, are not clearly understandable for humans or machines, having high amplitudes, noise, offsets, among other deformities. This raw data could be difficult to work with as these factors could be amplified, degrading the signal and making impossible to understand the data. Before carrying out signal processing, it is necessary to perform some simple actions that can help to analyse the signal.

3.1.1 Filtering

Filtering is the action of removing some band of frequencies from the signal. For low frequency signals, like biological signals, there is a great concern about low frequency noises.

There are four types of filters, according to their cut-off band [12]: Low-Pass, High-Pass, Band-Pass and Band-Reject filters. Low-Pass Filters (LPF) and High-Pass Filters (HPF) cut the bands, above and below the cut-off frequency (f_c), respectively. Band-Pass Filters (BPF) have two cut-off frequencies. These two cut-off frequencies define the lower and upper limits of the filter, keeping only the frequencies between these two. Band-Reject Filters (BRF) are the opposite of Band-Pass Filters, cutting a specific frequency band – the rejection band.

Figure 8 represents the frequency responses of the mentioned filters.

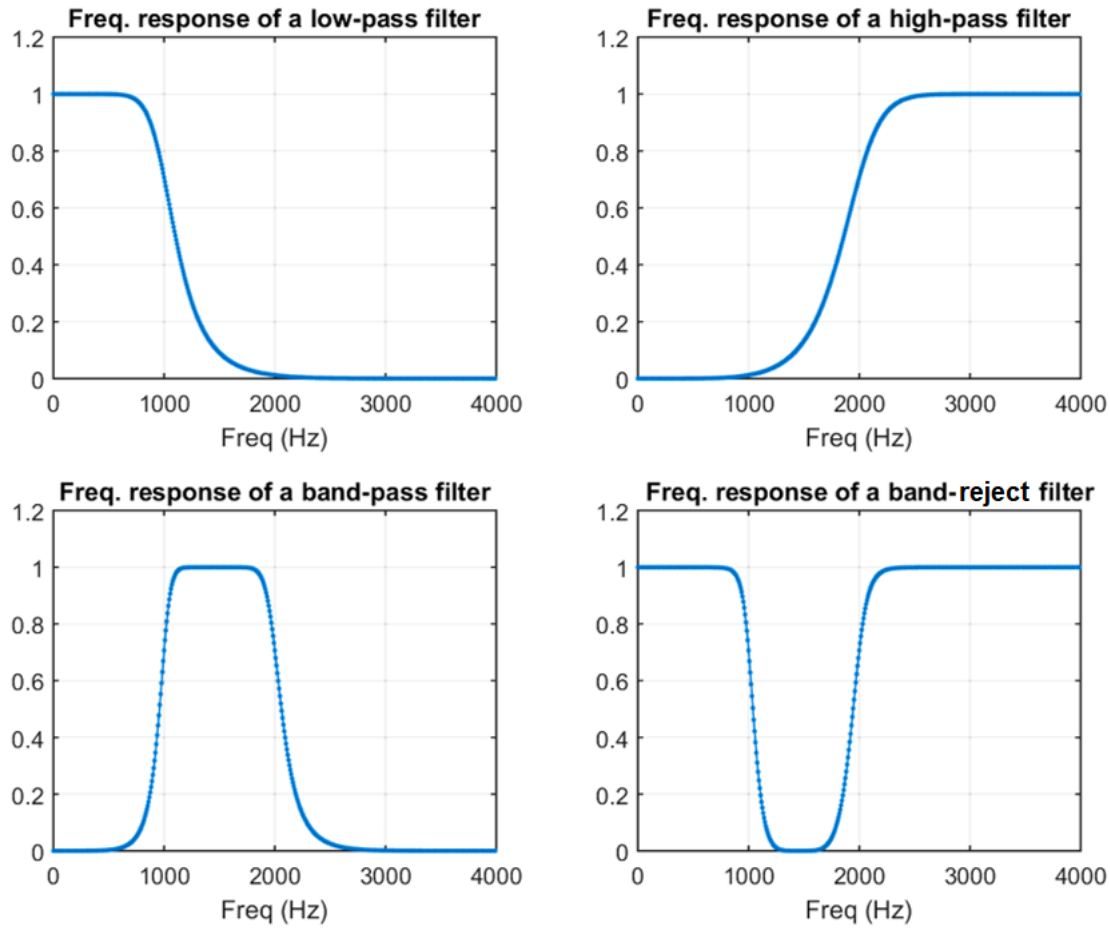


Figure 8 – Representation of an LPF (top left), HPF (top right), BPF (bottom left) and BRF (bottom right) filters' frequency responses.

Besides the filtering type, the filtering can be done using two types of digital filters [12] [13] [14]: Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters. FIR filters are non-recursive filters, as the output only depends on the present input and on a delayed input, characterised for having a linear phase response across the frequency spectrum. IIR filters have the output depending on the present input, on a delayed input and on a delayed output, being a recursive filter and having a non-linear phase response. This introduces distortion in the output signal but, on the other hand, IIR filters are better in computational cost, being more efficient, and being able to have lower filter order to obtain the same frequency response as a FIR filter.

The order of the filter is related to the number of the delayed samples used to compute the output. As the number of delayed samples used increases, the order of the filter increases as well, making the filter with sharper transitions.

Figure 9 and Figure 10 represent, respectively, the block diagrams and the phase responses of FIR and IIR filters.

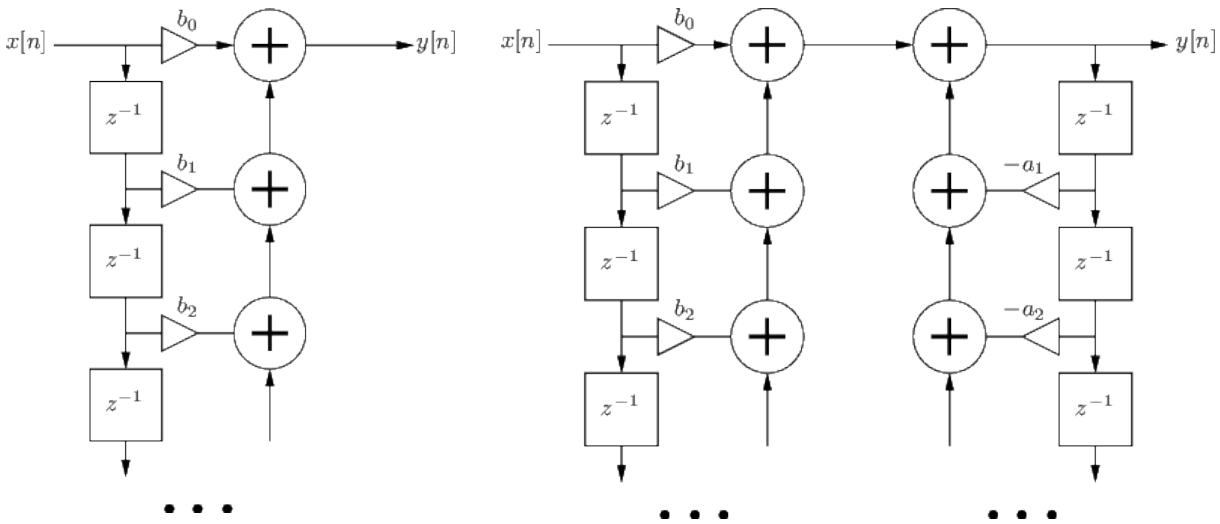


Figure 9 – Block diagram of a FIR filter (left) and an IIR filter (right).

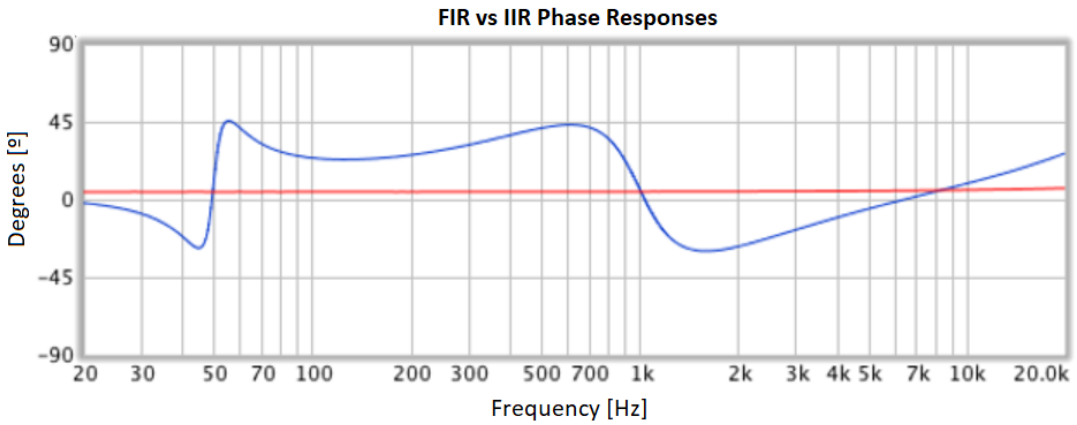


Figure 10 – Examples of FIR (red) and IIR (blue) filters' phase responses [14].

The alternating electric current, which provides energy to the equipment, has an associated frequency of 60 Hz, in most of the American continent, and 50 Hz, for the rest of the world. It can be said that the signal acquisition devices, when connected to this current, end up being over its influence. Since the ECG signal is a low-frequency signal, this means that these 50 Hz or 60 Hz frequency components will be present in the acquired signals [15].

3.1.2 Amplitude scaling

The amplitude scaling is a technique that significantly reduces the amplitude of a signal in a simple way. This technique consists in dividing a signal, analogue or digital, by a scaling factor. This factor must be greater than 1 in order to have a reduction in the maximum and minimum amplitudes of the signal, reducing its dynamic range. For a digital signal, the scaling coefficient can be calculated by:

$$C_{scaling} = \frac{2^{Nbits_q}}{2^{Nbits_t}} \quad \textbf{(1)}$$

where $Nbits_q$ is the number of bits of the input quantised signal and $Nbits_t$ is the number of the desired bits per sample.

This technique could be used for compression, since reducing the signal's amplitude, reduces the data bit-range, reducing the storage space needed. However, when dividing a digital signal by a coefficient, the new scaled samples could have a decimal part, that may be discarded by the processors, disabling the opportunity to rescale the signal to its original amplitude.

3.2 Direct time-domain techniques

Some of the studied compression methods in ECG data are the direct time-domain techniques. These methods are often used in heartbeat detection and counting, achieving good compression ratios but failing in the reconstruction of the signals, introducing distortion to the ECG signal.

3.2.1 Amplitude Zone Time Epoch Coding

The Amplitude Zone Time Epoch Coding (AZTEC) [16] [17] algorithm converts raw ECG samples into plateaus and slopes. The AZTEC plateaus are produced by utilising Zero-Order

Interpolators (ZOI) [17]. The stored values for each plateau are the amplitude and length values of the line. The length value is the number of samples that can be interpolated within an aperture.

The production of an AZTEC slope starts when the number of samples needed to form a plateau is less than 3. The slope is saved whenever a plateau of 3 samples or more can be formed. The stored values for the slope are the duration (number of samples of the slope) and the final elevation (amplitude of the last sample point).

The signal reconstruction is achieved by expanding the AZTEC plateaus and slopes into a discrete sequence of data points.

3.2.2 Turning Point

The main purpose of the Turning Point (TP) [16] [17] data reduction algorithm is to reduce the sampling frequency of an ECG signal from 200 to 100 Hz, without weakening the elevation of large amplitudes, given by the QRS complex.

The algorithm processes 3 data points at a time: a reference point (X_0) and two consecutive data points (X_1 and X_2). Only the reference point (X_0) and one data point, X_1 or X_2, is preserved, depending on which point best conserves the slope of the original 3 points.

3.2.3 Coordinate Reduction Time Encoding Scheme

The Coordinate Reduction Time Encoding Scheme (CORTES) [16] [17] algorithm is a hybrid of AZTEC and TP. CORTES applies TP to high frequency regions, such as QRS complexes, while applies AZTEC to the isoelectric regions of the ECG signal. The AZTEC and TP are applied in parallel to the incoming sampled ECG data. Whenever an AZTEC line is produced, a decision based on the length of the line is used to determinate whether the AZTEC data or the TP data is to be saved. If the line is longer than an empirically determined threshold, the AZTEC line is saved, otherwise, the TP data are saved. Only AZTEC plateaus are generated, AZTEC slopes are not produced.

The reconstruction is achieved by expanding the AZTEC plateaus into discrete data point and interpolating between each pair of the TP data.

3.3 Lossless encoding techniques

Lossless encoding techniques are often named source coding techniques. The primary objective of source coding is to represent a signal with a reduced number of binary symbols without distortion and can be classified in two major groups: entropy coding and dictionary-based coding. The lossless compression is achieved by removing the redundancy often found in raw data.

3.3.1 Huffman coding

The Huffman [18] [19] [20] [21] coding is an entropy source coding method that produces a variable length code. This coding is based on the probabilities of occurrence of each sample, where the most likely sample is encoded with fewer bits.

The Huffman encoding consists in a construction of a tree where the symbols to be coded represent the branches and are arranged by probability in descending order, from the top to the bottom, as illustrated by Figure 11.

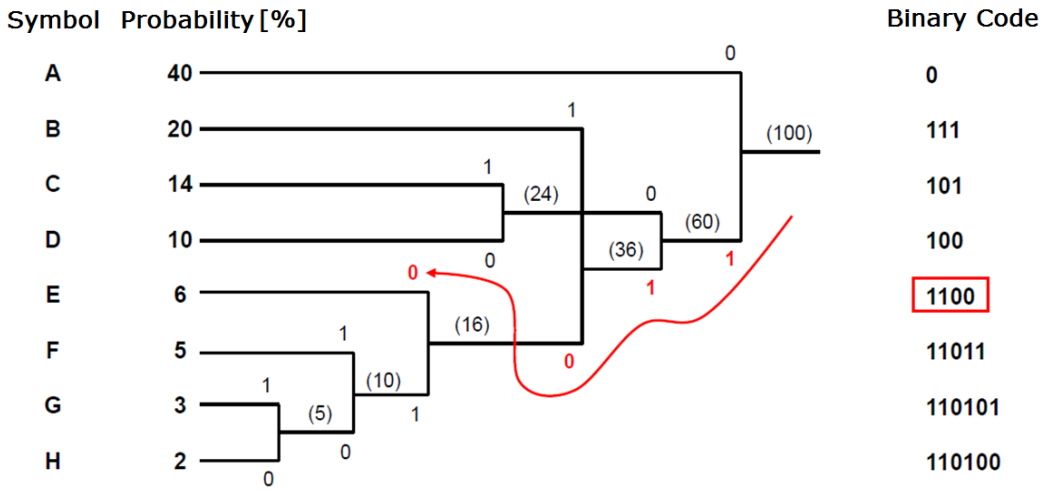


Figure 11 - Example of a Huffman coding tree.

The method consists in successively adding up the two lowest probabilities by creating a new symbol with a probability equal to the sum of the two. These two symbols are assigned with the binary value of "0" and "1" to the one with less and greater occurrence, respectively. This methodology ends when the sum of the two lowest probabilities is 100%, creating a node called the root node. After this procedure is finished, each symbol is represented by the

set of "0" or "1" from the root node to the symbol to be encoded, as exemplified in Figure 11 with the red line.

To perform the decoding, it is necessary that the receiver has the same tree used for encoding. The encoder matches the binary symbol received with the respective coded symbol. Without the coding tree it becomes impossible to reconstruct the original signal, offering some security to the information since only the receivers with the encoding tree are capable to decode the data.

3.3.2 Lempel-Ziv-Welch coding

The Lempel-Ziv-Welch (LZW) [18] [19] [20] [22] algorithm derives from the Lempel-Ziv 78 (LZ78) [23] algorithm, used in the compression of GIF image files, known as a dictionary-based method. The LZW dictionary is initially created with a set of code-words of n bits and, as the encoding is done, new entries will be added to the dictionary with combined sets of the n-bit code-words. Whenever a pair of symbols is read, the algorithm tries to find it in the dictionary, if the pair is not in the dictionary, the algorithm adds it for a future use. As the dictionary holds more entries, the algorithm becomes faster and with better compression ratio.

The decoder builds the same dictionary created by the encoder, having only the same n-bit initial inputs as in the encoder. The decoder reads a pair of encoded symbols, if the second encoded symbol is in the dictionary, it translates it to the original value, if not, it adds the pair of the two symbols to the dictionary as a new entry and reads a new symbol. This way, the new entries of the dictionary are added in the same way that was done by the encoder, making the data decoding possible.

With the amount of data that could be encoded, and with a limited dictionary size, it is expected that the dictionary could be filled up before all the data is encoded. This means that after the dictionary becomes full, the new entries cannot be added to the dictionary, limiting the compression performance of the algorithm. To overcome this, the algorithm replaces the oldest entry in the dictionary by the new one, never stopping the process of adding new entries.

Figure 12 represents the flowcharts that describe the various steps of compression and decompression performed by the LZW algorithm. The variable CHAR represents a n-bit code-word and the variable STRING is a sequence of CHARs.

