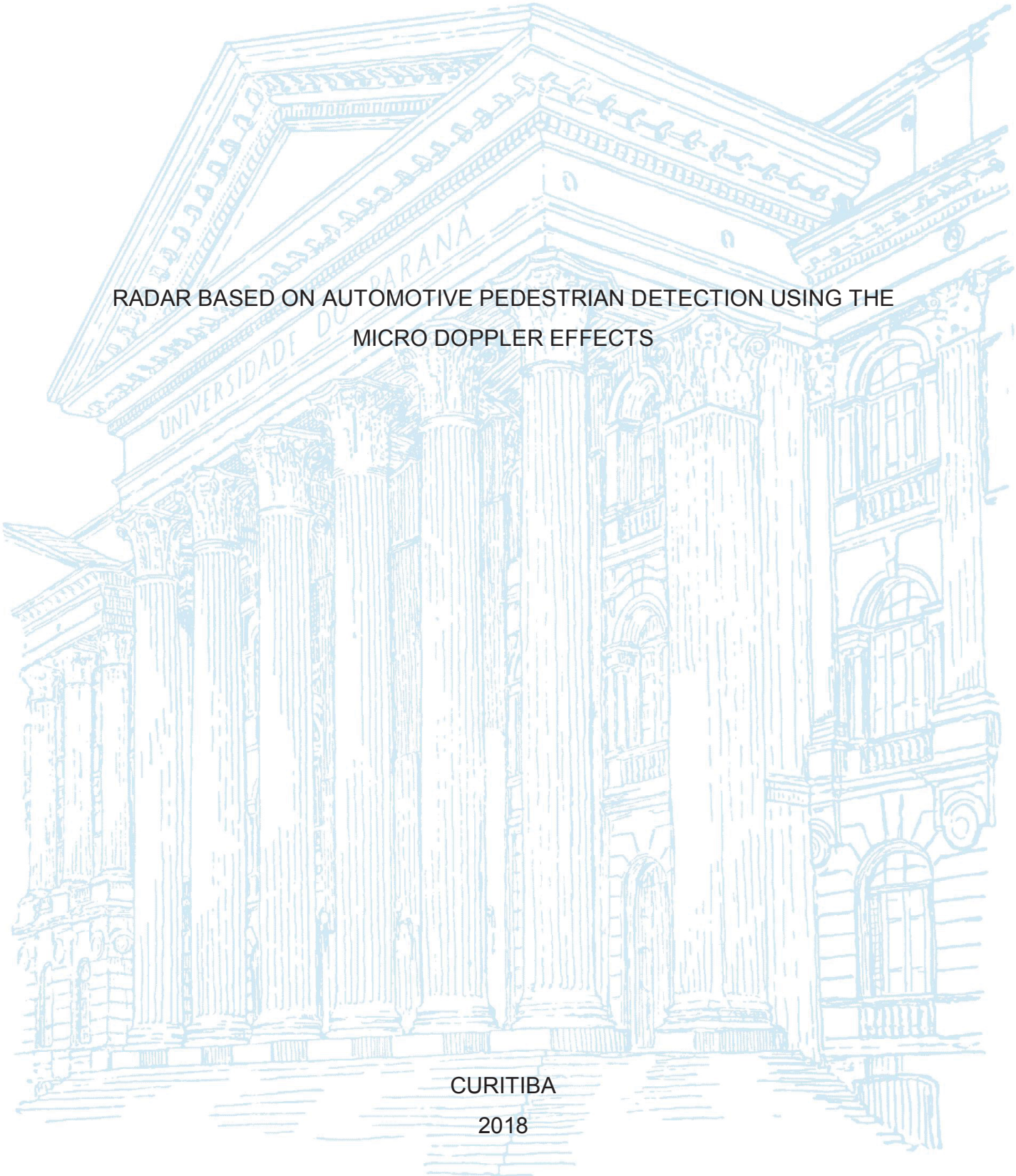
UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO VICTOR BRUNETI SEVERINO



RADAR BASED ON AUTOMOTIVE PEDESTRIAN DETECTION USING THE MICRO DOPPLER EFFECTS

CURITIBA

2018

JOÃO VICTOR BRUNETI SEVERINO

RADAR BASED ON AUTOMOTIVE PEDESTRIAN DETECTION USING THE MICRO DOPPLER EFFECTS

Dissertação apresentada ao programa de pôs graduação em engenharia elétrica do setor tecnológico da Universidade Federal do Paraná (UFPR) como requisito parcial para a obtenção do diploma de mestre em engenharia elétrica.

Orientador: Prof. Dr. Alessandro Zimmer.

CURITIBA

2018

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

S498r

Severino, João Victor Bruneti
Radar based on automotive pedestrian detection using the micro doppler effects / João Victor Bruneti Severino. – Curitiba : UFPR : Faculdade de Engenharia Elétrica e Ciências da Computação da Technische Hochschule Ingolstadt (THI), Alemanha, 2018.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, 2018.

Orientador: Alessandro Zimmer. -

1. Máquinas de vetores de suporte. 2. Radar. 3. Detecção de pedestres. I. Universidade Federal do Paraná. II. Faculdade de Engenharia Elétrica e Ciências da Computação da Technische Hochschule Ingolstadt (THI). III. Zimmer, Alessandro. IV. Título.

CDD: 623.76

Bibliotecária: Vanusa Maciel - CRB - 9/1928

MINISTÉRIO DA EDUCAÇÃO
SETOR TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA ELÉTRICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **JOÃO VICTOR BRUNETI SEVERINO** intitulada: **Radar based on automotive pedestrian detection using the micro Doppler effects**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 29 de Agosto de 2018.