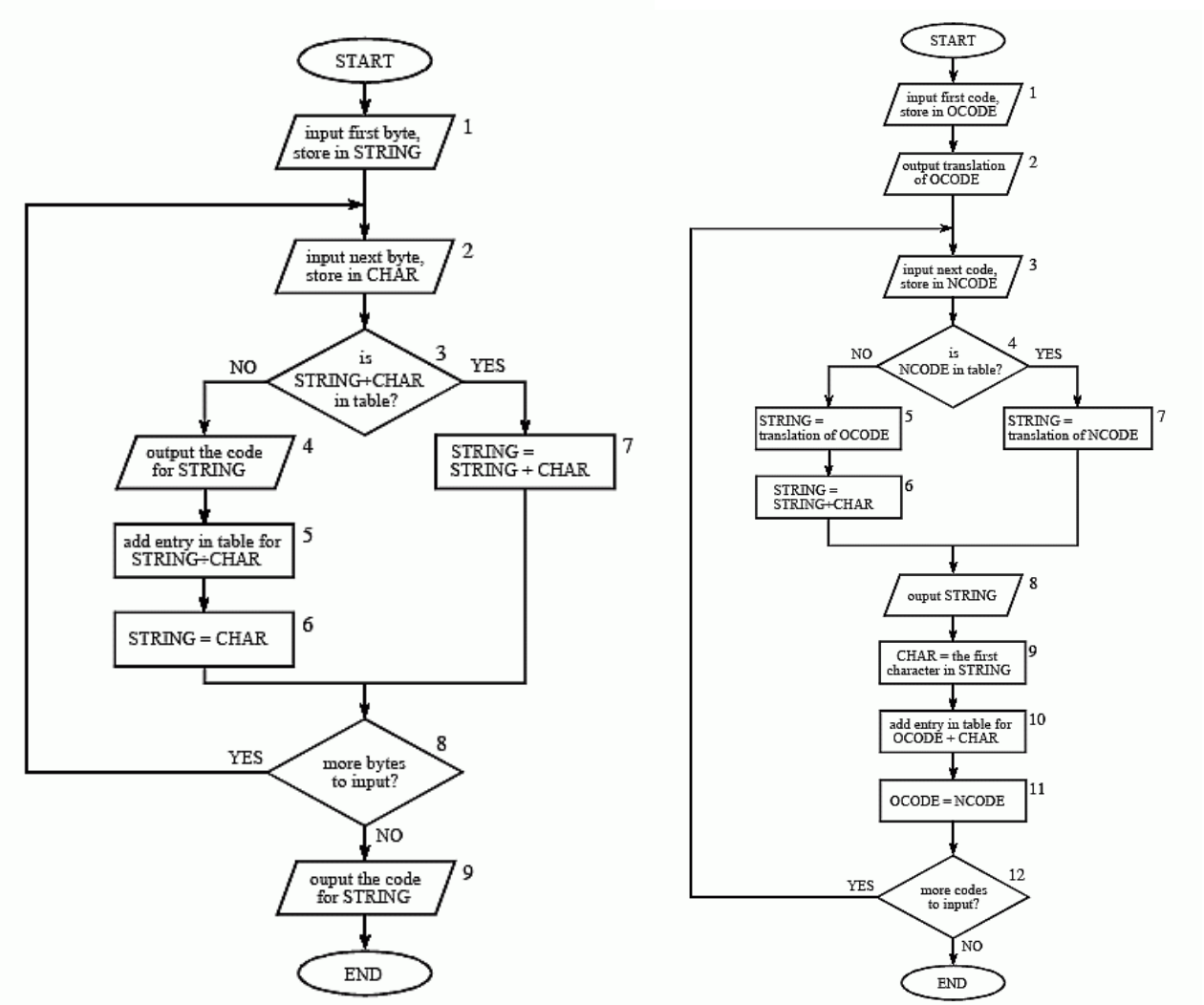


Figure 12 – Flowcharts of LZW compression (left) and decompression algorithms (right) [19].

3.3.3 DEFLATE algorithm

The DEFLATE [24] compression algorithm is based on the Lempel-Ziv 77 (LZ77) [23] algorithm, for duplicate string elimination, followed by Huffman coding, for bit reduction. This compression method is used when compressing files into a ZIP file extension.

The DEFLATE algorithm consists in dividing all the input data into blocks. For each block, the LZ77 algorithm finds repeated substrings and replaces the next occurrence of that substring by a pointer to the previous substring, with a pair of coordinates – distance and length. If a substring does not occur again it is not compressed, and the original sequence is kept. The original sequences and the match lengths are compressed with one Huffman tree and the match distances are compressed with another tree.

The Huffman trees created are encoded to go along with the rest of the data, so the receiver does not need to build the Huffman tree to decode the data. The Huffman trees

are transmitted by their code-lengths. These code-lengths are put all together into a sequence of numbers between 0 and 8 and once they are assembled they are compressed with Run-Length Encoding (RLE).

Once the receiver gets this encoded message it decompresses by doing the inverse actions by the reverse order. It decodes the Huffman trees, so it can get the match lengths and distances and the original sequences.

3.3.4 Differential Pulse Code Modulation

Pulse Code Modulation (PCM) [25] is the method used to convert an analogue signal to digital. This process consists in three steps – Sampling, Quantisation and Encoding.

Differential Pulse Code Modulation (DPCM) [18] [25] is a PCM technique that takes advantage of the resemblance between consecutive samples of a low-frequency signal. With this coding method, it is possible to represent a sample knowing the previous one, only being required to transmit the difference between two consecutive samples.

This technique can be used for signal compression if applied to a low-frequency signal, once the differences between consecutive samples have smaller values than the original amplitude of the signal. In high-frequency signals, this difference between consecutive samples could be greater than the original amplitude of the signal, not having any advantage for compression.

Figure 13 shows the block diagrams of the DPCM encoder and decoder.

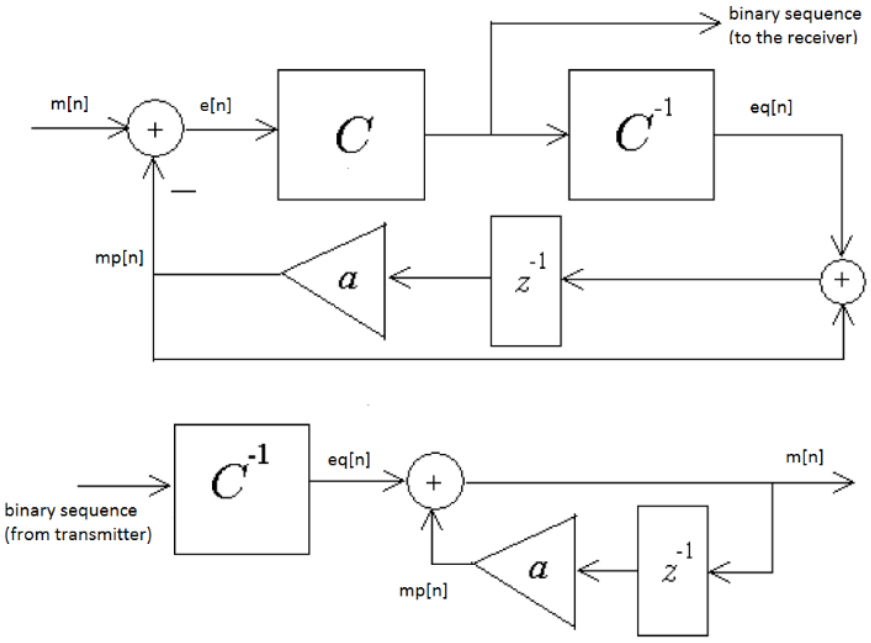


Figure 13 – Block diagrams of a DPCM encoder (top) and decoder (bottom). [Adapted from 32]

3.3.5 Linear Predictive Coding

Linear Predictive Coding (LPC) [26] [27] is a type of predictive encoding used to process audio and speech signals. This type of coding is characterised by being an Auto-Regressive (AR) model [27], that is, a sample is linearly dependent on the previous samples.

The main idea of LPC is the transmission of an error $e[n]$, which results from the subtraction between the original signal $m[n]$ and the predicted signal $p[n]$. The predicted signal is calculated by multiplying each original sample by a coefficient a_i. These coefficients are extracted by the autocorrelation of the signal, and the number of coefficients c to estimate the predicted signal depends on the purpose. Both signals mentioned above can be represented by:

$$\boldsymbol{p[n] = a_1 m[n-1] - a_2 m[n-2] - \cdots - a_c m[n-c]} \quad \textbf{(2)}$$

$$\boldsymbol{e[n] = m[n] - p[n]} \quad \textbf{(3)}$$

where n represents the sample to be encoded. With this method, instead of transmitting a signal with a dynamic range equal to $m[n]$, the error $e[n]$ is transmitted, which will have a dynamic range significantly lower than $m[n]$.

The decoding is done considering the error $e[n]$, the coefficients a_i, and the first c samples of the original signal. With this, $p[n]$ is calculated with the same equation as in the transmitter and the $p[n+i]$ is calculated with the previous predicted samples. The signal $m[n]$ is reconstructed with the following expression:

$$\boldsymbol{m[n] = p[n] + e[n]} \quad \textbf{(4)}$$

LPC and DPCM are two techniques that are quite related. It can be said that the LPC is equivalent to DPCM technique when there is only one coefficient a_1 and that coefficient is equal to 1.

3.3.6 Run-Length Encoding

The Run-Length Encoding (RLE) [18] [20] technique is a very simple method used to compress simple image files such as in the Bitmap (BMP) format, such as icons and animations. This technique consists in replacing the sets of repeated successive samples by the repeated sample value attached to the number of times that sample is successively repeated.Given the following set of samples:

35,35,35,35,35,47,47,12,12,12,12,12,12,12,12,96,51,51,51,51,51,47,47,47,47

this set of samples can be written this way after RLE:

$$35,4,47,1,12,7,96,0,51,4,47,3$$

where the even samples represent the number of times the odd sample that precedes it is repeated successively in the original set. If an even sample is equal to 0, it means that the previous sample was not successively repeated in the set of samples.

3.4 Lossy encoding techniques

Transformation-based methods are the most used techniques to perform lossy encoding of audio and image data. The transformation methods are lossless, but they are usually applied to enable better coefficient quantisation, introducing loss, which results in a lower quality output with high compression ratios. These techniques consist in discarding less significant information, which tends to be irrelevant to the human perception of the multimedia content.

Figure 14 represents the block diagram of the lossy encoding techniques.

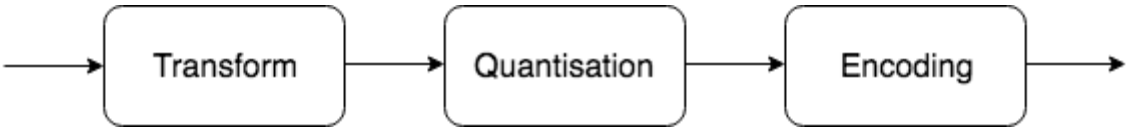


Figure 14 – Block diagram of the lossy encoding techniques.

3.4.1 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) [18] [19] [28] is the representation of a set of finite points through the summation of several cosine functions. The DCT method is used in various applications such as in lossy compression of audio signals, such as the MP3 format, where high frequencies are discarded.

The DCT is a technique similar to the Discrete Fourier Transform (DFT) [18] [29] with the exception that it uses only real values and projects the input signal on a cosine basis.

The DCT coefficients for a one-dimensional signal are computed by:

$$\boldsymbol{C(u) = a(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right]} \quad \textbf{(5)}$$

where $f(x)$ is the input sample to be transformed, N represents the total number of samples and $a(u)$ is expressed as:

$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & , u = 0 \\ \sqrt{\frac{2}{N}} & , u > 0 \end{cases} \quad \textbf{(6)}$$

The reconstructed signal can be defined by:

$$f(x) = \sum_{u=0}^{N-1} a(u)\, C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad \textbf{(7)}$$

There are other types of DCT adapted for each purpose [18], for example, for the compression of JPEG images, where there is a signal matrix, it is used a two-dimensional DCT.

3.4.2 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) [7] [18] [21] [26] is the decomposition of a signal when passed through an HPF and an LPF. With these filtering operations, two sets of coefficients are generated: approximation coefficients and detail coefficients

The approximation coefficients (cA) are generated by convolving the signal with the LPF's impulse response $g[n]$, and the detail coefficients (cD) are generated by convolving the signal with the HPF's impulse response $h[n]$, both followed by a dyadic decimation [30], usually called down-sampling.

The wavelet function provides a multi-resolution representation of signals with a collection of these two types of coefficients, each of them provides information about signal characteristics like location in time and frequency.

The advantage of DWT over Fast Fourier Transform (FFT) [28] [29] is that it performs multi-resolution analysis of signals with localisation. As a result, the DWT decomposes a digital signal into different sub-bands so that the lower frequency sub-bands will have a finer frequency resolution and a coarser time resolution compared to the higher frequency sub-bands.

Figure 15 represents a flowchart of the usage of DWT to extract level 3 coefficients.

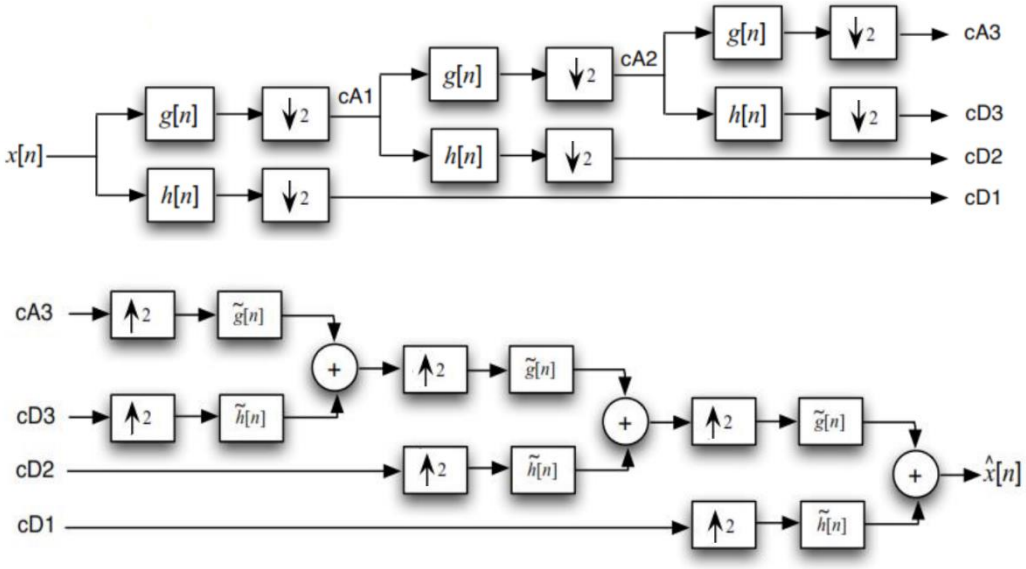


Figure 15 – Flowchart of forward DWT (top) and inverse DWT (bottom) [21].

3.5 Compression and distortion metrics

The signal transformations and encoding to require less storing space or bandwidth, can be evaluated in terms of compression and introduced error, as compared to the original signal. These errors can be classified by measuring parameters such as distortion, amplitude or noise differences. There are two types of metrics: compression metrics that include the Compression Ratio (CR) and the distortion metrics that are only applied to lossy techniques such as Root-Mean-Squared Error (RMSE) or Signal-to-Noise Ratio (SNR).

3.5.1 Compression Ratio

Compression algorithms reduce the number of bits to be stored or transmitted, by removing redundancies or discarding not so relevant data from the signals.

The Compression Ratio (CR) [28] [31] is one of the most used metrics in signal compression and measures the data reduction achieved by a given compression method. When testing a compression method, it is intended to obtain high CR while maintaining acceptable signal quality. The CR is the ratio between the length of the original signal and the length of the compression signal, in bits, expressed as follows:

$$\boldsymbol{CR = \frac{Original\ signal\ length_{[bits]}}{Compressed\ signal\ length_{[bits]}} : 1} \qquad \textbf{(8)}$$

3.5.2 Root-Mean-Squared Error

The Root-Mean-Squared Error (RMSE) [28] [31] is one of the most used distortion metrics to measure differences between values, representing the differences between input samples and output samples. It represents how far the output samples are from the input and it is calculated by the squared root of the summation of the mean of the squared differences between original samples and compressed samples. The lower the RMSE, the closer are the input and output samples. The RMSE expression can be defined as follows:

$$\boldsymbol{RMSE(S_o, S_r) = \sqrt{\sum_{k=1}^{N} \frac{\left(S_o(k) - S_r(k)\right)^2}{N}}} \quad \textbf{(9)}$$

where $S_o(k)$ is the original k sample, $S_r(k)$ is the reconstructed k sample and N is the signal length in samples.

3.5.3 Signal-to-Noise Ratio

The Signal-to-Noise Ratio (SNR) [31] [32] measures the quality of a signal affected by noise. Quantisation is the process of converting an analogue signal to digital and the output digital signal comes out with distortion, called quantisation noise. The ratio from the signal to the quantisation noise is called quantisation SNR (SNR_q) and is defined as:

$$\boldsymbol{SNR_{q_{[dB]}} = 6,02R_{[bits]} + 10\log_{10}\left(\frac{3P_{[W]}}{V^2_{[V]}}\right)} \quad \textbf{(10)}$$

where R is the number of quantisation bits, P is the normalised power of the signal and V the maximum quantisation value in Volts [V].

In a simulation environment, once the signal is transmitted, it is possible to calculate the difference between the signal present at the receiver and the original quantised signal. This difference between the two signals is the encoding noise. The power of the signal (P) divided by the power of that noise (N) represents the transmission SNR (SNR_t):

$$\boldsymbol{SNR_{t_{[dB]}} = 10\log_{10}\left(\frac{P_{[W]}}{N_{[W]}}\right)} \quad \textbf{(11)}$$

3.6 Comparison between lossless and lossy techniques

Compression methods can be classified in terms of the presence of loss, the amount of loss and time spent for data processing. The decision of the most suitable compression method to solve the problem may depend on this characteristic.

Lossless compression methods are the adequate choice to compress the signal without changing the original samples. These compression techniques are only applicable in a minority of the cases.

On the other hand, lossy methods can be applied in most cases and are recommended in situations in which some loss can be introduced on the data. However, the associated loss may not be tolerated, depending on the purpose of application, meaning that the loss level introduced by the encoding process must be controlled. Table 3 summarises the discussed compression methods according to their type.

Table 3 – Type overview of the discussed compression methods.

Technique Group	Coding Name	Losses
Direct time-domain techniques	AZTEC	Lossy
	TP	Lossy
	CORTES	Lossy
Source coding techniques	Huffman	Lossless
	LZW	Lossless
	DEFLATE	Lossless
	DPCM	Lossless
	LPC	Lossless
	RLE	Lossless
Transform coding techniques	DCT	Lossy
	DWT	Lossy

Chapter 4

Wireless Technologies

Nowadays, there is a wide range of wireless technologies that allow communication to a large diversity of applications. For small applications, with small data rates and small working radius, there is a main concern about the power consumption.

This chapter is composed by three sections, each one explaining the basic concepts, protocols and characteristics for three different technologies, and the fourth section compares these three technologies. Firstly, in section 4.1, Bluetooth® Low Energy, it is described the Bluetooth protocol designed for low-power applications. Section 4.2, ZigBee, addresses some technologies that developed for Wireless Personal Area Networks (WPAN). Section 4.3, ANT, describes the protocol that enables devices to run for years with a single battery cycle. Lastly, Section 4.4, A comparison on wireless technologies, summarises the main characteristics of these three technologies.

4.1 Bluetooth® Low Energy

The Bluetooth® technology [33] has emerged as a way to replace wired communications from computer peripherals such as mice, keyboards and headsets. At this time, Bluetooth is used in a wide range of health applications, such as blood pressure monitors and blood glucose meters, or in the fitness area, such as speed sensors or heart-rate meters.

Considering that most of the Bluetooth devices are battery-powered there is a growing need to reduce the energy consumption of this technology. The Bluetooth Low Energy (BLE) [34] [35] [36] technique addresses this issue.

BLE stack can be represented by three independent layers:

- Link layer – master-slave relationship;
- GAP (Generic Access Profile) layer – central-peripheral relationship;
- GATT (Generic Attribute) layer – client-server relationship.

At the Link layer, the master acts as a Scanner and the slave as an Advertiser. The Advertiser continuously sends basic information about itself and once the Scanner receives the information it needs, it tries to connect to the Advertiser. When the Advertiser accepts, the connection is established.

In the GAP layer, the central is the one initiating a connection, establishing connection intervals and other connection parameters. Almost everything is initiated by the central, for example, a connection pairing or parameter update. Although a peripheral can request the central to perform these actions, it is always up to the central to decide what to do.

The roles of the GATT layer come into play once a connection has been established. The GATT Server can, in general, be described as the device sitting on information or data, while the GATT Client is the one seeking this data. The GATT Client sends requests for information to the GATT Servers, which respond with the information requested by the GATT Client.

BLE allows communications up to 100 meters, theoretically, in the 2.4 GHz frequency band where transmission rates can go up to 2 Mbit/s but, for most applications, the required bit rate is usually around 0.3 Mbit/s. The average power consumption with this technology is around 15 mA of current, reducing the power consumption to about half as compared to standard Bluetooth.

4.2 ZigBee

ZigBee® [35] [36] [37] is a technology specially built for control and sensor networks on the IEEE 802.15.4 standard for Wireless Personal Area Networks (WPANs), and it is a product from the ZigBee Alliance. This communication standard defines Physical and MAC layers to handle many devices at low-data rates.

ZigBee's WPANs operate mostly in the 2.4 GHz frequency band at a maximum bit rate of 250 kbit/s for a periodic two-way transmission of data between sensors and controllers.

This technology supports different network configurations for master to master or master to slave communications and different topologies as mesh, star, and cluster tree. The system structure consists of three different types of devices such as ZigBee coordinator, Router and End device. Every ZigBee network must have at least one coordinator which acts as a root and bridge of a network. The coordinator is responsible for handling and storing the information while receiving and transmitting data. ZigBee routers act as intermediary devices that allow data to pass through them to other devices. End devices have limited functionality to communicate with the parent nodes to preserve battery power.

The two-way data transmission of ZigBee can be transferred in two modes: beacon and non-beacon modes. In a non-beacon mode, the coordinators and routers are continuously waiting for incoming data, having more power consumption. The access control uses a non-slotted Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) [38], which means that nodes must sense the radio channel before starting any transmission. If the channel is

busy, the transmitting device must wait a random time before listening to the radio channel again. In a beacon mode, when there is no data communication from end devices, the routers and coordinators enter into a sleep state. On a periodical basis, the coordinator wakes and broadcasts a special frame, called beacon, in intervals that can vary from 15 milliseconds to 252 seconds, to check if new data is available at the end devices. In this mode, the coordinator can reserve timeslots to a particular node to guarantee the quality of the service.

ZigBee offers a low-cost and a low-powered network widely deployed for controlling and monitoring applications with a range up to 100 meters. This communication system is less expensive and simpler than the other proprietary short-range wireless sensor networks.

4.3 ANT

ANT™ [35] [36] [39] is an ultra-low power wireless networking protocol which enables objects from everyday life to connect with each other easily in practical and valuable ways. In millions of consumer devices today, the ANT protocol is a proved solution for rapidly evolving applications in the Internet of Things (IoT) paradigm.

Enabling connected devices to run for years with 24 hours per day operation cycles on a coin cell battery, ANT pioneered the reliable ultra-low power wireless solution of choice for personal sport and fitness products. The ANT protocol is the base technology that powers the active ANT+ sport, fitness, and wellness product ecosystem.

The ANT protocol's ability to easily connect devices in flexible ways serves a broad range of applications. Use cases including simple device-to-device links, large numbers of smart objects linked wirelessly, and many more can be built with ANT.

This protocol is set up to use a single 1 MHz channel for multiple nodes thanks to a Time-Division Multiplexing (TDM) [40] technique. Each node transmits in its own time slot. Basic message length is 150 µs, while the message rate, or the time between transmissions, will range from 5 milliseconds to 2 seconds with an 8-byte payload per message. A 16-bit Cyclic Redundancy Check (CRC) [32] is used for error detection. Up to 65,536 timeslots can be accommodated per channel. If interference is encountered, the node transceivers can switch channels.

ANT accommodates three types of messaging: broadcast, acknowledged, and burst. Broadcast is a one-way communication from one node to many nodes. The receiving nodes transmit no acknowledgment but may still send messages back to the transmitting node. This technique is suited to sensor applications and is the most economical method of operation. In acknowledged messaging, the receiver confirms the reception of data packets. Although the

transmitter is informed about success or failure, it does not perform any retransmission. This technique is suited to control applications. The burst messaging is a complete multi-message transmission technique that uses the full data bandwidth. The receiving node acknowledges receipt and informs the existence of any corrupted packets and, if any, the transmitter should retransmit those corrupted packets. This technique is suited to data block transfer where the integrity of the data is crucial.

Companies like Nike®, Adidas®, Garmin® and Geonaute® are using ANT for heart rate monitors, cycling power meters as well as distance and speed monitors.

4.4 A comparison on wireless technologies

The outside market has a large variety of technologies and devices with specific characteristics that makes one more suitable than other for a given application. Looking at the various characteristics of the three wireless technologies discussed is possible to conclude that the choice between each technology focuses more in the type of the topology to be used in the application and in the number of devices.

Usually, low-power applications don't require high data rates neither large coverage radius. The channel bandwidth of each technology limits the number of simultaneous devices in the network.

ANT can have more linked devices, so it is more used in large network topologies, while ZigBee is used for small and low-cost networks. BLE can be used in network topologies, in which there is a Client requesting information from numerous Servers, but it can be used in a Peer-to-Peer (P2P) topology. Table 4 summarises some of the characteristics for the above-mentioned technologies.

Table 4 – Comparison of some characteristics of BLE, ZigBee, and ANT.

Technology	BLE	ZigBee	ANT
Theoretically coverage radius [m]	100	100	100
Frequency Bands [GHz]	2.4 to 2.483	2.4 to 2.483	2.4 to 2.483
Channel bandwidth [MHz]	2	5 (2 MHz used)	1
Number of channels	40	16	78
Topology types	Scatter net, P2P	Mesh, Star, Tree	Mesh, Star, Tree, P2P
Maximum Bit Rate [kbit/s]	2000	250	1000
Power per bit [µW/bit]	0.153	185.9	0.71

Chapter 5

Learning from Data

Machine learning is a term that is spoken more widely nowadays, and it is used in most of daily-use applications, like Google search, e-mail SPAM filters, product recommendations on online stores, among others.

Teaching a machine to learn how to correctly classify data demands a large dataset and good learning algorithm. This chapter explains the steps to create good learning algorithms and how they operate. Section 5.1, Feature engineering, describes the importance to machine learning of extracting features from a large dataset and how to do it in a correct way. Section 5.2, Classification problems, discusses some of the problems that should be taken into account while building a machine learning algorithm. Section 5.3, Machine learning algorithms, enumerates some machine learning techniques applied to ECG and accelerometer data classification, and how they operate. Lastly, section 5.4, Performance evaluation, describes the methods to appraise the performance of a machine learning technique and the techniques to improve the results.

5.1 Feature engineering

Machine learning algorithms usually do not interpret raw data but use features as a way to describe the data. A feature is a descriptor taken from the data.

Features are the input of machine learning algorithms and depend on the type of data to classify. It is necessary to realise the most influent topics, so the algorithms can predict an output based on those specific topics. There are some generic features, that can be used in most type of data, like standard deviation or entropy, but usually there are specific features that describe specific characteristics of a type of data.

To give an algorithm some features, it is necessary three phases [41] [42]: feature extraction, feature transformation, and feature selection. Feature extraction is when you define the features that could possibly describe data, having a considerable influence in the output classes. When you have collected enough features, it could be useful doing a feature transformation. Feature transformation consists in transforming the feature's value, so they can have a similar order of impact on the output doing, for example, feature normalisation or Principal Component Analysis (PCA) [42]. Since a large number of features could lead to a

non-generalised solution, it could be useful to discard some of the features with less impact in the output, called feature selection.

With an adequate subset of features it is possible to run any machine learning algorithms to classify the incoming data automatically. There is not a perfect way to get enough or relevant features so many tests should be done to acquire the best solutions.

5.2 Classification problems

Handling a large dataset to teach a machine learning from that dataset is not always simple. Sometimes, it is needed some caution when working with features or a dataset that is not balanced, since it can end up influencing the performance of the classifier, turning the results unreliable or even meaningless.

5.2.1 Over-fitting and Under-fitting

The output of a machine learning algorithm depends on the chosen features and when the feature selection is not done properly it could lead to two main problems [43] [44]: over-fit and under-fit.

Over-fit takes place when the model is very dependent on the training data, fluctuating according to each data small variation. Under-fit is the opposite situation of over-fit, and it happens when the model ignores the relationships between the input and the output of the data.

An over-fitted model usually has high variance and low bias, and an under-fitted model, has low variance and high bias. Variance is how much the model changes in response to the training data, having a low capability to classify correctly the data that are not equal to the training set. Bias is the flip side of variance, and represents the assumptions made by the model that lead to ignore the training data. In any model, it is necessary to ensure a trade-off between bias and variance to achieve the best balance and performance. Figure 16 shows three examples of predicted outputs, each with a different type of fitting.

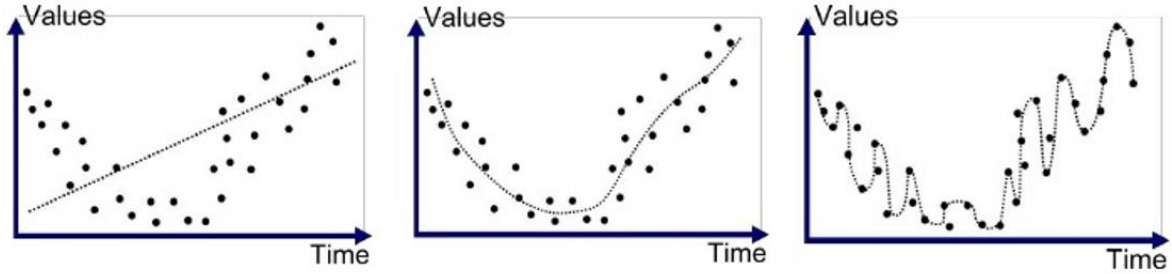


Figure 16 – Example of models with under-fitting (left), good fitting (middle) and over-fitting (right) [44].

5.2.2 Class imbalance

Many learning systems assume that training sets used for learning are balanced, thus, the patterns from the output classes have nearly the same number of examples in the dataset. However, this is not always the case and one class can be represented by a large number of examples, while the other is represented by a few. This problem is known as class imbalance [43] [45].

Class imbalance can be exemplified in a medical context, when trying to teach a machine to classify if somebody has a particular rare disease. The number of patients without that disease will be wider than the number of patients that will have it. If the classifier predicts all the patients as without disease, it will have a large success in prediction, however, that success is meaningless due to the imbalance in the dataset.

There are two simple way to fix the imbalance: oversampling the minority class or under-sampling the majority class. Oversampling is the technique to introduce new samples in the dataset. Those samples can be duplicated from the existing ones or synthesised by combinations of different minority samples with a method called Synthetic Minority Over-sampling Technique (SMOTE) [45]. Under-sampling is the technique to remove samples from the majority class, however, this can end up leaving out important samples that provide important differences between classes.

These techniques could seem not so worthy but, in most cases, they can solve the class imbalance problem, making the classifier more reliable, when applied in the correct way.

5.3 Machine learning algorithms

There are two types of machine learning [43] [46]: supervised machine learning and unsupervised machine learning. Each type of learning uses different approaches and depends on the type of dataset used to training the algorithm.

Supervised learning is when you have input variables and an output variable, and the algorithm learns the mapping function from the input to the output. The objective is to estimate the mapping function so when new input data that arrives, the algorithm can predict the output for it. There are two types of problems using supervised learning: regression and classification. Regression is when the output is a continuous value, like a price or a weight. Classification is when the output is a tag, like drowsy or awake.

Unsupervised learning is when you only have input variables and the output classification is only left to the classifier that does not have any sense about the output. Unsupervised learning problems can be resolved using two techniques: clustering and association. Clustering is when the classifier tries to discover isolate groups in the data. Association is when it is intended to discover rules that describe a large amount of data.

For each type of dataset, there is a different way to interpret and treat the data to get the desired result.

5.3.1 Linear Regression

Linear Regression [43] [47] is one of the simplest ways to implement a supervised machine learning algorithm. This technique is also used by statisticians to predict some events.

The model representation of the linear regression is composed by a specific set of input j features (x_j^i) and a set of well-known output values (y^i) for each i^{th} training example. This model assigns a scale factor for each input feature, named coefficient (θ_j), and an independent coefficient that gives some freedom for adjustments, named bias coefficient (θ_0).

The prediction of the Linear Regression model, named hypothesis $h_\theta(x)$, is written as follows:

$$\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_j x_j} \quad \textbf{(12)}$$

The objective of linear regression is to compute various values of θ so the Mean-Squared Error (MSE), usually called Cost Function $J(\theta)$ [43], between the hypothesis and the known

output is minimum. This technique is named Gradient Descent [43] and can be represented as follows:

$$\boldsymbol{\theta_j} = \begin{cases} \boldsymbol{\theta_j} - \frac{\alpha}{m}\sum_{i=1}^{m}\left(\boldsymbol{h_\theta}\left(\boldsymbol{x^i}\right) - \boldsymbol{y^i}\right) & , \ \boldsymbol{j = 0} \\ \boldsymbol{\theta_j} - \frac{\alpha}{m}\sum_{i=1}^{m}\left(\boldsymbol{h_\theta}\left(\boldsymbol{x^i}\right) - \boldsymbol{y^i}\right)\boldsymbol{x_j^i} & , \ \boldsymbol{j > 0} \end{cases} \quad \textbf{(13)}$$

where α represents the magnitude of each step and m the number of training examples. The Gradient Descent should be applied a set of times, called iterations, and stopped when the values of θ do not change significantly for each iteration. Having higher α means that, for each iteration, the value of θ changes more significantly, pushing the algorithm in the direction of the optimal value of θ. However, if α is too high, the algorithm may never reach the optimal value of θ, hopping it several times.

Linear Regression is usually used when the output is a continuous value, but it can be used in classification problems as well. However, it has a low reliability for being very sensitive to outliers, affecting the decision boundary of the classifier, degrading its performance.

5.3.2 Logistic Regression

Logistic Regression [43] [47] is based on the sigmoid function, created by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment.

The sigmoid function is an S-shaped curve that can convert any real number to a range between 0 and 1, but never reaching those limits. The hypothesis of the logistic function can be represented as follows:

$$\boldsymbol{h_\theta(x)} = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-(\theta_0+\theta_1 x_1+\cdots+\theta_j x_j)}} \quad \textbf{(14)}$$

Logistic Regression is a supervised machine learning algorithm, usually used to estimate probabilities but can be used to solve classification problems, defining a decision boundary for classification, usually 0.5. If the hypothesis is higher than 0.5 is classified as “1” and if less than 0.5 is classified as “0”.

The values of θ can be calculated as with the gradient descent expression as in (13) but with the logistic regression hypothesis.

5.3.3 Random Forest

Random Forest (RF) [48] is a supervised learning algorithm that can be applied to regression or classification problems. The algorithm builds several classification trees and the output is the combination of the results of each classification tree.

Each classification tree is a weak learner but, considering the trees used, the classifier can be a strong learner. The whole dataset is randomly divided in different subsets, each subset will have a different associated classification tree. In each classification tree, a specific number of features are selected, and the output of that tree will be the one that provides the best binary split. This is done for all the classification trees and, at the end, the output of each classification tree is combined to have the best decision for each case.

This technique is very popular since it is fast and robust against over-fitting and imbalanced datasets. Its randomness helps to make the model more robust than with a single decision tree. The number of trees and the number of variables considered in each node should be defined according to the needs. Figure 17 represents the architecture of a Random Forest classifier.

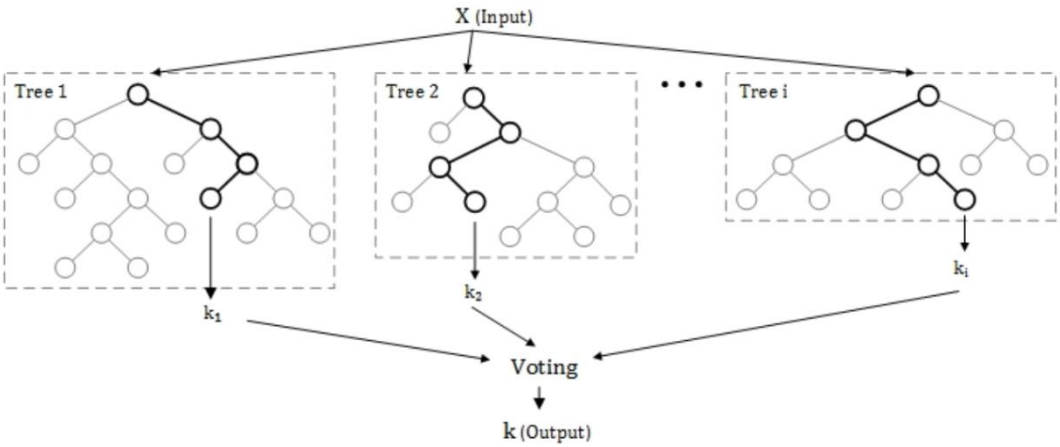


Figure 17 – Architecture of a Random Forest classifier [48].

5.3.4 Artificial Neural Network

Artificial Neural Network (ANN) [43] [49] is an information processing model based on the way the biological nervous system operates. It is composed by a large number of interconnected processing elements, named neurons, working together to solve specific problems.

The neurons in an ANN, named units, build complex networks that are remarkable for their ability to find a meaning from complicated or imprecise data, extracting patterns or detecting trends that are too complex to be noticed by humans or other computer techniques. This can be achieved with:

- Adaptive learning – the ability to learn based on data given for training;
- Self-organisation – it can create its own representation of the information during the training;
- Real time operation – computations can be carried out in parallel and hardware is being manufactured to take advantage of this capability;
- Fault tolerance – some network capabilities may be retained even with the partial destruction of the network.

ANN are a supervised learning technique composed by tree types of layers: input layer, hidden layers and output layer. Input layer is the initial point of the network, containing the training dataset. There are as many input units as features. The hidden layers are layers created by the ANN algorithm, that link the input layer to the output layer. There could be more than one hidden layer, but usually, one hidden layer is enough to solve most of problems, and the number of units in the hidden layer depends on the type of data. Each hidden unit takes in the input signals, works on them in different ways, and converts them into the corresponding output. The output layer, is the termination of network, where each unit represents an output class, defined by the summation of the weighted units of the previous hidden layer. For a binary output it could only be necessary one output unit.

Figure 18 represents a diagram of an ANN with two hidden layers.