ALESSANDRO ZIMMER
Presidente da Banca Examinadora (UFPR)

ALCEU DE SOUZA BRITTO JR
Avaliador Externo (PUCPR)

EDUARDO PARENTE RIBEIRO
Avaliador Interno (UFPR)

ACKNOLEDGMENT

- To my advisor, Prof. Alessandro Zimmer, for the opportunity to study abroad and for helping with the scholarship.
- To my advisors, Prof. Roberto Zanetti Freire and Prof. Leandro dos Santos Coelho, for all the technical support and ideas.
- To Center of Automotive Reasearch on Integrated Safety Systems and Measurement Area (CARISSMA) laboratory and Prof. Thomas Brandmeier, for accepting me and for giving all the support that I needed.
- To Dr. Dagmar Steinhauser, for the technical support and orientation;
- To Federal University of Paraná (UFPR) and Technische Hochschule Ingolstadt (THI), for creating the double degree possibility for the students;
- To Araucaria Foundation (FA), for the financial support in Germany.
- To my friends Matheus Hernandes and Luan Labigalini, for helping with the data base creation.
- To my family and friends, for the emotional support in every hours.

RESUMO

O desenvolvimento do carro autônomo é hoje em dia uma prática comum entre as maiores indústrias automotivas, e também em indústrias tecnológicas, como o Google e a Apple. Ao adicionar mais sensores, o veículo é capaz de se movimentar sozinho, identificar a trajetória correta, a distância para outros carros, e também a presença de objetos e seres vivos. Entretanto, existem muitos aspectos bloqueando o lançamento do carro autônomo. Como exemplo aspectos técnicos, como o caso do reconhecimento de pedestres. Embora, esse tópico seja abundantemente estudado para o uso de câmeras digitais, as mesmas não possuem confiabilidade nas medições de velocidade e distância, e ainda apresentam péssimos resultados quando há variação ou a falta de luz no ambiente. Baseado no que foi mencionado anteriormente, o foco dessa dissertação é de desenvolver e discutir a eficiência de um sistema de rápida identificação de pedestres, utilizando um novo radar de 79GHz de frequência. O principal objetivo é reconhecer o pedestre o mais rápido possível utilizando os efeitos micro Doppler do movimento humano em situações muito próximas de um acidente, junto com o método de classificação *support vector machine* (SVM). Objetivando essa meta algumas técnicas são usadas ao longo do trabalho. Primeiramente, a resolução de velocidade é melhorada com técnicas de otimização multiobjetivos, como algoritmos genéticos e *random search* para extrair o micro efeito Doppler. Então as informações de velocidade e distância são medidas pelo radar. Em sequência, um método de extração de características chamado de *video temporal gradiente* é aplicado. O método de machine learning SVM classifica os objetos em pedestre e não pedestres, com quadro diferentes métodos de treinamento. Por fim, é possível ver as vantagens do método de otimização que consegue atingir uma resolução de velocidade de 0,12 m/s. A comparação dos modelos de SVM mostra que o quarto modelo, utilizando *kernel* polinomial, apresenta os melhores resultados com uma acurácia de 99,5%. Entretanto, o tempo de processamento não é bom o suficiente, levando 72 ms para a classificação de um objeto.

Palavras-Chaves: Carro autônomo. Reconhecimento de pedestres. Micro Doppler. Otimização multiobjectivos. *Support vector machine*.

ABSTRACT

The development of the autonomous car is nowadays a common practice in all the greatest automotive factories in the world, also in companies outside the automotive market, like Google and Apple. By adding more sensors, the vehicle is now capable of moving alone, identifying the correct path, the distance from another cars, also the presence of objects and people. However, there are still many issues blocking the autonomous car to be released. There are technical aspects to be solved, as the pedestrian recognition issues. Although, the recognition is widely studied and applied using cameras and digital images, there are issues to be improved. Like the distance and velocity reliability and the problems occurred because the lack of light in the environment. Based on the before mentioned, the focus in this presented work is to develop and discuss the efficiency of a pedestrian recognition system, using one automotive radar of 79 GHz. The main goal is to early detect the pedestrian using the micro Doppler characteristics of a human body in near to crash situations. Aiming this goal some techniques are used in the work. Firstly, the velocity resolution is improved, in order to extract the micro Doppler characteristics of the objects. The improvement of velocity resolution is reached by the use of multiobjective optimization techniques, as genetic algorithm and random search. The information about velocity and range is measured by the radar. In sequence a simple feature extraction method called video temporal gradient transform the data. The result is used in a machine learning technique called support vector machine (SVM). Which classifies the objects between pedestrians and non-pedestrians, with four different approaches. Concluding the work, it is possible to see the advantages of the multiobjective optimization in order to extract the micro Doppler effects. The optimization reached the velocity resolution of 0,12 m/s. The SVM comparison show that the fourth model with a polynomial kernel presented better result with accuracy 99,5%. However, the processing time of the system was not good enough taking 72 ms to identify an object.