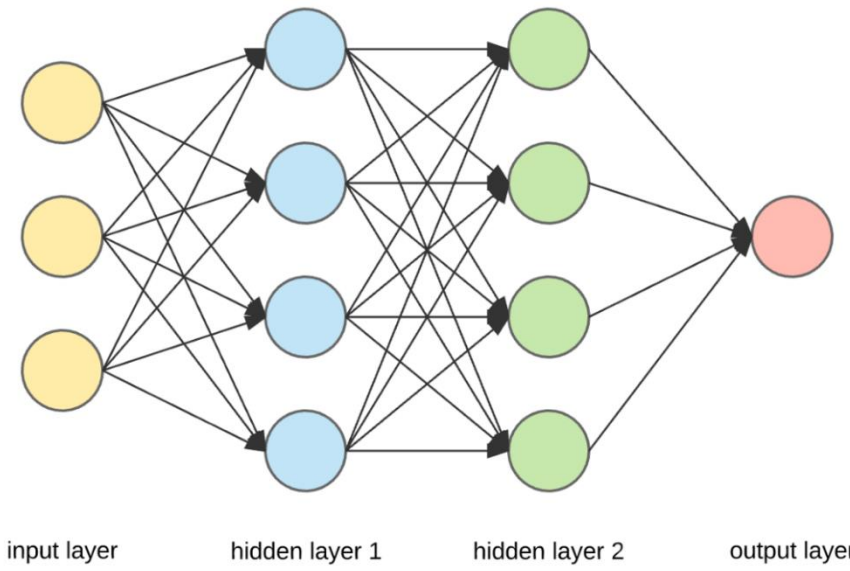


Figure 18 – Representation of an Artificial Neural Network and its layers [49].

5.3.5 Support Vector Machine

A Support Vector Machine (SVM) [43] [50] is a supervised machine learning algorithm used for both regression and classification problems. The SVM consists in calculating a decision boundary that best separates the various output classes from each other. The nearest points of each class to the decision boundary are the support vectors and the objective is to draw a decision boundary equally distant from the support vectors of each class. This distance is named margin, and it is intended to be maximum, so the classifier can be more robust to classify the various classes.

Figure 19 represents a decision boundary for a bi-dimensional data, with its support vectors.

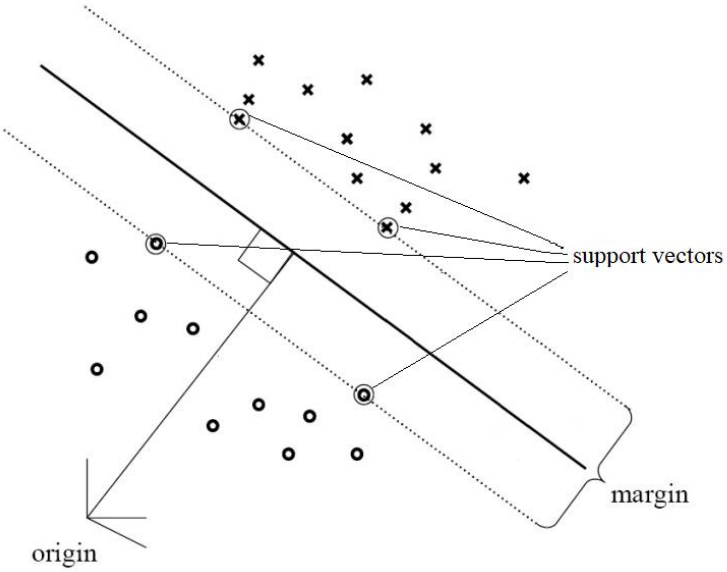


Figure 19 – Illustration of a decision boundary and its support vectors.

For large datasets, it is expected that the data is not linearly separable, having more disperse data and increasing the classifier's error. In this situation, using more features could be a way to try to separate the data, but sometimes features are hard to extract from the dataset. The SVM have a method to get these new features automatically, named kernels. Kernels are functions that transform non-linear spaces into linear ones, so data can be linearly separable as well.

There are several types of kernels but the most known are the Polynomial kernel, Gaussian kernel and Sigmoid kernel described in Table 5.

Table 5 – Type of Kernel and its inner product [50].

Type of Kernel	Inner product	Comments
Polynomial Kernel	$K(\vec{x},\vec{x}_i) = (\vec{x}^T\vec{x}_i + \theta)^p$	Power p and threshold θ are specified by the user
Gaussian Kernel	$K(\vec{x},\vec{x}_i) = e^{-\frac{\|\|\vec{x}-\vec{x}_i\|\|^2}{2\sigma^2}}$	Width σ^2 is specified by the user
Sigmoid Kernel	$K(\vec{x},\vec{x}_i) = \tanh(\eta\vec{x}\vec{x}_i + \theta)$	Mercer's Theorem [50] is satisfied only for some values of η and θ

5.4 Performance evaluation

In a machine learning context, the performance could be validated by how far the predicted value is from reality. In a binary classification problem there are four situations to express the classifier throughput [52]:

- true positives (t_p) – data points classified as positive that are actually positive;
- true negatives (t_n) – data points classified as negative that are actually negative;
- false positives (f_p) – data points classified as positive that are actually negative;
- false negatives (f_n) – data points classified as negative that are actually positive.

5.4.1 Classification metrics

These four attributes are used to get more intuitive and comparable measurements such as accuracy, specificity, recall or precision [52].

Accuracy is the measurement that is used in most cases which represents the rate of data points correctly classified and it is calculated as follows:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{t_p+t_n}}{\boldsymbol{t_p+t_n+f_p+f_n}} \quad \textbf{(15)}$$

This measurement is not so useful in a class imbalance situation where the number of positives and negatives is not equally distributed. Imagining that the number of positives in the dataset is large comparing to the number of negatives, if the algorithm classifies all output as positives the accuracy will be high, however it does not mean that the algorithm is reliable.

Specificity can handle this problem. It gives the rate of the predicted negatives that are actually negative, according to the total number of actual negatives, and is described as follows:

$$\boldsymbol{Specificity} = \frac{t_n}{t_n + f_p} \quad \textbf{(16)}$$

However, it could happen the same imbalance in an inverse way. If the number of negatives in the dataset is large comparing to the number of positives and the classifier predicts all data points as negatives, the accuracy will be high, the specificity will be high, although it is not possible to infer about the performance of the method.

Recall is the inverse of sensitivity. Recall gives the rate of the predicted positives that are actually positive, according to the total number of actual positives, and is defined as follows:

$$\boldsymbol{Recall} = \frac{t_p}{t_p + f_n} \quad \textbf{(17)}$$

For a reliable decision about the classification, there is another metric, named precision. Precision is defined as the rate of the predicted positives that are actually positive, according to the total number of predicted positives, and is calculated as follows:

$$\boldsymbol{Precision} = \frac{t_p}{t_p + f_p} \quad \textbf{(18)}$$

It is possible to combine the recall and precision in one single metric, named F_1 Score. This metric is the harmonic average [53] of both metrics, representing the average rate between these two measurements, and is defined as follows:

$$\boldsymbol{F_1} = \mathbf{2} * \frac{\boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}} \quad \textbf{(19)}$$

These various measurements are needed to evaluate the reliability of a classification method and can help on choosing the right classifier for a specified purpose.

5.4.2 Reliable evaluation

To have a trusty data evaluation, it is not recommended to test the classification method on the same dataset that was used to train the classifier.

The dataset is usually divided in two sets [43] [54]: training set and test set. The training set is the dataset used to train the classifier. It represents the majority of the entire dataset, containing 60% to 80% of the data points, depending on the size of the entire dataset. The test set is a dataset used to test the classifier's performance after being trained by the training dataset. It simulates the new incoming data to be classified, without putting the

classifier at real working conditions. When there are several classification methods involved, is common to split the training set in another dataset [43] [54]: the validation set. This validation set is used to adjust the performance of a classification method before testing its performance, avoiding over-fitting and under-fitting in the data classification.

Figure 20 illustrates a split of a dataset in its subsets of data.

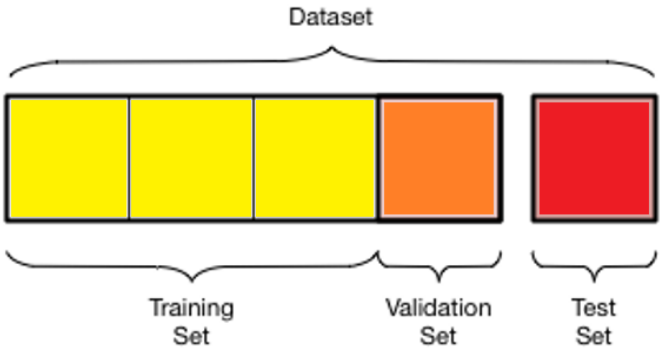


Figure 20 – Example of a dataset splitting according to its training, validation and test sets.

To achieve a consistent performance test, it is needed to test the performance more than once, with different test sets. Although, collecting a large dataset is very difficult and sometimes it is necessary to spend years collecting data. Cross-validation methods [54] are used to enhance the test results by dividing the dataset into different training and test sets combinations. This way, the classifier is trained and tested with different subsets of data, for the same complete dataset.

The ultimate cross-validation method is named exhaustive cross-validation and is meant to train and test the classifier in all the possible ways of combinations in the training and test sets. This can achieve the most reliable tests for a certain classifier, however it demands a considerable data processing to do so. There are simpler cross-validation methods, like k-fold cross-validation [54], that does not require so much data processing and it can achieve reliable results. The k-fold cross-validation consists in having completely different test sets, for the entire dataset. This means that the data points used for testing the classifier are used only once, and the new test set should have completely different data points from the previous test sets. k is the number of folds that will create the results.

Figure 21 represents a diagram with the k-fold cross-validation method.

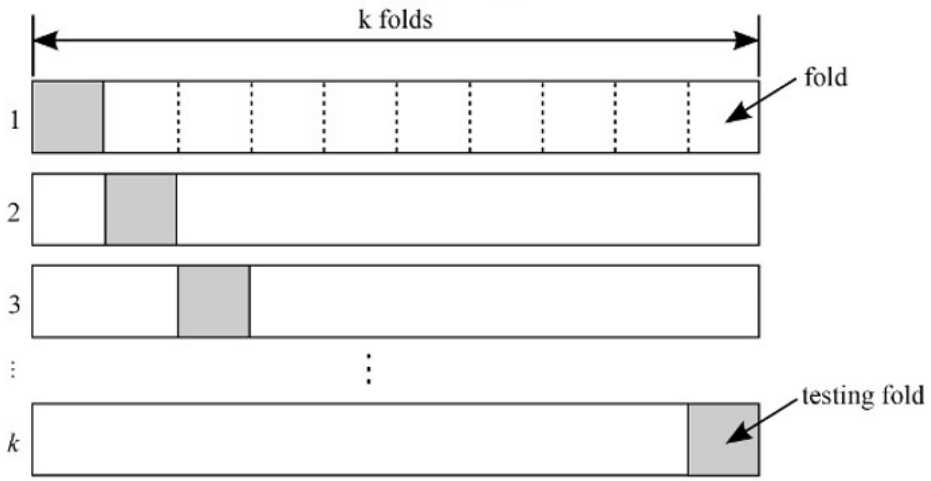


Figure 21 – Representation of a k-fold cross-validation technique.

The final result is the mean of the performance achieved with all the k different tests done plus or minus the standard deviation of the performance for all k different tests, expressed as follows:

$$\boldsymbol{P_{classifier} = \frac{P_1+P_2+\cdots+P_k}{k} \pm \sigma} \quad \textbf{(20)}$$

where $P_{classifier}$ is the result of the classifier's performance, P_i is the performance of the classifier for the i^{th} fold, and σ the standard deviation of the tests done. P is a metric to evaluate the performance; it could be accuracy, specificity, recall, or precision.

For considering the classifier as adequate it is necessary to consider different metrics, as explained in the previous section, and it is needed to have a small standard deviation for each metric. A small standard deviation means that the results are stable, and the values for new incoming data will not diverge too much from the average of the tests. If the standard deviation is high, the new incoming values to classify could diverge a lot, meaning that the classifier cannot be trusted.

5.4.3 Classifier combination

When making a decision, the results can be improved if they not rely only on a single classifier, but in many classifiers. Classifier combination [48] [55] is a technique that consists in combining the results of several classifiers to form a unique opinion.

If the classification results of different classifiers can be fused in an efficient way, then the outcome of such classifier combinations can have superior results. There are two approaches for combining classifiers [55]: sequential and parallel combination. In sequential

combination, the result of one classifier is used as input to another classifier. The sequential order of the classifiers is important and if it is changed, the final result can be different. Parallel combination consists in merging the results of several tested classification algorithms. As opposed to the sequential combination, the classifiers can be tested in any order.

There are many methods to apply parallel combination, but the simplest method is the majority voting. In majority voting, the outcome from the classifier combination are the most voted results from all the classifiers. Imagining three classifiers, C_1, C_2 and C_3, for the first test sample, if C_1 and C_3 classified that sample as “0” and C_2 classified that sample as “1”, the outcome of the classifier combination will be “0”.

Chapter 6

Proposed Solution

This project will be focused in the acquisition device that will transmit the data to the gateway and in the classification algorithm that classifies the data and determinates if the driver is drowsy or not. To accomplish this, a dataset was provided by the Swedish National Road and Transport Research Institute [56] with samples of eighteen different people, tested in awake and drowsy condition states for the same car and track. This dataset is composed by ECG, EEG and EOG biometric signals, and car's movement signals such as velocity, lateral and longitudinal acceleration, Steering Wheel Angle (SWA) and yaw rate. In the experiment, each person was classifying his sleepiness according to the Karolinska Sleepiness Scale (KSS) test while driving, adding a KSS value to each data sample.

Figure 22 represents the block diagram of the proposed solution.

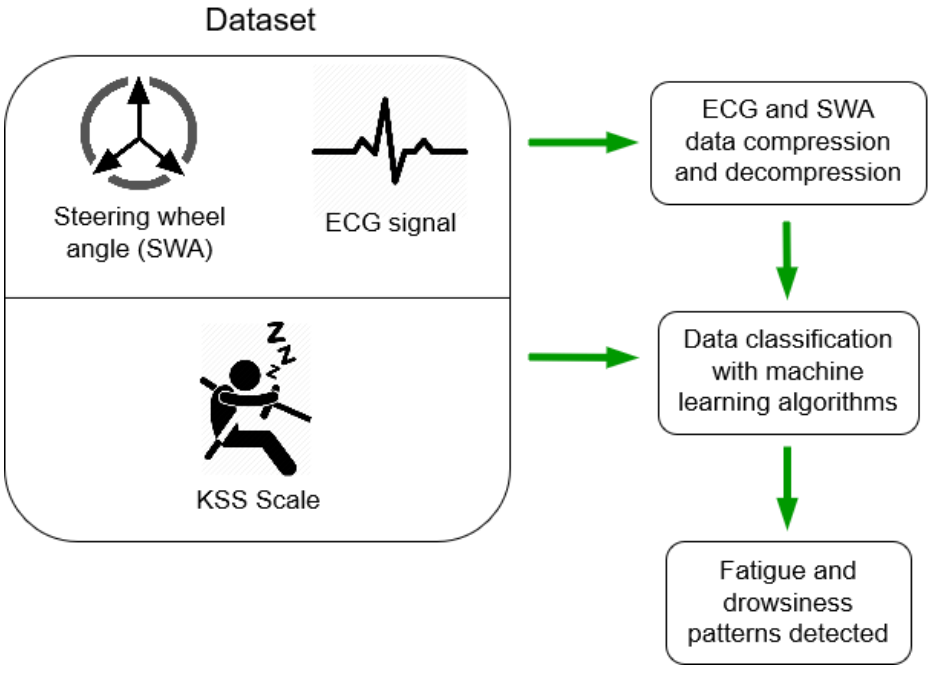


Figure 22 – Block diagram of the proposed system.

This chapter clarifies the steps to implement an acquisition device, covering a BLE solution for transmission, and describes the experimental tests required to implement compression and classification algorithms. Section 6.1, ECG and SWA data acquisition, explains how the ECG and SWA data could be acquired. Section 6.2, Steering wheel motion monitoring, shows how it is possible to monitor steering wheel movements with an

accelerometer. Section 6.3, Compressing the data, gives an overview of the methods to compress both types of data. Section 6.4, Wireless data transmission, describes how the BLE solution could be implemented to transmit the data. Lastly, section 6.5, Features and classification, describes the ECG and SWA features that could be used as inputs for classification algorithms and the dataset classes distribution.

6.1 ECG and SWA data acquisition

The CardioWheel [2] is a system that encompasses all the blocks of an acquisition system solution. This system can collect, in a non-intrusive way, the driver's ECG signal, using dry-electrodes placed in a conductive leather cover, and the SWA signal, using an accelerometer placed in the centre of the steering wheel.

The dry-electrodes can sense the heartbeat, by its electrical impulses, while the person places the hands on the steering wheel. This electrical continuous signal is converted from analogue to digital with an Analogue-to-Digital Converter (ADC) and the resulting samples are read by a microcontroller. The dry-electrodes are placed in a steering wheel leather cover that can fit into any automobile.

Figure 23 shows a picture of a leather cover for steering wheels with the electrodes attached to it.

Figure 23 – Conductive leather cover for the steering wheel with the electrodes placed on it.

The SWA signal is recorded by a three-axis accelerometer, placed in the centre of the steering-wheel behind the airbag. The driver, while moving the steering wheel, causes a variation in each accelerometer axis, and with it, is possible to estimate the rotational angle of the steering wheel.

shows where the CardioWheel mainboard is placed in the steering wheel.

Figure 24 – Location of the mainboard in the steering wheel.

This device has a ST® ARM® Cortex® STM32F446RE [57] microcontroller that acquires ECG and accelerometer data with, respectively, off-board dry-electrodes and an on-board ST® LSM6DSL [58] accelerometer. It also incorporates an on-board Nordic® nRF52832 [59] BLE module for wireless communication. Figure 25 shows pictures of the CardioWheel mainboard.

Figure 25 – Top side (left) and bottom side (right) of the CardioWheel mainboard.

6.2 Steering wheel motion monitoring

To estimate the steering wheel's rotation angle, it is necessary to know how the accelerometer is oriented [59]. This means that, depending on the orientation of the accelerometer, the data could be understood in different ways. The main characteristic that can be recorded with the accelerometer is the rotation angle of the steering wheel (θ), usually called Steering Wheel Angle (SWA).

Figure 26 illustrates a front view of a steering wheel with the axial orientation of the accelerometer.

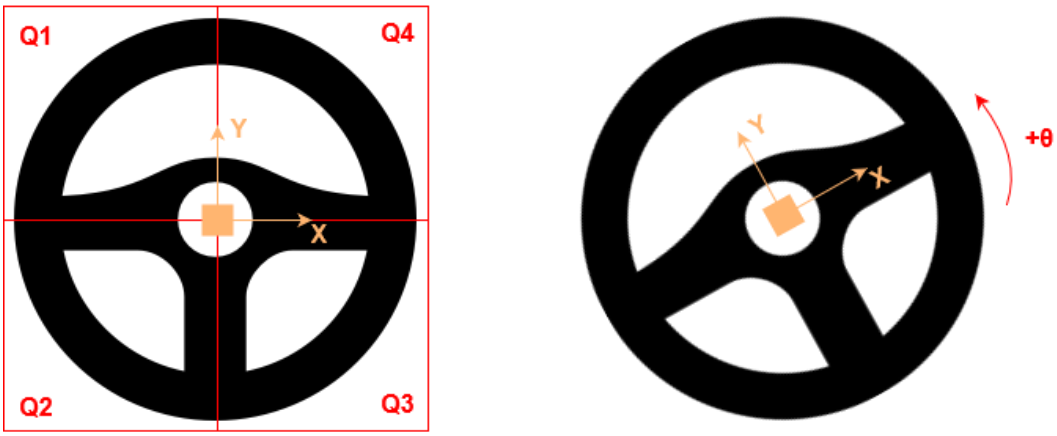


Figure 26 – Front view of a steering wheel with the rotational angle θ and the accelerometer's axial orientation.