Keywords: Autonomous car. Pedestrian recognition. Micro Doppler. Multiobjective optimization. Support vector machine.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ABS - Anti-lock Breaking System

ACC - Adaptive Cruise Control

AD - Analog to Digital

ADAS - Advanced Driver Assistance System

ADC - Analog to Digital Converter

AFE - Analog Front End

AR-LP- Auto Regressive Linear Prediction

CPU - Central Processing Unit

CW - Continuous Wave

DA - Digital to Analog

ESC - Electronic Stability Control

FMCW - Frequency Modulated Continuous Wave

FPGA - Field Programmable Gate Array

GIDAS - German In-Depth Accident Study

GPS - Global Positioning System

IF - Intermediary Frequencies

IO - Input Output

LFM - Linear Frequency Modulated

LIDAR - Light Detection and Range

LVDS - Low-Voltage Differential Signaling

MCDM - Multi-Criteria Decision making

m-D - micro Doppler

MLP - Multi Layer Perceptron

MOO - Multi-Objective Optimization process

MOOD - Multi-Objective Optimization Design

MOP - Multi-Objective Problem definition

NIS - Negative Ideal Solution

NSGA-II - Nondominated Sorting Genetic Algorithm

OEM - Original Equipment Manufacturer

OFDM - Orthogonal Frequency Division Multiplexing

PIS - Positive Ideal Solution

PLL - Phased-Locked Loop

PROMETHEE - Preference Ranking Organization Method for Enrichment of Evaluations

R&D - Research and Development

RADAR - Radio Detection and Range

RAM - Random Access Memory

RBF - Radial Basis Function

RF - Radio Frequency

SAE - Society of Automotive Engineers

SV - Support Vector

SVM - Support Vector Machines

TBD - Track-Before-Detect

TOPSIS - Technique for Order of Preference by Similarity to Ideal Solution

VCO - Voltage Controlled Oscillator

WHO - World Health Organization

CONTENTS

1. INTRODUCTION

The automotive industry, since the beginning of modern cars, is working to develop better cars to the customers, especially when the subject is safety. Vehicle safety became one of the biggest concerns of all big trends. Some devices were embedded to the vehicle over time, like three points seat belts in 1959, ABS breaks in 1978, airbags in 1980 and stability control in 1995.

Today, cars are safer than models that were developed one decade ago, but it is still not enough. The World Health Organization (WHO) [1] reported that nearly 1.24 million people die every single year in traffic accidents. Road injuries have been recognized as an international major public health concern. According to WHO, traffic crashes will become, in 2030, the fifth leading cause of death worldwide. Because of these circumstances, the safety of road transport is still a priority worldwide.

A considerable amount of these accidents occurs because imprudence or lack of dexterity of the driver. According to a recent study published by the *Virginia Commonwealth University* [2], more than seventy percent of traffic accidents are attributed to human error, such as errors of perception (i.e., distance judgment error), driving operation errors (on braking, for example) or simply by lack of attention [3].

Therefore, with the advance of technology, drivers assistant devices emerged. These systems are called Advanced Driver Assistance Systems (ADAS), and can be defined as the combination of technologies that provides relevant information to the driver, with the possibility of automating both difficult and repetitive tasks, and that provides an overall increase in safety of the car [4].

Global Positioning System (GPS) navigation, taking an example, has become increasingly the most common in original equipment manufacturer (OEM) infotainment systems since his first appearance in the 1990s. Adaptive Cruise Control (ACC), which can adjusts the speed of the vehicle automatically, to maintain a safe distance from vehicles ahead. Electronic Stability Control (ESC) can improve the stability of vehicle by detecting and reducing loss of traction (skidding). Both ESC and ACC are other examples of ADAS commonly found in the new vehicles nowadays.

Over the years the technology will be improved as well the ADAS devices and applications, therewith the cars will be safer also less dependent of the drivers, until reach a point of completely independency. At this point, the car would be called

Autonomous Car; the automotive industry is making many efforts to develop new cars that are briefly described in the sequence.

According to the society of Automotive Engineers (SAE) international (2014), there are six levels of autonomous driving [5]. The level zero is where only the driver control the entire vehicle all by himself, this is the most common nowadays. The level one, the car's system sometimes take the control over the longitudinal or the lateral dynamics but the driver must monitor the system permanently and be able to react immediately, like parking assist and ACC. The level two is very close to the level one, the only difference is that the car's system sometimes take the control over the longitudinal and the lateral dynamics, like a Traffic-Jam Assist. In the level three the driver would not monitor the vehicle permanently, instead, the driver would be alert by the car when to take over the control, an example is an autonomous driving in highways. In the level four, the system takes over the full control over the vehicle, in a special use case, and manages all possible situations for this use-case autonomously. In the level five the system control entirely the car, all people are passengers.

To reach the level of autonomous driving new systems are being developed, especially those concerning the safety of people inside and outside the vehicle. Cameras and different types of sensors are being installed together in the vehicles in order to detect the environment around the vehicle. At the same time, computational techniques of signal processing and pattern recognition are being developed and tested to be used in the field of Autonomous car.

For example, Tesla already have an autonomous system with eight surround cameras providing 360 degrees of visibility around the car at up to 250 meters of range. Twelve ultrasonic sensors, allowing for detection of both hard and soft objects. It has also radars, which are more capable to provide data through heavy rain, fog and dust [6]. While Google already tested its autonomous car over three million miles in complex city streets [7]. In the other hand, Uber is having lot of troubles with their self-driving vehicle. In the tests, the drivers have to take over control and the driving is not smoothly as it should be [8].

One important topic in autonomous cars is the pedestrian recognition, to avoid traffic accidents between them and vehicles. According to the Brazilian ministry of health [9, 10], more than 12,000 pedestrians dies in traffic accidents by year in Brazil. The main studies about recognition uses RADARs (Radio Detection and Range), cameras, LIDARs (Light Detection and Range) separately or a combination between

them, along with computational techniques of signal processing and pattern recognition.

The introduction chapter is divided in five sub-chapters. First an overview of the problem with the state of the art found in literature. The second, the main objective along with the specific ones are presented. The third sub-chapter discuss about the contributions of this work. In the fourth is presented the scientific methodology of the work. For last, the structure of the work is explained.