Given this accelerometer's axial orientation, the rotation angle of the steering wheel (θ) is measured by the following expression:

$$\tan(\boldsymbol{\theta}) = \frac{A_x}{A_y} \Leftrightarrow \boldsymbol{\theta} = \arctan\left(\frac{A_x}{A_y}\right) \quad \textbf{(21)}$$

where A_x and A_y represent the measured accelerations with the same direction as x and y axes, respectively.

Assuming that there are four quadrants, as represented in Figure 26, it is possible to describe the instantaneous g force range for each quadrant.

Table 6 summarises those instantaneous g force range for each quadrant.

Table 6 – Rotation angle (θ) for each quadrant and its axis g force range [60].

Quadrant: Rotation angle	Accelerations
Q1: $0^{\circ} \leq \theta < 90^{\circ}$	$0\ g \geq A_x > -1\ g$
	$-1\ g \leq A_y < 0\ g$
Q2: $90^{\circ} \leq \theta < 180^{\circ}$	$-1\ g \leq A_x < 0\ g$
	$0\ g \leq A_y < 1\ g$
Q3: $180^{\circ} \leq \theta < 270^{\circ}$	$0\ g \leq A_x < 1\ g$
	$1\ g \geq A_y > 0\ g$
Q4: $270^{\circ} \leq \theta < 360^{\circ}$	$1\ g \geq A_x > 0\ g$
	$0\ g \geq A_y > -1\ g$

Another important parameter to consider is the inclination angle of the steering wheel (γ). Depending on the vehicle and on the driver, the steering wheel could be adjusted to different inclinations to suit the driver's body structure. In each case, the instantaneous g force will be distributed by the three axes in different way according to the inclination angle. Besides this, the car can also be in an inclined plane, therefore this inclination angle is relevant to calibrate the axial system of the accelerometer relative to the car's direction, making more accurate the estimation of the SWA.

Figure 27 illustrates a steering wheel side view with the accelerometer's axial system.

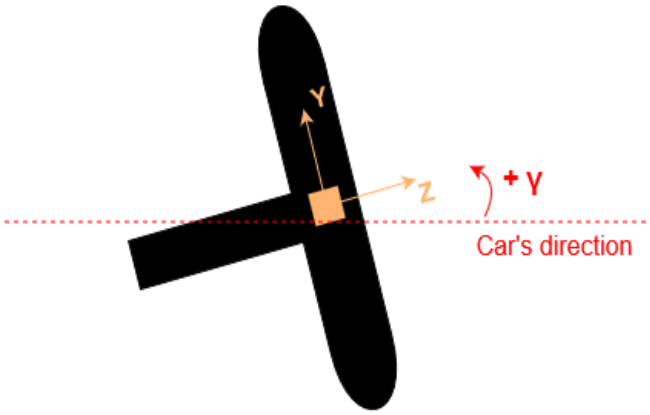


Figure 27 – Side view of a steering wheel with the inclination angle γ and the accelerometer's axial orientation.

Given this accelerometer's axial orientation, the inclination angle (γ) can be written as follows:

$$\tan(\gamma) = \frac{A_z}{A_y} \Leftrightarrow \gamma = \arctan\left(\frac{A_z}{A_y}\right) \quad \textbf{(22)}$$

where A_y and A_z represent the measured accelerations with the same direction as y and z axes, respectively.

With these two angles, rotation (θ) and inclination (γ), it is possible to get accurate measurements for monitoring the behaviour of the steering wheel while driving.

6.3 Compressing the data

With the amount of data collected, it is necessary to have a good and reasonable compression method for data transmission.

The direct time-domain techniques are tested in the literature and have their performance well documented [16] [17] so it's possible to know, beforehand, that they are not the best solution for this project. Compression is an opposite of signal quality and it is privileged a good signal quality at the receiver for purposes of good ECG pattern recognition. If the application was to count heart-rate, it could be possible to have more compression ratio in detriment of signal quality.

The Amplitude Zone Time Epoch Coding (AZTEC) provides high data compression ratios, nearly 10:1 [16], but, due to the discontinuity that occurs in the reconstructed ECG waveform, the fidelity of the reconstructed signal is not acceptable by cardiologists, and for the project purpose, it is needed a signal with almost zero distortion to easily detect the ECG patterns. There is a modified AZTEC method that can improve the signal fidelity by 50% but it still may not be a good solution.

The Turning Point (TP) method has as its main purpose to reduce the sample frequency of a signal by a half and it was tested with a 200 Hz signal. This method produces a fixed compression ratio of 2:1 [16] where the reconstructed signal raises some distortion, since the saved points do not have the same space time intervals. The TP method makes hard to detect the ECG patterns in the reconstructed signal.

Since Coordinate Reduction Time Encoding Scheme (CORTES) applies a combination of AZTEC and TP, the distortion in the reconstructed signal is acceptable for ECG pattern recognition. The compression ratios achieved with CORTES are nearly 4.8:1 [16], and although the hybrid method reduces significantly the distortion, it is not enough for the target of the project.

The transform coding and source coding techniques are documented as good compression methods for ECG signals [18] [19] [21] and it is intended to test these methods and prove their performance.

Accelerometer signals can have waves similar to ones found in ECG signals and it is also intended to test the compression methods in the SWA data. For performance evaluation, the SNR_q will be the reference for comparing to the SNR_t measure to qualitatively evaluate the distortion of the compression methods tested.

6.4 Wireless data transmission

Since CardioWheel has a Bluetooth® Low Energy (BLE) module, it is necessary to create a custom profile to handle this data, transmitting it to a gateway. This device works as a BLE Server while the gateway works as a BLE Client that will request ECG and SWA data from the Server. The Generic Attribute (GATT) [61] protocol states that a BLE Profile is structured in three components: Services, Characteristics, and Descriptors.

A Service is the part of the profile that encapsulates a specific behaviour. These Services are composed by Characteristics. A Characteristic is a value that defines each action for a specific behaviour and it is composed by Descriptors. Descriptors are attributes that define the Characteristic value, and could be, for example, read/write permissions, security roles, among others.

There are a lot of predeveloped Services that could be integrated in each BLE Profile, according to the needs. In this project, was included the Battery Service, for battery analysis purposes, and the Device Information Service, for specific device related information. Nevertheless, it is necessary to create a two custom Services to handle the ECG data and the SWA data in distinctive ways.

Since CardioWheel acts as BLE Server, the two custom Services for the BLE Profile created are composed by one single characteristic that is responsible to load the ECG or SWA data from the memory. This characteristic is composed by two descriptors: Read Descriptor and Notify Descriptor. The Read Descriptor is responsible to provide the data to the BLE Client requests. Whenever the BLE Client wants data, it searches for the to Read Descriptor identifier and asks for data. However, searching for the Read Descriptor without synchronisation could end up in getting repeated data, as the BLE Server may not have new data to deliver. For this situation, the BLE Client is continually searching for the Notify Descriptor identifier. Each time the BLE Server has new data, it updates the Notify Descriptor, and the BLE Client, only searches for the Read Descriptor if it has an update in the Notify Descriptor.

Figure 28 represents the hierarchy of the custom BLE profile created.

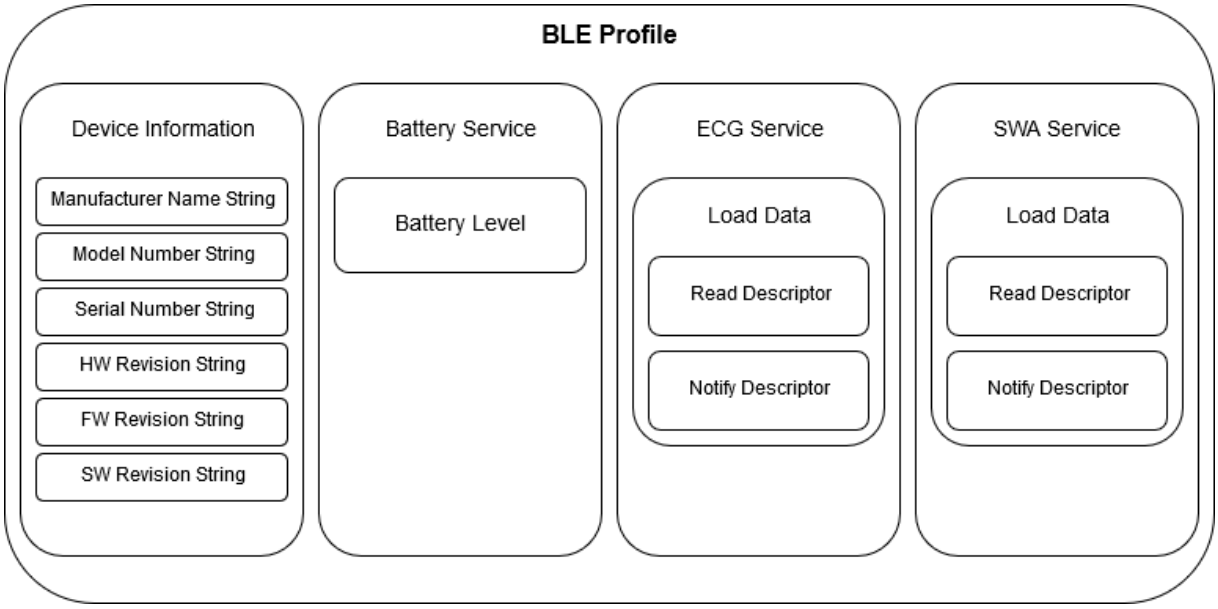


Figure 28 – Hierarchy of the custom Bluetooth Low Energy Profile.

6.5 Dataset for classification

To achieve the goal of fatigue and drowsiness detection, it is necessary a good data classifier to predict the driver's state according to the acquired data. Using the two types of data, ECG and SWA, it is possible to teach a machine to make that prediction.

Different machine learning algorithms need to be tested in order to conclude which can predict the driver's state and which has the best performance in that prediction.

It is indispensable a good dataset, containing various ECG and SWA for different driver's states, to train machine learning algorithms. The dataset provided by the Swedish National Road and Transport Research Institute contains signals from 18 different people, including ECG and SWA, for the same car and track, in both awake and drowsy states, as well as the KSS values for each data sample. The features from those signals will be the input and the KSS values will be the output to train the classifier.

Figure 29 represents a pie chart with the distribution of the KSS values in the given dataset.

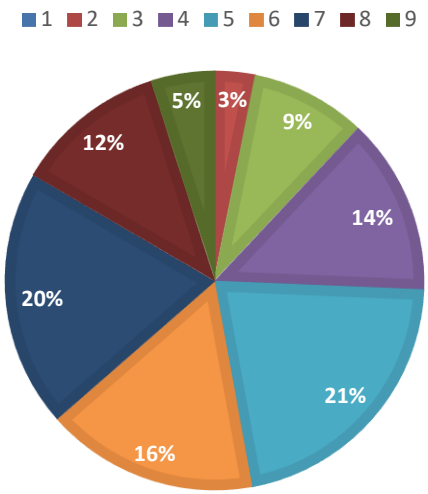


Figure 29 – Pie chart with the distribution of the different KSS classes in the dataset.

To simplify this 9-class output, it is better to transform this into a binary classification problem, where "0" represents the awake state and "1" the drowsy state. According to the KSS scale, from the value 6, the driver is showing some signs of sleepiness, although they are not so significant. The most credible approach for a binary classification is considering the KSS values above 7 as a drowsy state [48], however, the approach in which the dataset becomes more balanced is using KSS values above 6 for classifying a drowsy person. To overcome the class imbalance for the first scenario it will be used the oversampling method to synthesise more drowsy samples.

Figure 30 represents a pie chart with the different dataset balancing using binary

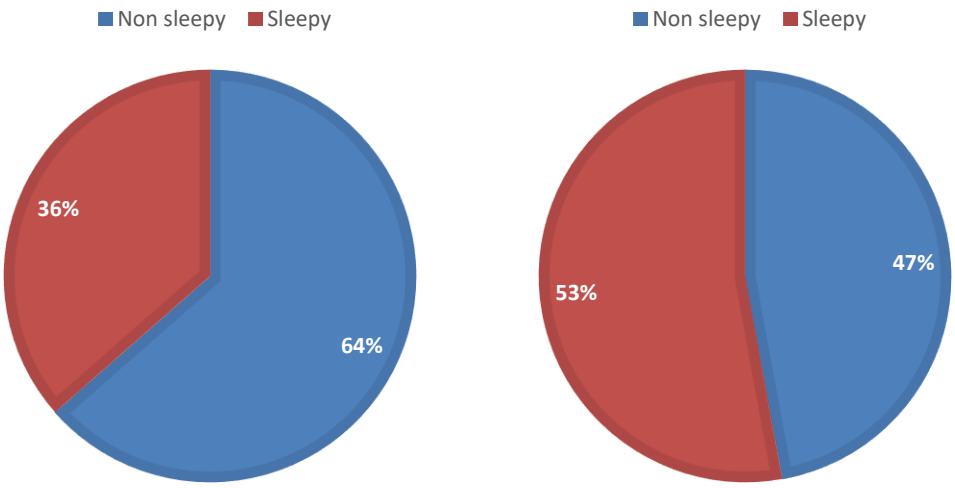


Figure 30 – Awake and drowsy state distribution for a KSS 7 and above (left) and KSS 6 and above (right).

classification for both approaches.

To evaluate the performance of the classifier for a binary problem, it is required to consider the four possible situations, according to the actual and predicted values., Since "0" represents the awake state and "1" the drowsy state, the true positives (t_p), true negatives (t_n), false positives (f_p) and false negatives (f_n) can be represented in a confusion matrix [52] as in Table 7.

Table 7 – Classification confusion matrix.

		Actual	
		Alert	Drowsy
Predicted	Alert	t_n	f_n
	Drowsy	f_p	t_p

The dataset is composed by pairs, each pair representing the same person but in different sleepiness states. To apply cross-validation with a good reliability, each awake-drowsy pair will be used as a test set, reaching 33 results for each tested algorithm. It will be tested each pair, in order to reduce the standard deviation of the results.

Regarding the current problem, the consequence of misclassifying a drowsy driver as awake has potentially more risk than the opposite case. This means that, after cross-validating the performance of the algorithm, it is crucial to minimise the number of false negatives. If a person is driving in a drowsy state, it is fundamental that the vehicle warns the

driver, however, if the vehicle warns the driver when he is awake, the driver could simply ignore the alarm.

Chapter 7

Experimental Evaluation

With the indicators and experimental tests defined, a solution should arise. A Matlab® [62] program was used to test all the scenarios for compression and classification and an Arduino® ATMega 2560 [63] development board was used to test the accelerometer capabilities to estimate the Steering Wheel Angle (SWA).

This chapter reports all the tests and results of the experimental evaluation that was done. Section 7.1, ECG signal pre-processing, describes the first steps for ECG signal processing. Section 7.2, Accelerometer protocol and data acquisition, explains what is necessary to do to put the accelerometer working and the corresponding experimental results. Section 7.3, Compression assessment, enumerates the results obtained with compression algorithms. Section 7.4, Building the classifier, shows the results obtained with the machine learning algorithms.

7.1 ECG signal pre-processing

As mentioned in the previous chapters, the non-intrusive signal acquisition leads to the existence of high level of noise, making the signal less perceptible for humans and machines. Before start to analyse the signal, it is mandatory to have a signal pre-processing to prepare the signal for future analysis.

7.1.1 Filtering

When working with alternating electric current, the 50/60 Hz noise is always present. This noise is more relevant when working with low-frequency signals and demands an initial signal filtering in order to extract only the desired ECG signal.

A FIR filter is recommended since the objective is to preserve the greatest number of characteristics of the biometric signal and, using a filter with a linear phase response, it is possible to estimate the distortion that the filter will introduce and compensate it on the receiver, whereas with the IIR filter it could not be done.

The main frequency components of the ECG signal are below the 40 Hz and in consideration of this, a Hamming-window Low-Pass Filter with a cut-off frequency (f_c) of 40

Hz and an order of 2000 was implemented in order to remove the frequency of the electric current of 50 Hz and, at the same time, preserve the main characteristics of the signal.

Figure 31 illustrates the frequency response of the designed filter and Figure 32 represents an ECG signal, with and without filter.

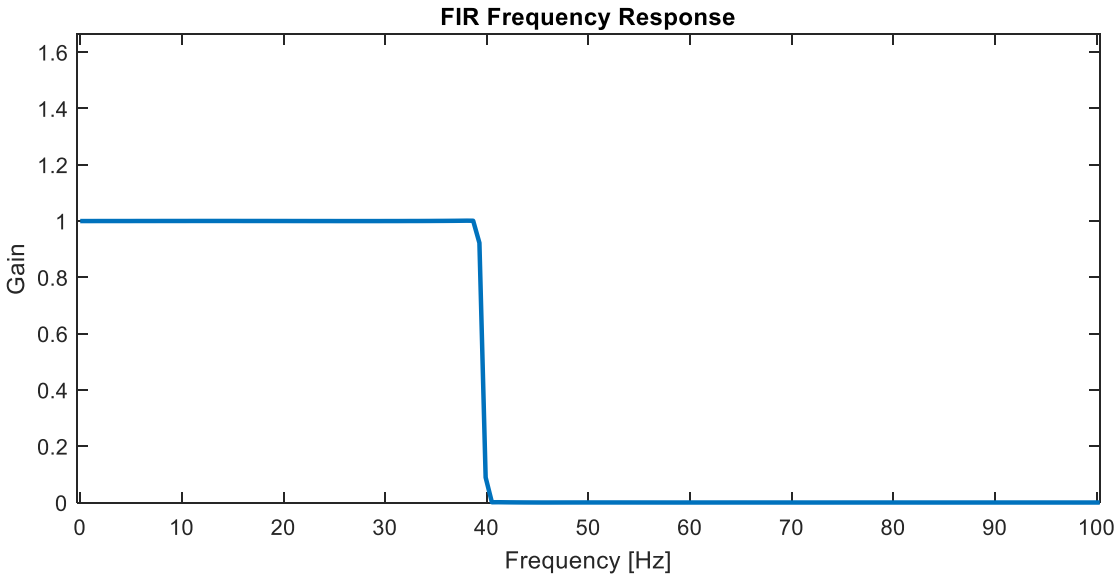


Figure 31 – Frequency response of a Hamming-window Low-Pass FIR filter, with order 2000 ($f_c = 40\ Hz$).

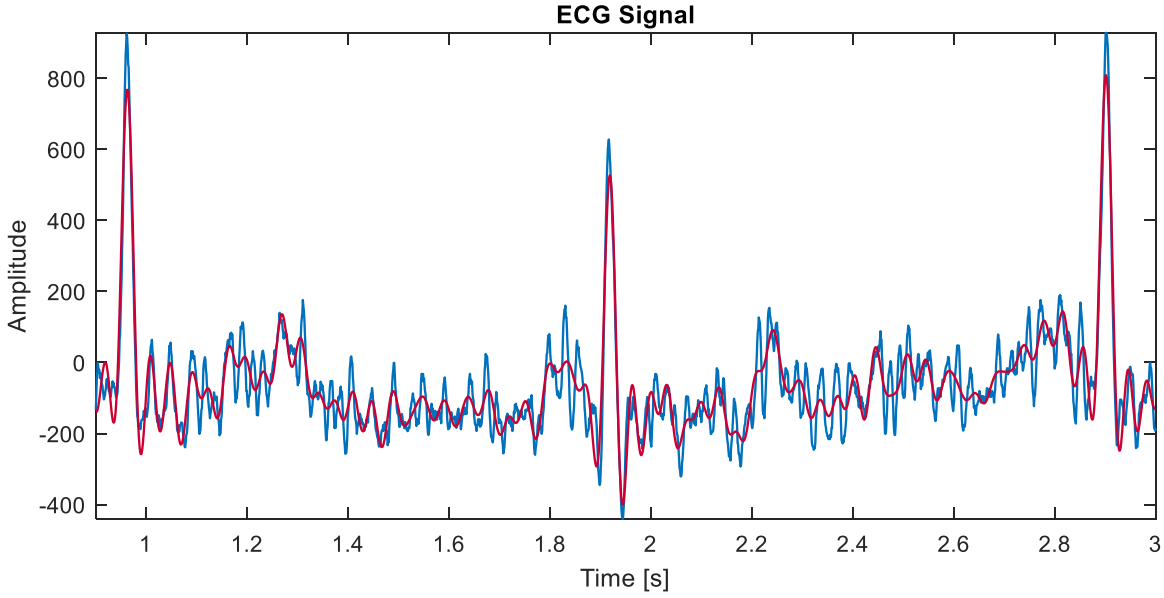


Figure 32 – ECG signal without filtering (blue) and with Hamming-window Low-Pass FIR filtering (red), with order 2000 ($f_c = 40\ Hz$).

7.2 Accelerometer protocol and data acquisition

Every sensor requires a master to retrieve the measures performed and the way to do that task depends on the manufacturer and the purpose of the device. The inertial sensor used is a ST® LSM6DSL [58]. This component includes an accelerometer and a gyroscope and is capable to link it with a magnetic sensor to provide more accurate measures.

7.2.1 Accelerometer operation mode

There are three modes of operation in this component, accelerometer only, gyroscope only, and both operational. In section 6.2 it was concluded that the accelerometer data was enough to fulfil the objective of retrieving the car's SWA. The axes orientation in the steering wheel depends on the position of the mainboard in relation to the steering wheel and is crucial to regulate the equations (21) and (22) before starting the measurements.

The axes of the mentioned accelerometer are oriented as illustrated by Figure 33.

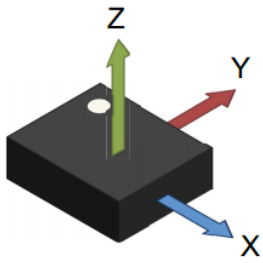


Figure 33 – Accelerometer's three axes orientation [58].

This accelerometer has four scales of operation, 2, 4, 8 or 16 g. For the application it was chosen the 2 g full-scale mode since it is relevant the instant of gravitational acceleration for each axis and not the variation of the acceleration. This smaller full-scale mode brings some advantages as the device is less susceptible to variations, acting like an internal High-Pass Filter (HPF).

About battery consumption concerns, this component has three modes: high performance, normal, and low-power modes. This allows the user to adjust the Output Data Rate (ODR) in order to reduce the battery consumption. It is only necessary that the ODR is higher than the sample frequency at the master. Since the human reaction time is about 200 milliseconds, corresponding to 5 Hz, it is needed a sampling frequency higher than that value and, for this reason, it was chosen an ODR of 208 Hz. This value is higher enough to sample

human movements with a large margin, does not create too much data samples and can reduce the battery consumption if needed.

Figure 34 shows all possible values for ODR and its power modes.

ODR_ XL3	ODR_ XL2	ODR_ XL1	ODR_ XL0	ODR selection [Hz] when XL_HM_MODE = 1	ODR selection [Hz] when XL_HM_MODE = 0
0	0	0	0	Power-down	Power-down
1	0	1	1	1.6 Hz (low power only)	12.5 Hz (high performance)
0	0	0	1	12.5 Hz (low power)	12.5 Hz (high performance)
0	0	1	0	26 Hz (low power)	26 Hz (high performance)
0	0	1	1	52 Hz (low power)	52 Hz (high performance)
0	1	0	0	104 Hz (normal mode)	104 Hz (high performance)
0	1	0	1	208 Hz (normal mode)	208 Hz (high performance)
0	1	1	0	416 Hz (high performance)	416 Hz (high performance)
0	1	1	1	833 Hz (high performance)	833 Hz (high performance)
1	0	0	0	1.66 kHz (high performance)	1.66 kHz (high performance)
1	0	0	1	3.33 kHz (high performance)	3.33 kHz (high performance)
1	0	1	0	6.66 kHz (high performance)	6.66 kHz (high performance)
1	1	x	x	Not allowed	Not allowed

Figure 34 – ODR frequency values according to its power modes [58].

7.2.2 Selecting between I²C and SPI

This accelerometer has two modes of serial communication: Serial-to-Peripheral Interface (SPI) or Inter-Integrated Circuit (I^2C). Depending on the application, one mode could be better than another.

SPI is mostly used for low-power applications using a four-wire communication that enables a full-duplex bus. This allows data to flow simultaneously to and from the master, with rates of up to 10 Mbit/s.

I^2C is an official standard serial communication protocol, with 8-bit data packets, that was designed for communications between chips in the same board. This protocol demands only two wires for communication and can reach up to 3.4 Mbit/s, with new data transmission modes, and 100 kbit/s for the original designed mode. I^2C can have multiple devices in the same bus through device addressing.

Since I^2C is an official data protocol, it is preferable for avoiding compatibility issues, and since it is not needed high speed communications, the I^2C data transmission protocol is the one that best suits this application.

Table 8 describes the SPI and I^2C pinouts.

Table 8 – SPI and I^2C pinout.

Serial type	Pinout	Description
SPI	MISO (Master in Slave Out)	The Slave line for sending data to Master
	MOSI (Master out Slave In)	The Master line for sending data to the Slaves
	SCK (Serial Clock)	Clock pulses for data synchronisation generated by Master
	SS (Slave Select)	The Master line for enabling or disabling Slaves
I^2C	SDA (Serial Data)	Serial bus to send data packets
	SCL (Serial Clock)	Clock pulses for data synchronisation generated by Master

7.2.3 Accelerometer protocol

Every device that works as a slave has a specific way to enable communications with the master, called handshake, and specific ways to read/write data from/to the device, according to the used protocol.

For an I^2C communication, the handshake of this accelerometer is composed by a Start (ST) condition followed by a 7-bit Slave Address (SAD) with the 8^{th} bit set to “0”. The Start condition is done by transiting SDA from HIGH to LOW while keeping SCL HIGH and the 8^{th} bit of the handshake is always set to “0”, meaning write. The 8^{th} bit after the SAD is used to designate the Read (R) or Write (W) intention of the master, “1” or “0”, respectively.

For each received byte, it is needed to send an acknowledge back, so the accelerometer sends a Slave Acknowledge (SAK) to confirm the handshake. After that, the master sends new packets with the 8-bit Sub-address (SUB) of the register which it intends to read or write. A new Slave Acknowledge is sent and, if it is intended to write, the master could send the data to write to the register, if it is intended to read, the master issues a Start Repeated (SR) condition followed by the 7-bit SAD with the 8^{th} bit set to “1”. The accelerometer sends a SAK followed by the data located in that register. If the master wants more data, it replies with a Master Acknowledge (MAK), or, if the master wants to terminate the communication, a No Master Acknowledge (NMAK) followed by a Stop (SP) condition. The Stop condition is done by a LOW to HIGH transition on SDA line while the SCL line is HIGH.

Figure 35 represents the read and write communications with the accelerometer.

Master writing on byte to slave

Master	ST	SAD + W		SUB		DATA		SP
Accelerometer			SAK		SAK		SAK	

Master writing multiple bytes to slave

Master	ST	SAD + W		SUB		DATA		DATA		SP
Accelerometer			SAK		SAK		SAK		SAK	

Master reading one byte from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Accelerometer			SAK		SAK			SAK	DATA		

Master reading multiple bytes from slave

Master	ST	SAD+W		SUB		SR	SAD+R			MAK		MAK		NMAK	SP
Accelerometer			SAK		SAK			SAK	DATA		DATA		DATA		

Figure 35 – I^{2}C accelerometer read/write communications [58].

7.2.4 Initial calibration

The measurements done by an accelerometer are largely dependent from its axes orientation. It is hard to hold the accelerometer in the same position for every vehicle and, to do an accurate measurement, it is necessary that the accelerometer could calibrate itself.

Depending on the car, the accelerometer could be fixed in different ways. This implies that the accelerometer's axes could have different orientations than those exemplified in section 6.2, and the rotation degrees could be measured wrongly. To avoid this, an initial calibration process should be done. The driver, before start driving, should rotate the steering wheel to its limits, right and left. This allows the accelerometer to get the maximum values of the rotational angle θ and, since the rotation of the steering wheel is symmetrical, it can stipulate an offset to adjust these angles, defined as follows:

$$\boldsymbol{\theta_{offset}} = \frac{\mathbf{max_{left}} + \mathbf{max_{right}}}{\mathbf{2}} \quad \textbf{(23)}$$

where $\max_{left}$ and $\max_{right}$ are the rotational limits of the steering wheel for both sides. The calibrated value of θ, can be written as follows:

$$\boldsymbol{\theta_{new}} = \boldsymbol{\theta_{old}} - \boldsymbol{\theta_{offset}} \quad \textbf{(24)}$$

Figure 36 illustrates the initial calibration movement.

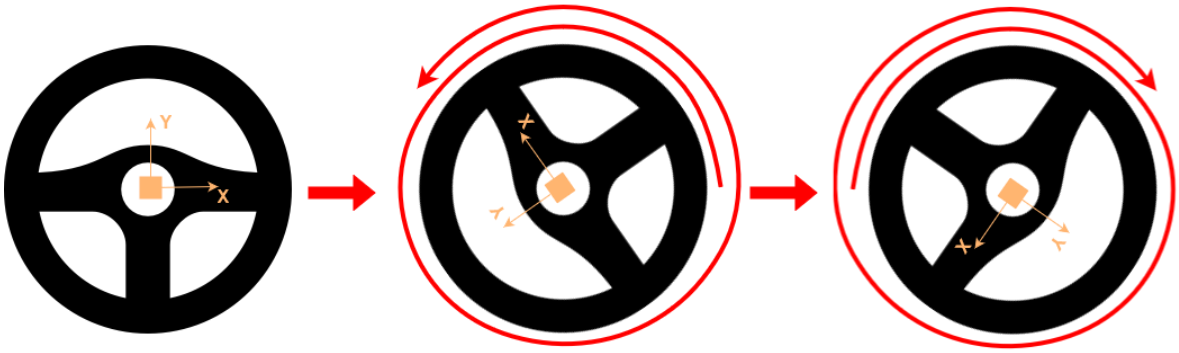


Figure 36 – Representation of the initial calibration movement.

7.2.5 Acceleration data comparison

The final assessment for the accelerometer can be done when the measures for steering wheel motion monitoring are working as expected. To test the accuracy of the device, the accelerometer was fixed, with the same orientation as in Figure 26, on a gaming steering wheel. The steering wheel, in turn, was linked to a PC that is running the driving simulator rFactor™ 2 [64] with the MoTec [65] plugin, that enables data analysis. The acquisition system was composed by an Arduino® ATMega 2560 that was sampling the accelerometer with a rate of 100 Hz and sending the data via Serial Port. The simulator was getting the SWA using the potentiometer inside the gaming steering wheel, with a sampling frequency of 10.24Hz.

The accelerometer and potentiometer sampling were done at the same time with the recording of one lap in a racing track. For the obtained results, it is possible to notice that the accelerometer data and the potentiometer data are very similar, meaning that the accelerometer is getting a correct real-time SWA.

Figure 37 illustrates the results obtained for the Estoril Circuit [66].

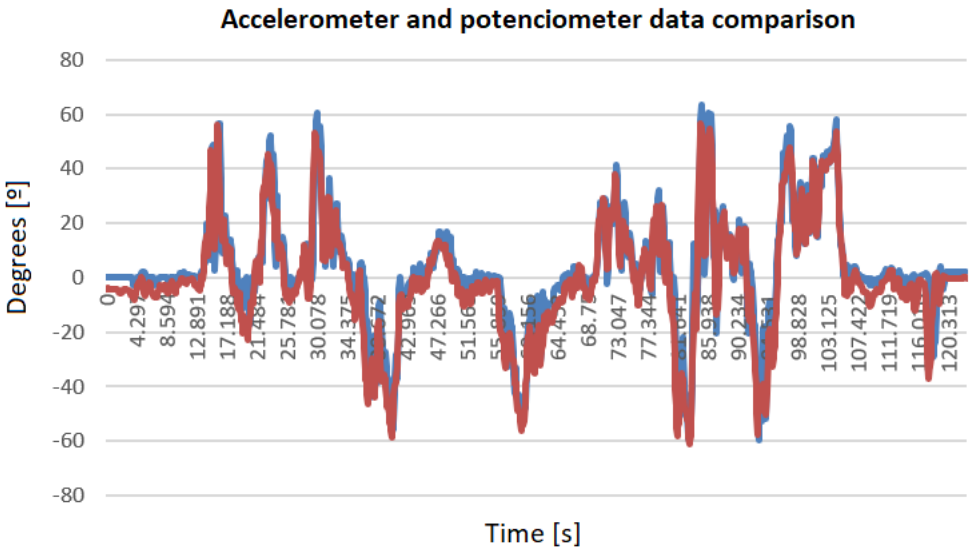


Figure 37 – Accelerometer (red) and potentiometer (blue) angles in degrees at Estoril Circuit.

7.3 Compression assessment

Algorithms in Matlab® were developed to test source and transform compression methods and compare the various metrics with the goal of verifying the method that best suits this application.

As mentioned in section 3.5.3, there are two Signal-to-Noise (SNR) metrics. The quantisation SNR (SNR_q) will serve as the basis for comparing transmission SNR (SNR_t). Considering that the number of bits per symbol after quantisation (R) is 12 for ECG signal and 8 for SWA signal, the supply voltage of the microcontroller (V) is 5 V and the power of the signal (P), the expression of the SNR_q can be written as follows:

$$SNR_{q_{ECG}} = 6.02R + 10\log_{10}\left(\frac{3P}{V^2}\right) = 6.02 * 12 + 10\log_{10}\left(\frac{3P}{5^2}\right)$$

$$SNR_{q_{SWA}} = 6.02R + 10\log_{10}\left(\frac{3P}{V^2}\right) = 6.02 * 8 + 10\log_{10}\left(\frac{3P}{5^2}\right)$$

The ECG and SWA signals used to test the compression methods are the ones represented by Figure 32 and Figure 37, respectively, having a mean power per cycle of 6.2468 W for ECG signal and 6.6596 W for SWA signal. The quantisation SNR are the follows:

$$SNR_{q_{ECG}} = 70.9884\ dB$$

$$SNR_{q_{SWA}} = 47.1863\ dB$$

7.3.1 DPCM + Huffman coding

Huffman is a good technique for compression as it offers some data protection, since both transmitter and receiver should have the same Huffman tree to decode the information. The compression that Huffman offers is due to its variable length code-words. Each code-word has a variable length depending on the probability of occurrence of the corresponding symbol. It is needed a 12 to 8-bit compression before Huffman coding, so the output of it could fit in the BLE frame.

The DPCM compression method, which retrieves the difference between samples of the ECG signal, can obtain values between the 8-bit range of values ([-128, 127]), turning the signal quantised with 8 bits instead of 12. This binary reduction results in a CR of 1.5:1 obtaining an infinite SNR_t. This means that all the DPCM samples fit in the 8-bit range, being a lossless compression method. If any DPCM sample is outside the 8-bit range, the samples are truncated at that limit to be sent via BLE, introducing error in the transmitted signal.

These results are reasonable taking into account that this is a simple computational process but, should be noted that, the DPCM technique depends on the sampling frequency of the signal so, if the sampling frequency is reduced, the samples will be further distanced and the values of the difference between samples may be outside the range obtained with a sampling frequency of 1000 Hz.

However, with Huffman the CR could increase slightly since, Huffman's algorithm outputs variable length codes with less than 8 bits. For transmission purposes, the codes generated by Huffman will be transmitted with frames of 8 bits, making Huffman useful only for data protection.

The SWA data has less amplitude, so it is not needed the initial DPCM coding to change the signal's bit range. With the SWA data, this method also achieved an infinite SNR_t, meaning that it can be applied for both types of signal.

7.3.2 Amplitude scaling + RLE

The amplitude scaling is the simplest method of signal compression tested, however, a division by a coefficient always introduces a small associated error because, when the quantised samples are divided by a coefficient, they get decimal values, which for future wireless transmission, they cannot exist, always having the need to round the values of the new scaled signal.

The RLE coding method does not appear to be the best method for compression since the biometric signal has large variances in the samples. However, when scaling is applied,

the samples will have to be rounded for wireless transmission and, after rounding the samples, the signal will have several successively repeated samples.

In order to achieve a CR higher than DPCM, it was concluded that the number of bits at the output of amplitude scaling must be equal to or greater than 6. Values smaller than 6 bits would cause the signal to have low-power levels and, as soon as samples are rounded for transmission, the signal loses much of its content.

The results of the performed tests for ECG signal are shown in Table 9.

Table 9 – CR, RMSE and SNR_t for the different output lengths for ECG signal.

Amplitude Scaling output length [bits]	CR	RMSE	SNR_t [dB]
8	**1.76:1**	**4.66**	**31.92**
7	3.67:1	9.40	25.83
6	7.95:1	19.09	19.68

Using amplitude scaling and RLE it is possible to achieve high CRs, however, the SNR_t is not acceptable, being almost a quarter of the initial SNR_q.

Since the SWA data has an 8-bit range, there is no need to apply amplitude scaling. However, since SWA has high variability, the RLE cannot be applied because the SWA signal does not have a lot of successive repeated samples.