1.1. OVERVIEW

The overview of this work follows to workflow presented in figure 1**,** explaining why the techniques and methods were chosen. First the micro-Doppler (m-D) effects, then a method to extract the m-D effects, in the sequence the signal feature extraction of the signal, and for last the pattern recognition method to recognize the pedestrian.

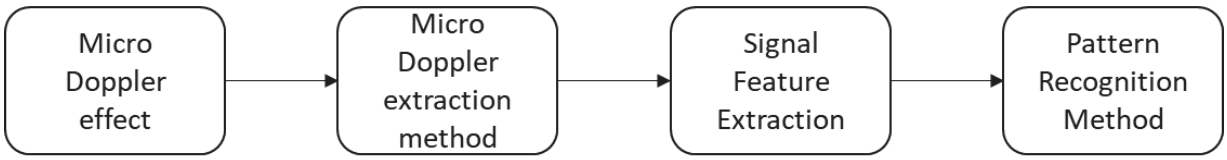


Figure 1 – Project workflow.
Source: The Author (2017).

To classify pedestrians in environments with multiples objects it is common to use the micro Doppler effects. They occur when, in addition to the constant Doppler frequency shift induced by the bulk motion of a radar target, the target or any structure on the target undergoes micro-motion dynamics, such as mechanical vibrations or rotations, the micro-motion dynamics induce Doppler modulations on the returned signal, referred to as the micro-Doppler effect [11]. Lan Du *et al.* presented great results when comparing the micro Doppler signatures of a walking pedestrian, two pedestrian and a wheeled vehicle [12]. Domenic Belgiovane and Chi-Chil Chen studied the micro Doppler signatures. Between pedestrians and bicycles, analytically and experimentally [13]. While Youngwook Kim and Hao Ling analyzed the micro Doppler signatures for seven different activities of the human body [14]. In all works, it is clear that with a clear micro Doppler signature it is easier to detect a pedestrian. Figure 2 illustrates the comparison of different micro Doppler signatures.

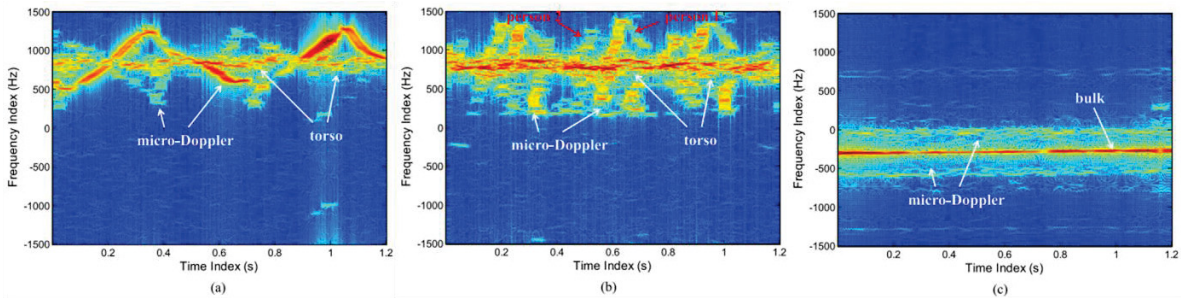


Figure 2 - Time-frequency spectrograms of some measured data: (a) single walking person; (b) two people walking; and (c) a moving wheeled vehicle.
Source: Lan Du et al. (2016) [12].

Markus Andres *et al.* used the micro Doppler effects of the human body to classify a pedestrian. In this study, they used auto-regressive linear prediction (AR-LP) creating extra artificial waves in the wave modulation without increasing the measurement time. With more waves, it is easier to extract the m-D effects [15]. In addition, T. Thayaparan *et al.* applied a quadratic time-frequency S-method-based approach in conjunction with the Viterbi algorithm to extract the micro Doppler signatures [16]. Both of the previous mentioned methods and others not mentioned uses complex mathematics calculations continuously with the transmission of the radar signal. This imply in more time to make more calculations in each measured frame, which is a problem for real applications.

To avoid the many calculations on every frame a simpler way to extract the micro Doppler effects can be used. The extraction of the micro Doppler effect could be considered as the minimization of the radar minimum detectable velocity resolution. Thereby an optimization method could calculated the best radar parameters for improving the velocity resolution function before the radar measurements. As a result, only one calculation would be made. On other hand, there are more functions affected by the same radar parameters, it would need a multiobjective optimization design (MOOD) method.

In the literature, Ajit Kumar Sahoo and Ganapati Panda presented a multiobjective optimization (Nondominated Sorting Genetic Algorithm-II (NSGA-II)) to optimize the radar parameters of a linear frequency modulated (LFM) pulse train to achieve reduced grating lobes, low peak sidelobe level and narrow mainlobe width [17]. In addition, Yongjun Liu *et al*. proposed a multiobjective optimization problem to achieve the adaptive weighted-optimal and Pareto-optimal waveform design. Which could improve simultaneously the accuracy of range and velocity in radar and the

channel capacity in communication [18]. There are also other uses of multiobjective optimization in radar problems. Vikas Baghel *et al.* used multiobjective genetic algorithm and radial basis function (RBF) networks to compress the performance for range resolution [19]. Satyabara Sen *et al*. proposed the same multiobjective genetic algorithm to design an orthogonal frequency division multiplexing (OFDM) radar signal. This way, it would minimize the upper bound on the estimation error, improving the efficiency of sparse-recovery and maximize the squared Mahalanobis-distance, increasing the performance of the underlying detection problem [20].