7.3.3 DCT

Another method of compression is based on the use of DCT. The DCT is similar to DFT with the exception that it uses only real values and projects the input signal on a cosine basis. The DCT can achieve a good performance in signal compression if there are no concerns with the signal amplitude, but only with sample reduction. When applying DCT, the ECG signal is transformed into the one illustrated by Figure 38.

The signal has high amplitudes at the beginning, however, from the sample 4800 and beyond there are no significant amplitude variations, and from sample 11000 until the end, the amplitude is so low that can be discarded. Considering the first 11000 samples, this method can achieve a CR of 5.36:1, and the reconstructed signal has a SNR_t of 54.94 dB.

These results are very reasonable, however, after compression, the signal should be transmitted via BLE using 8-bit frames and the amplitudes that the DCT method generates are incompatible with that transmission, making the DCT not suitable to apply to ECG signal compression.

For the SWA, the behaviour is the same as with the ECG signal, achieving a CR of 11.10:1 and an SNR_t of 32.46 dB, but, in the same way, the signal amplitudes generated by the DCT make it impossible to transmit in 8-bit frames.

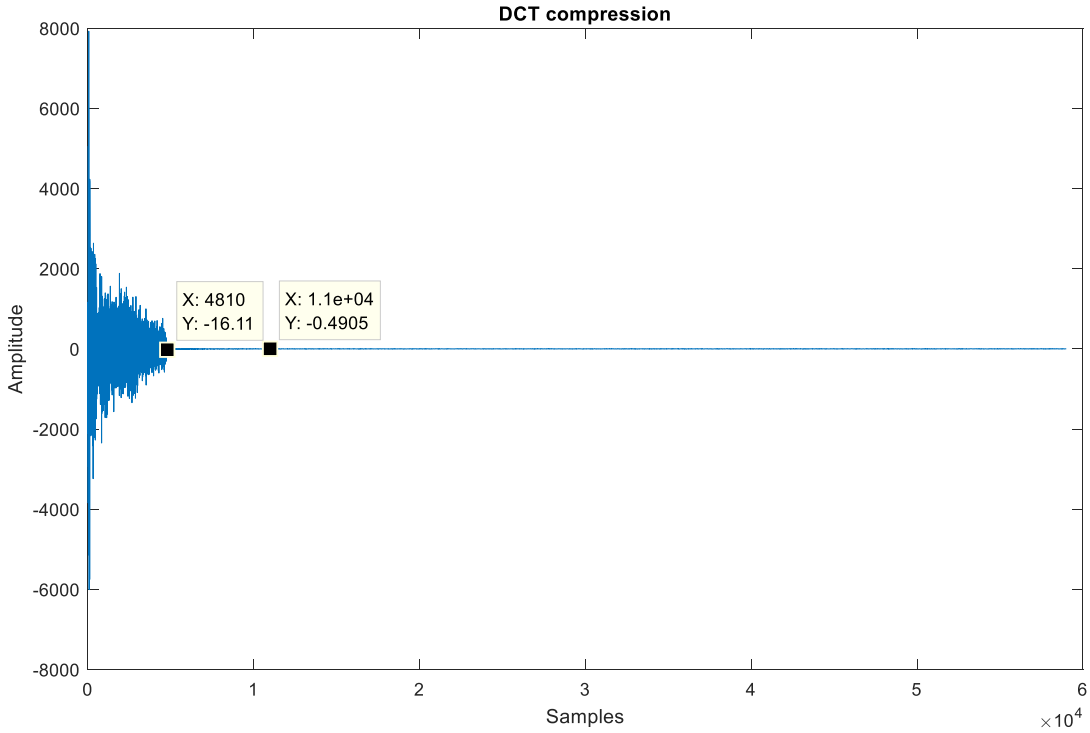


Figure 38 – ECG signal after applying DCT compression.

7.3.4 Amplitude scaling + DWT

The DWT is the decomposition of a signal when passed through an HPF and an LPF creating two sets of coefficients. The compression in the DWT is achieved by only transmitting the approximation coefficients while the detail coefficients are discarded due to its low amplitude. As long as the coefficient level increases, the number of approximation coefficients reduces, but their amplitudes increase. To avoid having high amplitudes in the approximation coefficients, it is possible to reduce the signal amplitude at the beginning of the DWT algorithm with amplitude scaling. For this case, it was applied the amplitude scaling, from 12 to 8 bits before applying the DWT.

The most commonly set of wavelets applied to ECG data is the Daubechies wavelets [7] [21] [26]. These wavelets are known for having the highest number of vanishing moments for a given width. Vanishing moments are the sudden amplitude changes of the signal. Since the

ECG signal has a lot of vanishing moments, the Daubechies are the best to apply DWT, specifically the 'db10' wavelet.

Applying this technique to ECG signal, it is possible to achieve the results summarised by Table 10.

Table 10 – CR, RMSE and SNR_t for the different coefficient levels for ECG signal.

Coefficient Level [bits]	CR	RMSE	SNR_t [dB]
1	3.00:1	3.29	34.96
2	**5.99:1**	**3.56**	**34.26**
3	11.97:1	17.80	20.29

For SWA signal, it was applied the same technique, without the initial amplitude scaling, and using the same Daubechies wavelets. Table 11 summarises the results obtained for SWA compression.

Table 11 – CR, RMSE and SNR_t for the different coefficient levels for SWA signal.

Coefficient Level [bits]	CR	RMSE	SNR_t [dB]
1	2.99:1	0.87	28.46
2	**5.97:1**	**0.70**	**30.31**
3	11.86:1	3.08	17.43

7.3.5 LPC + LZW coding

The Linear Predictive Coding (LPC) is a predictive technique that consists in transmitting the error between the original signal and the predicted one. If the predictor is well dimensioned, the error between the two signals is very small, so a small number of bits is used to represent this error.

According to the LPC method, it is necessary to have $n - 1$ samples of the original signal at the receiver and the a_i coefficients for transmission, so the receiver can predict the next samples of the signal and correct them with the transmitted error. Considering that the signal to be transmitted is an ECG signal, it is needed to have a model with $a + 1$ samples of an ECG signal and the a coefficients to create the predictor.

To create that model, it was used the mean of the first $a+1$ samples of three ECG signals. The initial coefficients of the LPC predictor both at the transmitter and receiver are the mean of the a coefficients calculated for each ECG signal.

The LZW algorithm is applied to the low entropy error from the LPC encoder. To transmit the in 8-bit frames of the BLE, the LZW dictionary should have a maximum of 256 entries. The initial dictionary size depends on the entropy of the error. For the tested ECG, the error after LPC is represented by the histogram of Figure 39.

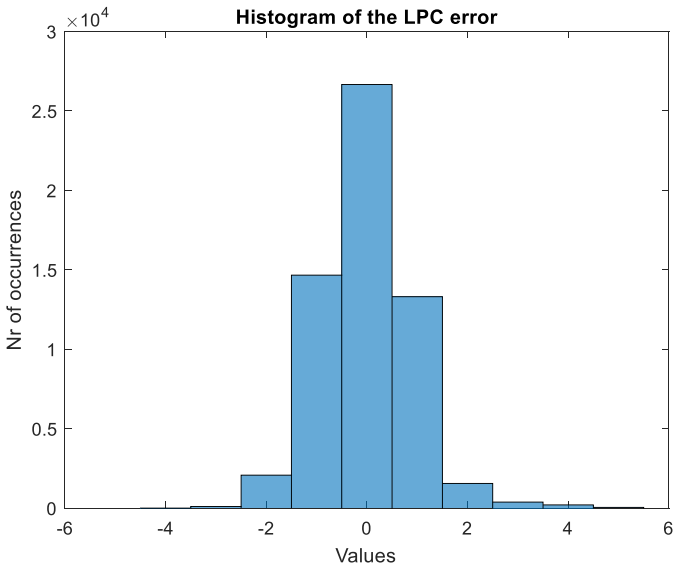


Figure 39 – Histogram of the error after LPC using 10 coefficients.

The entropy of the error is between [-4; 4], so an initial dictionary with 32 entries is adequate to handle the prediction for new ECG signals. However, only 224 entries are not enough to encode the entire signal so the LZW's algorithm starts refilling the dictionary, replacing the oldest entry for the new one. With this technique, more new entries are created to improve the compression ratio of the algorithm.

Table 12 summarises the performance of the hybrid technique, using LPC and LZW, for ECG signal.

Table 12 – CR, RMSE and SNR_t for the different number of coefficients for ECG signal.

Number of coefficients	CR	RMSE	SNR_t [dB]
5	4.29:1	10.64	24.76
10	**4.57:1**	**21.94**	**18.47**
20	4.99:1	39.18	13.43

For the entire signal, the performance using this technique is not the best, having poor values for RMSE and SNR_t, however, this is only at the beginning of the decoding. On a few iterations, the algorithm improves its performance, correcting the error by itself, matching perfectly the original signal, having an infinite SNR_t and no error.

Figure 40 illustrates the initial differences between the original ECG and the predicted one.

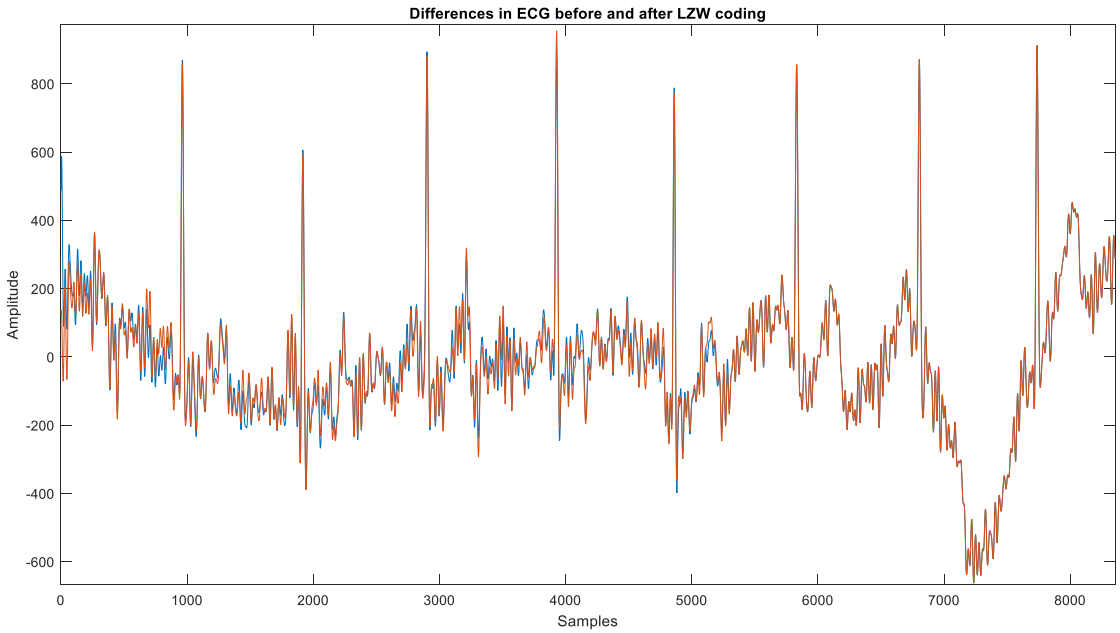


Figure 40 – Differences between the original ECG (blue) and the predicted ECG (red) signals.

For SWA, this hybrid method can be applied, as well, obtaining the results summarised by Table 13.

Table 13 – CR, RMSE and SNR_t for the different number coefficients for SWA signal.

Number of coefficients	CR	RMSE	SNR_t [dB]
5	5.64:1	3.92	15.34
10	**5.78:1**	**1.46**	**23.90**
20	5.77:1	0.64	31.10

From the results obtained for ECG and SWA, it is possible to conclude that having a large number of coefficients does not mean better performance. In the ECG signal, using more coefficients causes the algorithm to be slower in correcting the prediction error. In the SWA

signal, using more coefficients lead to a decrease in the CR. For these reasons, it was concluded that 10 coefficients is the best option to apply for the LPC with LZW technique.

7.3.6 LPC + DEFLATE algorithm

The DEFLATE technique, known to be used in the ZIP file compression, is based on the LZ77 algorithm with Huffman coding. As with other techniques, for transmission purposes, the 12-bit ECG signal should be compressed to an 8-bit signal. For a lossless initial compression, it was used the LPC.

In the same way as with the LZW algorithm, it was used 10 coefficients to apply the LPC technique for the DEFLATE algorithm. With this, it was possible to achieve a CR of 3.45:1 for the same values of RMSE and transmission SNR. For SWA it was achieved a CR of 4.02:1 of the same values of RMSE and transmission SNR.

7.3.7 Discussion of the compression results

With all the compression tests done, it is possible to compare them in order to establish the best solution for ECG and SWA signal compression. According to the CR values obtained, the technique using Amplitude Scaling with DWT proved to be the technique which attains higher compression and lower RMSE. However, this is a lossy technique introduces some distortion in the signal, that cannot be acceptable for precise analysis, like medical analysis.

For lossless compression, the technique using LPC and LZW is the one with the best CR, taking into account that this algorithm needs some time to correct the prediction error and to be effectively a lossless method.

Both techniques are proved to be efficient when compressing the SWA data as well, being compatible for the compression of ECG and SWA signals.

Table 14 and Table 15 summarise all the results obtained with the tested compression topologies for ECG and SWA signals, respectively.

Table 14 – CR, RMSE and SNR_t values achieved with each method for ECG signals.

Method	CR	RMSE	SNR_t [dB]
DPCM + Huffman	1.5:1	0	infinite
Amplitude Scaling + RLE (8 bits)	1.76:1	4.66	31.92
DCT (no amplitude reduction)	5.36:1	0.33	54.94
Amplitude Scaling + DWT (Level 2)	**5.99:1**	**3.56**	**34.26**
LPC + LZW (10 coefficients)	**4.57:1**	**21.94**	**18.47**
LPC + DEFLATE	3.45:1	21.94	18.47

Table 15 – CR, RMSE and SNR_t values achieved with each method for SWA signals.

Method	CR	RMSE	SNR_t [dB]
DPCM + Huffman	1.5:1	0	infinite
Amplitude Scaling + RLE (8 bits)	N/A	N/A	N/A
DCT (no amplitude reduction)	11.10:1	0.55	32.46
Amplitude Scaling + DWT (Level 2)	**5.97:1**	**0.70**	**30.31**
LPC + LZW (10 coefficients)	**5.78:1**	**1.46**	**23.90**
LPC + DEFLATE	4.02:1	1.46	23.90

7.4 Building the classifier

Matlab® was used to build and train a classifier for fatigue and drowsiness detection. A good dataset is important to train the classifier and performance will be dependent on the type of dataset used.

As mentioned in section 6.5, the dataset used to train the classifier is not balanced for the KSS values that define the drowsy or fatigue states. The oversampling technique can synthesise more minority samples making the performance of the classifiers more reliable.

A relationship between the ECG and the SWA signals, and the KSS scale, should be defined with features that will be used as an input to the classifier. In the literature some features were tested as adequate to describe the relationship between ECG or SWA signals with the KSS scales [48] [67]. Table 16 enumerates the features for both signals that were used to train the classifier.

Table 16 – Features used to train the classifier for both types of signal [48] [67].

Signal	Feature	Description
ECG + SWA	SDV	Standard deviation
	ENT	Shannon entropy
	RMS	Root-Mean-Square
ECG	NRP	Number of R peaks per window
	DBR	Mean difference between R peaks
	MAR	Mean amplitude of R peaks
	ADR	Amplitude deviation of R peaks
	VLF	Very-Low Frequency power [0, 0.04] Hz
	LFP	Low Frequency power [0.04, 0.15] Hz
	HFP	High Frequency power [0.15, 0.4] Hz
	LHR	Low-High frequency Ratio
SWA	ZCR	Zero-Crossing Rate
	HTR	Holding time below $\pm$ 3 degrees
	MAS	Mean acceleration applied to the steering wheel
	ASD	Angular Speed Deviation
	EXT	Number of extremes

7.4.1 Linear Regression

Linear Regression is the simplest machine learning algorithm and it is most applied for regression problems.

For a classification problem, this method can be easily influenced by outliers so usually it is preferable using a more complex algorithm. However, the fatigue and drowsiness can be linearly described, so it is interesting to test its performance.

Since Linear Regression uses Gradient Descent, it was used a step α of 0.01 and a maximum of 400 iterations to discover the optimal value of θ.

Table 17 shows the obtained values with this method.

Table 17 – Confusion matrix for Linear Regression.

		Actual		Signal
		Alert	Drowsy	
Predicted	Alert	0.2610 ± 01288	0.2322 ± 0.1664	ECG
	Drowsy	0.2390 ± 0.1288	0.2678 ± 0.1664	
	Alert	$\mathbf{0.3267 \pm 0.0554}$	$\mathbf{0.2665 \pm 0.1260}$	**SWA**
	Drowsy	$\mathbf{0.1733 \pm 0.0554}$	$\mathbf{0.2335 \pm 0.1260}$	
	Alert	0.2948 ± 0.0976	0.2388 ± 0.1320	ECG + SWA
	Drowsy	0.2052 ± 0.0976	0.2612 ± 0.1320	

The values obtained using only the SWA signal are more consistent, having lower standard deviation and better results for the alert states.

7.4.2 Logistic Regression

Logistic Regression can be seen as a solution for non-linear problems. Although it is mostly used for regression problems, it is more reliable for classification than Linear Regression.

This technique uses Gradient Descent to discover the optimal value for θ and it was used a step α of 0.01 with a maximum number of iterations of 400.

Table 18 shows the obtained values with this method.

Table 18 – Confusion matrix for Logistic Regression.

		Actual		**Signal**
		Alert	Drowsy	
Predicted	Alert	0.2051 ± 0.1166	0.1718 ± 0.1500	ECG
	Drowsy	0.2949 ± 0.1166	0.3282 ± 0.1500	
	Alert	0.2710 ± 0.0375	0.2602 ± 0.0543	SWA
	Drowsy	0.2290 ± 0.0375	0.2398 ± 0.0543	
	Alert	$\mathbf{0.3046 \pm 0.0752}$	$\mathbf{0.2523 \pm 0.1029}$	**ECG + SWA**
	Drowsy	$\mathbf{0.1954 \pm 0.0752}$	$\mathbf{0.2477 \pm 0.1029}$	

The values obtained using the ECG and SWA signals are more consistent, having an acceptable standard deviation and better results of accuracy.

7.4.3 Artificial Neural Network

Artificial Neural Network (ANN) is a more complex algorithm for data classification. It is based on the biological neuronal system and each unit in the ANN is similar to a biological neuron.

There are as many input units as features and, from Table 16, it is possible to notice that there will be 11 input units for ECG signal and 8 input units for SWA signal. The number of hidden units can influence the output performance, so it was tested the performance using different number of hidden units, for one hidden layer.

Table 19 shows the performance values using different number of hidden units.

Table 19 – Performance for ANN for different number of Hidden Units (HU).

HU	Accuracy	Specificity	Recall	Precision	F_1 Score	Signal
3	0.4655 ± 0.1046	0.4972 ± 0.2454	0.4338 ± 0.2223	0.4667 ± 0.1217	0.4274 ± 0.1422	ECG
	0.5635 ± 0.0524	0.4173 ± 0.1431	0.7097 ± 0.1708	0.5492 ± 0.0428	0.6113 ± 0.0778	SWA
	0.5396 ± 0.0813	0.5581 ± 0.1948	0.5212 ± 0.2295	0.5410 ± 0.1026	0.5072 ± 0.1497	ECG+SWA
5	0.4638 ± 0.1014	0.5581 ± 0.2594	0.3696 ± 0.2575	0.4494 ± 0.1538	0.3698 ± 0.1829	ECG
	$\mathbf{0.5606 \pm 0.0586}$	$\mathbf{0.4465 \pm 0.1052}$	$\mathbf{0.6747 \pm 0.1606}$	$\mathbf{0.5460 \pm 0.0486}$	$\mathbf{0.5980 \pm 0.0872}$	**SWA**
	0.5426 ± 0.0732	0.5521 ± 0.2009	0.5331 ± 0.2443	0.5405 ± 0.0950	0.5087 ± 0.1621	ECG+SWA
10	0.4898 ± 0.1086	0.5256 ± 0.2570	0.4540 ± 0.2527	0.4936 ± 0.1359	0.4431 ± 0.1651	ECG
	0.5562 ± 0.0646	0.4671 ± 0.1060	0.6452 ± 0.1783	0.5419 ± 0.0595	0.5827 ± 0.1013	SWA
	0.5375 ± 0.0790	0.5615 ± 0.1959	0.5135 ± 0.2436	0.5202 ± 0.1349	0.5104 ± 0.1832	ECG+SWA

With the obtained values it is possible to conclude that the number of hidden units does not have much influence in the output results. However, the results using only SWA signal and using 5 hidden units were the best for the ANN learning algorithm.

7.4.4 Support Vector Machine

Support Vector Machine is another more complex machine learning algorithm. The support vectors are the nearest points of each class to a decision boundary, that is equally distant from each support vector.

The Gaussian Kernel was chosen to test the SVM learning algorithm. Table 20 shows the obtained results for this method.

Table 20 – Confusion matrix for Support Vector Machine.

		Actual		Signal
		Alert	Drowsy	
Predicted	Alert	0.2736 ± 0.1507	0.1293 ± 0.1158	ECG
	Drowsy	0.2264 ± 0.1507	0.3707 ± 0.1158	
	Alert	0.2607 ± 0.0789	0.1312 ± 0.0601	SWA
	Drowsy	0.2392 ± 0.0789	0.3688 ± 0.0601	
	Alert	$\mathbf{0.2835 \pm 0.0595}$	$\mathbf{0.1595 \pm 0.0540}$	**ECG + SWA**
	Drowsy	$\mathbf{0.2165 \pm 0.0595}$	$\mathbf{0.3405 \pm 0.0540}$	

For SVM, it is possible to conclude that using the ECG and SWA signals the learning algorithm had better performance, with the best accuracy and the lowest standard deviation.

7.4.5 Discussion of the classification results

With all the classification tests done, it is possible to compare them in order to define the best solution for fatigue and drowsiness detection.

According to the performance values obtained, the SVM learning algorithm proved to be the technique which obtained a better performance in classification. For the applied method, it is possible to notice that using both signal, the number of false negatives is $15.95 \pm 5.4\%$ meaning it will fail from 10 to 20% of the times in classifying that the person is falling asleep.

For ANN, it was possible to conclude that the hidden unit number does not have significant influence in the output results.

In all the tests done, the ECG signal standard deviation was higher than 10% while with SWA the standard deviation values are around 6%. This means that the algorithms applied are most effective in classifying the SWA data than ECG data, being less susceptible to variations.

Table 21 summarises the results obtained with the different classification algorithms.

Table 21 – Summary of the performance achieved using both signals for each method.

Method	Accuracy	Specificity	Recall	Precision	F_1 Score
LinReg	0.5560 ± 0.0858	0.5895 ± 0.1951	0.5224 ± 0.2640	0.5506 ± 0.1072	0.5057 ± 0.1838
LogReg	0.5523 ± 0.0754	0.6091 ± 0.1593	0.4954 ± 0.2058	0.5553 ± 0.1005	0.5053 ± 0.1427
ANN	0.5426 ± 0.0732	0.5521 ± 0.2009	0.5331 ± 0.2443	0.5405 ± 0.0950	0.5087 ± 0.1621
SVM	$\mathbf{0.6241 \pm 0.0490}$	$\mathbf{0.5670 \pm 0.1189}$	$\mathbf{0.6811 \pm 0.1081}$	$\mathbf{0.6167 \pm 0.0559}$	$\mathbf{0.6415 \pm 0.0565}$

Chapter 8

Conclusion

In order to prevent car accidents and to improve road safety, monitoring systems capable of detecting drowsiness patterns and warning the driver about his physical and psychological condition are extremely needed.

In this dissertation, a complete model for a monitoring system was studied, based on the CardioWheel system, developed by CardioID. This model consists in ECG and steering wheel movement data acquisition, compression, transmission and classification for detection drowsiness and fatigue patterns.

From the CardioWheel model, the ECG data acquisition is done using dry-electrodes in a conductive leather that is covering the steering wheel. While the driver has his hands on the steering wheel, the electrodes can sense the electrical impulses caused by the heartbeat, creating a continuous electrical signal.

The steering wheel motion monitoring is possible to be done by a three-axis accelerometer placed in the centre of the steering wheel. When the driver moves the steering wheel, it changes the acceleration felt in each axis of the accelerometer. Using an Arduino® ATMega 2560, the SWA was extracted by applying trigonometry expressions with the magnitudes of the g force felt in each axis.

To compress all this amount of data, transform and source coding techniques were tested using Matlab® and the method that achieved better compression is the lossy hybrid method using Amplitude Scaling and DWT with a CR of 5.99:1. However, the lossless hybrid method using LPC and LZW obtained a good CR as well – 4.56:1. The choice of which method best suits varies with the type of application and for ECG pattern recognition it is best to preserve the signal than reduce the amount of data.

Since the CardioWheel has a BLE module, the transmission could be ensured by BLE technology with the Profile that was created for the transmission of ECG and SWA. The Profile enables a gateway for getting the ECG and SWA data from the acquisition system.

Also using Matlab, the fatigue and drowsiness detection was accomplished by testing different machine learning algorithms in a two-class problem. The algorithm that reached the best accuracy was the SVM, with an accuracy of 0.6241 ± 0.0490. The percentage of false positives is $15.95 \pm 5.4\ \%$ meaning that from 10% to 20% of the times, the classifier can't predict that the driver is drowsy.

It is crucial to take into account that the KSS scale, used as output in the supervised learning task, is a subjective scale and subjective measures are based in self-rating scores

given by the drivers and, although they are helpful in understanding the driver's condition, they are highly depended on the personal evaluation and interpretation.

With these results it is possible to conclude that it is possible to implement the system defined in this dissertation. However, the results are not persistent since they not account for all the possible conditions.

8.1 Future work

This dissertation was more focused in the compression and the classification methods, however, to fulfil all the proposed approach there are some topics that should be investigated more deeply.

In the acquisition device, more efficient compression algorithms can be implemented, that can enable higher compression ratios with a tolerant loss in the signal quality.

The BLE Profile that was created for the transmission of ECG and SWA was not tested but its conception was studied, in order to guarantee a good structure for a future implementation. After testing the BLE Profile, more characteristics may be useful and can be added.

The results for data classification are very dependent from the dataset used to train the classifier. The tested conditions are the same for the entire dataset and this influences the data. For more reliable results this should be tested with a more complete dataset that contains different circuits, different vehicles or different weather conditions.

Also, in the dataset, the ECG signal was extracted in an intrusive way, so when testing machine learning algorithms with signals acquired in a non-intrusive way, it is expected that the quality of the signal decreases and the extracted features could not be relevant enough to describe the signal differences. Besides this, more features could be defined in order to improve the results for intrusive or non-intrusive ECG data.

Since the SVM method proved to be the best of the tested algorithms, it is necessary to test different kernels in order to find if it is possible to improve the algorithm results. Besides this, different classifier combinations could be done, sequential or parallel combination, in order to test, as well, if there is any improvement in the obtained results.

References

[1] Mobileye®, “ADAS,” 2018. [Online]. Available: https://www.mobileye.com/our-technology/adas/. [Accessed 29 September 2018].

[2] A. Lourenço, A. P. Alves, C. Carreiras, R. P. Duarte and A. Fred, “CardioWheel: ECG Biometrics on the Steering Wheel,” Springer International Publishing Switzerland, 2015.

[3] D. Haupt, P. Honzik, P. Raso and O. Hyncica, “Steering wheel motion analysis for detection of the driver's drowsiness,” *Mathematical Models and Methods in Modern Science,* pp. 256-261.

[4] E. Eitel, “Basics of Rotary Encoders: Overview and New Technologies,” 7 May 2014. [Online]. Available: https://www.machinedesign.com/sensors/basics-rotary-encoders-overview-and-new-technologies-0. [Accessed 19 September 2018].

[5] Y. VinayKumar, P. Srinivasulu, Y. Greeshma, A. M. Vamsi Krishna and S. Md., “Design and Simulation of MEMS Capacitive Accelerometer,” *IJECT,* vol. 2, no. SP-1, pp. 233-236, 2011.

[6] U. S. Tiwary and B. Gohel, “Automated Risk Identification of Myocardial Infarction Using Relative Frequency Band Coefficient (RFBC) Features from ECG,” *The Open Biomedical Engineering Journal,* vol. 4, pp. 217-222, 2010.

[7] C. Sarltha, V. Sukanya and Y. Naraslmha Murthy, “ECG Signal Analysis Using Wavelet Transforms,” Heron Press Ltd., 2008.

[8] C. Carreiras, A. Lourenço, H. P. da Silva and A. Fred, “Comparative Study of Medical-grade and Off-ter-person ECG Systems”.

[9] T. Sellman, “Tired all the time: Is it sleepiness or is it fatigue?,” 24 January 2017. [Online]. Available: https://multiplesclerosis.net/living-with-ms/tired-all-the-time-is-it-sleepiness-or-is-it-fatigue/. [Accessed 19 September 2018].

[10] A. Shahid, “Karolinska Slepiness Scale (KSS),” *THAT and One Hundred Other Sleep Scales,* vol. 47, pp. 209-210, 2012.

[11] A. Joshi, S. Kale, S. Chandel and D. K. Pal, “Likert Scale: Explored and Explained,” *British Journal of Applied Science & Technology,* vol. 7, no. 4, pp. 396-403, 2015.

[12] IIT Kanpur, “Digital Filters,” [Online]. Available: https://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/Digi_Sign_Pro/pdf/chap-9.pdf. [Accessed 19 September 2018].

[13] B. Yadav, S. Rana, N. Yadav and M. P. Bansal, “Comparison of IIR Filter and FIR Filter and their Types,” *International Journal of Electrical and Electronics Engineers,* vol. 8, no. 2, pp. 356-361, 2016.

[14] Mini DSP, “FIR vs IIR filtering,” [Online]. Available: https://www.minidsp.com/applications/dsp-basics/fir-vs-iir-filtering. [Accessed 19 September 2018].

[15] H. T. Nguyen, “50Hz Interference and Noise in ECG Recordings - A Review,” *Australasian Physical & Engineering Sciences in Medicine,* vol. 17, no. 3, pp. 108-115, 1994.

[16] Priyanka and I. Saini , “Analysis ECG Data Compression Techniques- A Survey Approach,” International Journal of Emerging Technology and Advanced Engineering, 2013.

[17] S. M. S. Jalaleddine, C. G. Hutchens, R. D. Strattan and W. A. Coberly, “ECG Data Compression Techniques - A Unified Approach,” IEEE, 1990.

[18] K. Sayood, Introduction to Data Compression, Morgan Kaufmaan, 2006.

[19] S. W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, 1997.

[20] L. B. Portolés, “Lossless Compression of ECG Signals - Performance Analysis in a Wireless Network,” 2009.

[21] M. M. Abo-Zahhad, “ECG Signal Compression Algorithms In Wireless Data Transmission Context,” Assiut University Faculty of Engineering, 2014.

[22] M. K. Mathur, A. R. Garg and M. Upadhayay, “Application of LZW Technique for ECG Data Compression,” *International Journal of Advances in Computer Networks and its Security,* pp. 374-377.

[23] S. M. Choudhary, A. S. Patel and S. J. Parmar, “Study of LZ77 and LZ78 Data Compression Techniques,” *International Journal of Engineering Science and Innovate Technology,* vol. 4, no. 3, pp. 45-49, 2015.

[24] S. Oswal, A. Singh and K. Kumari, “Deflate Compression Algorithm,” *International Journal of Engineering Research and General Science,* vol. 4, no. 1, pp. 430-436, 2016.

[25] N. S. Jayant, “Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers,” *Proceedings of the IEEE,* pp. 611-632, 1974.

[26] S. Ardhapurkar, R. Manthalkar and S. Gajre, “Eletrocardiogram Compression by Linear Prediction and Wavelet Sub-Band Coding Techniques,” 2011.

[27] J. S. Erkelens, “Autoregressive Modelling for Speech Coding: Estimation, Interpolation and Quantisation”.

[28] M. K. Chhipa, “Performance Analysis of Various Transform Based Methods for ECG Data,” International Journal of Scientific and Research Publications, 2013.

[29] C. S. Burrus, “Fast Fourier Transforms,” Houston, Texas, 2008.

[30] D. S. G. Pollock, “Two-Channel Filter Banks and Dyadic Decompositions”.

[31] R. Khanam and S. N. Ahmad, “Selection of Wavelets for Evaluating SNR, PRD and CR of ECG Signal,” *International Journal of Engineering Science and Innovate Technology,* vol. 2, no. 1, pp. 112-119, 2013.

[32] C. Meneses Ribeiro, “Sistemas de Comunicação Digital,” Lisboa, 2015.

[33] Bluetooth SIG, Specification of the Bluetooth System, 2014.

[34] K. Townsend, “Introduction to Bluetooth Low Energy,” Adafruit Industries, 2018.

[35] T. Aasebø, “Lecture Slides of Wireless Technologies - Bluetooth, ZigBee and ANT”.

[36] A. Dementyev, S. Hodges, S. Taylor and J. Smith, “Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario”.

[37] A. Tomar, “Introduction to ZigBee Technology,” element15, 2011.

[38] E. Casilari, J. M. Cano-García and G. Campos-Garrido, “Modeling of Current Consumption in 802.115.4/ZigBee Sensor Motes,” *Sensors,* vol. 10, pp. 5443-5468, 2010.

[39] Dinastream Innovations, “ANT Message Protocol and Usage,” 2014.

[40] E. A. Ince, “Lecture notes of Time Divison Multiplexing (TDM),” EEE 360: Communication Systems I, 2003.

[41] S. Khalid, T. Khalil and S. Nasreen, “A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning,” in *Science and Information Conference*, London, 2014.

[42] P. Cunningham, “Dimension Reduction,” 2007.

[43] A. Ng, “Lecture Slides from Machine Learning Course,” Coursera, [Online]. Available: https://en.coursera.org/learn/machine-learning#. [Accessed 20 Setember 2019].

[44] W. Koehrsen, “Towards Data Science,” Medium, 28 January 2018. [Online]. Available: https://towardsdatascience.com/overfitting-vs-underfitting-a-conceptual-explanation-d94ee20ca7f9. [Accessed 20 September 2018].

[45] X. Guo, Y. Yin, C. Dong, G. Yang and G. Zhou, “On the Class Imbalance Problem,” in *Fourth International Conference on Natural Computation*, 2008.

[46] J. Brownlee, “Supervised and Unsupervised Machine Learning Algorithms,” Machine Learning Mastery, 16 March 2016. [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/. [Accessed 20 September 2018].

[47] F. Lindsten, A. Svensson, N. Wahlström and T. B. Schön, “Statistical Machine Learning,” 2018.

[48] C. Silveira, “Driver's Fatigue State Monitoring using Physiological Signals,” *Master's Thesis,* 2017.

[49] A. Dertat, “Towards Data Science,” Medium, 8 August 2017. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6. [Accessed 20 September 2018].

[50] M. Hoffman, “Support Vector Machines - Kernels and the Kernel Trick,” 2006.

[51] J. Thickstun, “Mercer's Theorem”.

[52] D. M. W. Powers, “Evaluation: From Precision, Recall and F-Measure to Roc, Informedness, Markednes & Correlation,” *Jorunal of Machine Learning Technologies,* vol. 2, no. 1, pp. 37-63, 2011.

[53] L. Molnár, “Arithmetic, Geometric and Harmonic Means in Operator Algebras and Transformations among them,” *Mathematics Subject Classification,* 2010.

[54] Z. Reitermanová, “Data Splitting,” *WDS' 10 Proceedings of Contributed Papers,* vol. 1, pp. 31-36, 2010.

[55] S. Tulyakov, S. Jaeger, V. Govindaraju and D. Doermann, “Review of Classifier Combination Methods”.

[56] VTI, “Swedish National Road and Transport Research Institute,” VTI, [Online]. Available: https://www.vti.se/en/. [Accessed 21 September 2018].

[57] ST®, “STM32F446xC/E - Datasheet - production data,” 2016.

[58] ST®, “LSM6DSL - Datasheet - production data,” 2017.

[59] Nordic®, “nRF52832 Product Specification v1.4,” 2017.

[60] S. Lawoyin, D.-Y. Fei and O. Bai, “Accelerometer-based steering-wheel movement monitoring for drowsy-driving detection,” *Proc IMechE Part D: J Automobile Engineering,* vol. 229, no. 2, pp. 163-173, 2015.

[61] Bluetooth SIG, “GATT Specifications,” [Online]. Available: https://www.bluetooth.com/specifications/gatt. [Accessed 20 September 2018].

[62] MathWorks®, “Matlab®,” [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed 22 Setembro 2018].

[63] Arduino®, “Getting Started with Arduino and Genuino MEGA2560,” [Online]. Available: https://www.arduino.cc/en/Guide/ArduinoMega2560. [Accessed 22 September 2018].

[64] rFactor™, “rFactor 2,” [Online]. Available: https://www.rfactor.net/. [Accessed 22 September 2018].

[65] MoTec, [Online]. Available: https://www.motec.com.au/home. [Accessed 22 September 2018].

[66] Circuito do Estoril, [Online]. Available: http://www.circuito-estoril.pt/pt/. [Accessed 22 September 2018].

[67] F. Friedrichs and B. Yang, “Drowsiness Monitoring by Steering Wheel and Lane Data Based Features Under Real Driving Conditions,” *18th European Signal Processing Conference,* pp. 23-27, 2010.