After extracting the m-D effects of a human body, the classification should separate the pedestrians from the other objects. However, it is possible to facilitate the classification task using methods to extract important features from the signals before the classification. Many methods could be used with the m-D effects to extract better features. Igal Bilik *et al*. presented good classification results using features based on physical model-based features [21], while Burkan Tekeli *et al*. used information theoretic features [22] and Igal Bilik *et al*. used speech processing motivated features (cepstrum, mel-frequency cepstrum, linear predictive coding) [23, 24]. The method used in this work, proposed by Alexey Castrodad and Guillermo Sapiro, uses the video temporal gradient and a sparse dictionary previous learned as a feature for classification [25]. Because of its simplicity and the low cost in terms of calculations. The gradient compares the data between two frames in a video, while the dictionary has the information about the kinds of trajectories and the characteristics of each object. However, for this work, it was not used the sparse learning dictionary, because there were too few kinds of objects to made another classifier.

With the principal features of the signal, it is needed to classify it into pedestrian or not. Many ways of classifying pedestrians are found in the literature. M. Heuer *et al*. proposed a classification method, for pedestrian recognition, using a track-before-detect (TBD) filter to incorporate raw measures [26]. While, Hermann Rohling *et al.* presented a pedestrian detection procedure, based on 24GHz automotive radar, classifying the data with a well-known powerful polynomial classification [27, 28]. Machine learning methods are also used to classify pedestrians; Seongkeun Park *et al.* used a multi layers perceptron (MLP) neural network to classify pedestrians, based on the different reflections characteristics of people and car to microwaves [29]

Another machine learning method wide presented in the literature for pedestrian recognition is the support vector machine (SVM). Seongwook Lee *et al.* proposed a

SVM method, using the root radar cross section, which represent the reflection characteristics of targets, to classify pedestrians and cars [30]. Also, W. J. Liu *et al.* presented a study using two SVM classifiers and an integration scheme to be used with radars for intersection surveillance [31]. While, Steffen Heuel and Hermann Rohling proposed a target recognition system in urban scenarios, using SVM, to classify the human and lateral car moving in feature extraction and classification [32].

1.2. OBJECTIVES

As mentioned before, there are many researchers studying how can a car detect people. Many different sensors are used in these studies. The most used are cameras, or cameras merged with other sensors like RADARs and LIDARs. The first problem with cameras, is that they do not work well when there are lack of light in the environment. When it is night, or when it is raining or foggy. Another problem is the complex calculation to extract information, and sometimes distance and velocity are not trustworthy. For the reasons previously mentioned the camera was not selected for this work. The selected sensor was the RADAR, which do not depend on the light of the environment, it is easier and faster to da the extraction of the velocity and range, and the values are trustworthy.

The scope of the work was defined by the CARISSMA lab. The intention of the work is to early detect the pedestrian in situations near to the crash (0-15m). After the completion of this work, another work will calculate when the crash with the pedestrian is unavoidable, and with this information, security systems as airbags and seat belts would be activated before the crash.

As mentioned in the overview, some techniques were chosen to do the pedestrian detection. First, the use of the micro Doppler effect to detect a pedestrian. The use of these effect were suggested by the laboratory, and they presented good results in pedestrian detection as mentioned before. The second technique was the multiobjective optimization. It is not found in the literature for defining the FMCW modulation, but for other types of wave modulation and radar configuration. Because this method makes the calculation just once, before the RADAR start collecting data, it is much faster than the other methods presented in the overview. In addition, the last method was the support vector machine, for the pedestrian detection. This method was

chosen because its simplicity and for dealing very well for problems with high non-linearity. Also is commonly found in the literature.

To synthetize, the main goal of this work is to early detect a pedestrian in a traffic scenario using a 79 GHz RADAR and considering the micro Doppler Effects of a human body. According to the aforementioned goals of this work, it is possible to enumerate the specific objectives as follows.

- Summary and conclusion of literature studies;
- Study of the Radarlog and its parameters;
- Optimize the Radar velocity resolution, with multiobjective techniques, in order to extract the micro Doppler effect of pedestrians;
- Create a testing set-up for radar with focus on pedestrian safety;
- Generate a database with the data from all the tests;
- Implement a detection method using support vector machines;
- Validate the support vector machine method using different configurations and the data from the tests;
- Compare the results and conclude the work.

1.3. CONTRIBUTIONS

The first contribution of this work was the application of a mathematical multiobjective optimization technique in a frequency modulated continuous wave, in order to extract the micro Doppler effects. In the literature, there are many uses of this technique using radars [17-20] but not with the parametrization of a FMCW. The closest work found in the literature uses the multiobjective optimization to parametrize an OFDM wave in order to improve the velocity and range accuracy [20]. The result of the optimization was a velocity resolution of 0.43 *km/h* using only the multiobjective optimization. The work of *Andres et al.* [15] reached a velocity resolution of 0.45 *km/h* using a linear prediction to simulated artificial chirps. This linear prediction called AR-LP could be added to this work improving even more the velocity resolution.

The second contribution was the beginning of a data bench with traffic accidents involving pedestrians in the near field. It was made 220 different tests with a person crossing the street in front of a car in different ways. This data bench was just the

beginning of a larger work. In sequence more pedestrians and more objectives will be added to the data bench.

The third contribution was the comparison of the classifiers. Four different methods to train the SVM were discussed for the problem. The third and fourth method were not find in the literature, being an original contribution of this work.

Moreover, the last contribution was the result of the work. *Park et al.* proposed a work using multi-layer perceptron neural networks to detect pedestrian [29]. They used the radar cross section as the most important feature in the classifier, and they made tests with the pedestrian and a vehicle in front of the radar. They achieve an accuracy of 92.78%. *Heuel and Rohling* presented a pedestrian recognition using a 24 GHz radar based on the radar cross section, using a support vector machine classifier. In their data bench, they classified pedestrian between vehicles and bicycles. They achieve an accuracy of 94.2%. In addition*, Lee et al.* used support vector machines do detect pedestrians. Their tests classified pedestrians from longitudinal cars, lateral cars and others. They reached an accuracy of 96.3%.In this work the classifier achieve an accuracy of 99.5% using support vector machine based on the micro Doppler effects. However the better results, should be considered that the data bench was simpler than the ones compared previously in this paragraph. The tests of this work contained the pedestrian, noise and few objects.

1.4. METHODOLOGY

Considering the application field of pedestrian recognition with radars through machine learning techniques, the research methodology adopted during the development of this project can be defined.

As main characteristic, this work has an applied nature, since it pursuits the state-of-the-art of automotive pedestrian recognition. In addition, it presents a comparison between four different training methods and three different Kernel functions, using the same radar. As problem approach, the work is a quantitative research; the methods performance and the comparison of the results are all expressed in a numerical form.

About the objectives, this research, in the beginning, has an exploratory study, evaluating the previously literature of radar modulations wave and support vector machine techniques. In sequence, this research adopts an explanatory study, where

using computational simulations, intends to compare which method works better with radar data, and which one has the best performance.

Evaluating the technical procedures, the research has an experimental nature, since many computational and real time simulations are used to analyses this study.

1.5. STRUCTURE

The presented document is organized in six main chapters, being this introductory chapter the first. The second chapter elucidates the radar structure and its wave modulations. The third chapter is devoted to briefly explain the technique called support vector machine and its variables. The fourth chapter describes all the work developed inside the CARRISMA labs at Ingolstadt, Germany. The fifth chapter shows the results in terms of accuracy and in time of performance. The sixth chapter contains the conclusion of the work and the next steps. At last, is presented the radar hardware structure in Appendix A.

2. RADIO DETECTING AND RANGE (RADAR)

The aim of the present chapter is to state basic concepts regarding radio detection and range devices, known by RADAR. The first part is devoted to introduce the RADAR as well as its different kinds, uses and configurations. The second part a brief explanation about the different types of frequency modulation and its characteristics. The third part is to explain Doppler and Micro Doppler effects and how they are important.

2.1. INTRODUCTION

Radar is an electromagnetic system for the detection and location of objects. It operates by transmitting a particular type of waveform and detects the nature of the echo signal. The RADAR was designed to detect through the conditions that the human eye cannot see perfectly, as fog, haze and darkness. In addition, it has the advantage of being capable of measuring the distance or range of the object [33].

The principle of the RADAR is based on a transmitting antenna emitting electromagnetic waves generated by an oscillator, a portion of the transmitted signal from the waves are intercepted by an object (target), which reflects them in all directions. A receiving antenna absorbs the energy reflected in the direction of the radar and deliver to a receiver. The receiver antenna processes the energy to detect any target, and measures the position and the relative velocity. The angular position (the direction of the arrival of reflected waves) and the distance are measured by the time delay between the emitting wave and the receiving wave. For the velocity, it measures the frequency shift between transmitted and received signal, called Doppler Effect, to distinguish moving targets [34].

Radar's usually are anisotropic radiators, where the transmitting antennas are designed to transmit more energy in one direction than in another. These antennas are called directives and they employ the radiated power P_t into some particular direction generating a gain G_T. This variable of an antenna is a measure of the increased power radiated in the direction of the target as compared with the power that would have been radiated from an isotropic antenna [33]. The equation 2.1 presents the anisotropic Radar radiated power.

$$P_D = \frac{P_T G_T}{4\pi R^2}, \tag{2.1}$$

where P_t is the radiated power, G_T is the antenna gain and R is the distance from the radar to the object. When the RADARs are isotropic, the gain is withdraw from the equation.

The power, after transmitting, is reradiated when intercepted by an object in many directions. The antenna measures a portion of the reradiating power. This portion is called radar cross section, and it is represented by σ. Equation 2.2 defines the power density of echo signal at radar.

$$P_e = \frac{P_T G_T}{(4\pi)^2 R^2} \frac{\sigma}{4\pi R^2}. \tag{2.2}$$

The cross section has particular characteristics about the size of the target. If the effective area of the receiving antenna is denoted A_e, the power P_R received by the radar is presented in equation 2.3 [33].

$$P_R = \frac{P_T G_T}{(4\pi)^2 R^2} \frac{\sigma}{4\pi R^2} A_e = \frac{P_T G_T A_e \sigma}{(4\pi)^2 r_{TO}^2 r_{OR}^2}, \tag{2.3}$$

where, r_{TO} is the distance between transmitting antenna and the object and r_{OR} is the distance between object and the receiving antenna. Equation 2.4 presents the relation between receiving antenna gain G_R and effective area A_e.In addition, the final radar equation is illustrated in 2.5.

$$G_R = A_e \frac{4\pi}{\lambda^2}, \tag{2.4}$$

$$P_R = P_T \frac{G_R G_T \lambda^2 \sigma}{(4\pi)^3 r_{TO}^2 r_{OR}^2}. \tag{2.5}$$

2.2. WAVE MODULATION

One way to classify the radar is by its wave modulation. Modulation is the process of variation of amplitude, intensity, frequency or phase of a wave. The two

more common modulations are the Pulse radar and the Continuous Wave radar (CW). However, there are more types, as the Synthetic Aperture radar, the Phased Array Radar and secondary radars [33].

In the next sub chapter the pulse radar and the continuous wave are explained, in addition the micro Doppler effect and its particularities are highlighted.

2.2.1. Pulse radar

A pulse radar is a device that emits short and powerful pulses, while in the silent period receives the echo signals. Pulse radars are mostly designed for long distances and transmit a relatively high pulse power [33]. Figure 3 illustrates a pulse radar wave can be seen. Where T_P is the period of the transmission, τ_P is the pulse width and τ the silent time between the transmitted signal and the received.

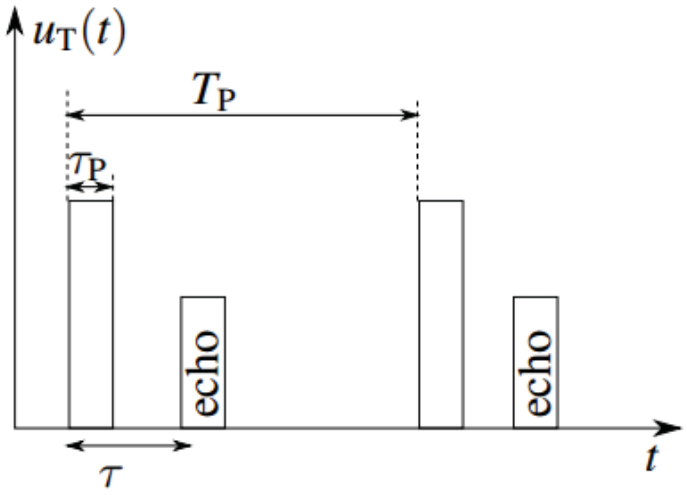


Figure 3 - Pulse radar wave.
Source: Integrated Safety and Assistance Systems, Class 3, THI [35].

The equation 2.6 represents the waveform of the pulse radar, described mathematically. $A(t)$ is a function of the amplitude in time, when the antenna is transmitting the value is 1, otherwise 0. f is the pulse repetition frequency, τ is the receiving time and $\varphi(t)$ is the phase of the signal. Equation 2.7 illustrates the RADAR range resolution, where the duration of the pulse τ_P have influence in the radar performance and c represents the light speed.

$$s(t) = A(t)\sin[2\pi f(t)\tau + \varphi(t)] \,, \tag{2.6}$$

$$\Delta r = 0.5\ \tau_P\ c. \tag{2.7}$$

The pulse Radar can be monostatic or bistatic. The first one, the transmitter and the receiver are in the same site. One of the disadvantages of this device is the necessity of a duplexer, a device that prevent the high power from the transmitter to cause damage to the receiver, also channel the echo signal to the receiver and not the transmitter. When the antenna is transmitting, it cannot receive any signal. The advantage is that the design is compact and time devices can be used in a central synchronization block to change the antenna between transmission and reception [33].

Figure 4 shows a simple block diagram of the monostatic pulse radar. The transmitter block is responsible of sending short duration radiofrequency (RF) pulses that are radiated by the antenna, the receiver amplify and demodulate the RF signals. The duplexer, as commented before, protect the receiver from the high energy from the transmitter and channels the echo pulses to the receiver. The modulator shapes the signal of the amplitude $A(t)$ and the synchronizer controls the timing devices responsible to control the pulses and echo signals.

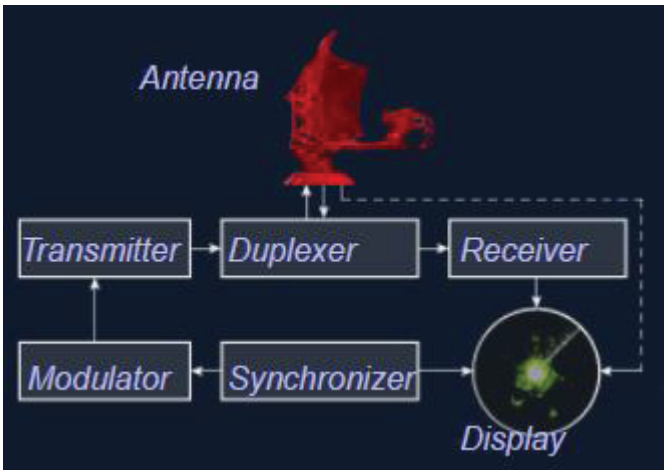


Figure 4 – Block diagram of a Monostatic pulse radar.
Source: Radar Tutorial [36].

In the bistatic pulse radar, the transmitter has its own antenna in a different location of the receiver's one. The first advantage is that the protection for the receiver from high energy originated in the transmitter is not needed anymore. The second is that the echo signal can be received while the pulses signals are being generated.

The pulse radar uses the time of flight principle to measure the distance to a target. This principle is based on the time difference between the emission of a signal and its return to the sensor, measuring the distance between a sensor and the target. Equation 2.8 represents the distance r. Moreover, equation 2.9 explains the maximal distance to unambiguously detect objects r_{max} , where T_P is the period between pulses [33].

$$r = \frac{c}{2}\tau \ , \tag{2.8}$$

$$r_{max} = \frac{c}{2} T_P \ . \tag{2.9}$$

2.2.2. Continuous wave radar

Continuous wave radar (CW) transmits a high frequency signal continuously. With that, the echo signal is received and processed permanently. To prevent the energy from the transmitter into the receiver, the antennas are spatially separated from each other. Simple continuous wave radar devices have the disadvantage that they cannot determine target range, because it lacks the timing mark necessary to allow the system to time accurately the transmit and receive cycle and to convert this into range [33, 34].

Because of that, one specific case of continuous wave radar, called Frequency Modulated Continuous Wave radar (FMCW), is used. The FMCW radiates continuous waves as a continuous wave radar, but in contrast to the CW, the FMCW radar can change its operating frequency during the measurement, modulating the transmission signal in frequency (or in phase). Possibilities of Radar measurements through runtime measurements are only technically possible with these changes in the frequency (or phase).

To measure the distance, this method transmits a signal which increases or decreases in the frequency periodically. When an echo signal is received, that change of frequency gets an echo delay τ as the pulse radar technique. In pulse radar, however, the runtime must be measured directly. In FMCW radar the runtime is measured by the differences in phase or frequency between the transmitted and the received signal.

An example of a FMCW is presented in the figure 5, where T is the period of the signal, τ is the delay in the received signal, Δf is the frequency variation of the signal and δf is the frequency difference between transmitted and received signal.

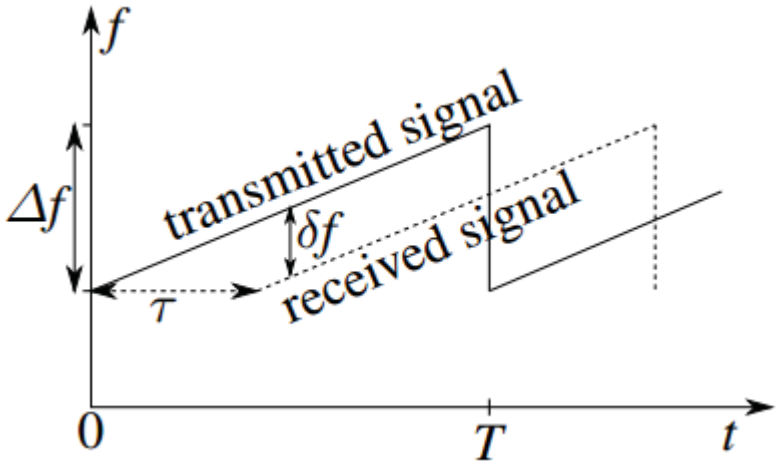


Figure 5 – Continuous wave radar wave.
Source: Integrated Safety and Assistance Systems, Class 3, THI [35].

Based on the figure 5 and using the concept of similar triangles, the relation in equation 2.10 can be assumed:

$$\frac{\tau}{T} = \frac{\delta f}{\Delta f} . \tag{2.10}$$

By isolating the echo delay and substituting it on the equation 2.8, the distance can be defined as equation 2.11 for a FMCW modulation.

$$r = \frac{c}{2}\tau = \frac{c}{2} T \frac{\delta f}{\Delta f} . \tag{2.11}$$

One great advantage of the FMCW is the capability of measuring the relative velocity, even though the radar can only measure the relative velocity of a target in comparison with the source. For that, the radar uses the Doppler Effect. This effect happens when the receiver is moving relative to its source producing a change in the frequency of a wave. When receiver and source move depart from each other, the arrival time between successive waves increases, reducing the frequency, when they

approach each other the frequency increases. The difference of frequency is called Doppler frequency and it is expressed as f_D. The Doppler frequency is presented in the equation 2.12, where f_0 is the initial frequency, λ_0 is the length of the wave and $v_{r,rel}$ is the relative velocity of the target [33, 34].

$$f_D = \left(f_0 - f_0 \frac{2v_{r,rel}}{c}\right) - f_0 = -f_0 \frac{2v_{r,rel}}{c} = -\frac{2\,v_{r,rel}}{\lambda_0}. \quad (2.12)$$

Isolating the relative velocity from the equation above, the equation 2.13 is deduced.

$$v_{r,rel} = -\frac{f_D \lambda_0}{2} = -\frac{f_D c}{2f_0}. \quad (2.13)$$

Besides velocity and range, is also possible to measure the azimuth angle of the targets. There are mainly three different methods to do so. The first is scanning with a mechanical rotation radar. With the spatial rotation of the receiving antenna, it is possible to know the angle where the signals are coming; the second uses two separate antennas, Rx_1 and Rx_2 to measure the angle, as the figure 6 explains, where ϕ is the angle of the target and $\Gamma\lambda$ the distance between the antennas.

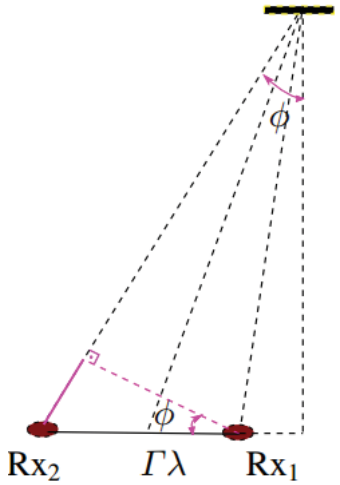


Figure 6 – Target's angle measurement
Source: Integrated Safety and Assistance Systems, Class 4, THI [35].

Considering the echo delay of the antenna Rx_1 as τ_1 and the echo delay of the antenna Rx_2 as τ_2 the equation 2.14 is assumed:

$$\tau_2 = \tau_1 + \frac{\Gamma\lambda \sin(\phi)}{c}. \quad (2.14)$$

The other method is the multiple beam procedure. Multiple beams are generated by multiple antennas and signal processing. Their characteristics pattern is stored and the comparison of the received power of each beam estimates the angle. Figure 7 shows an example of the multiple beams amplitude per aperture angle.

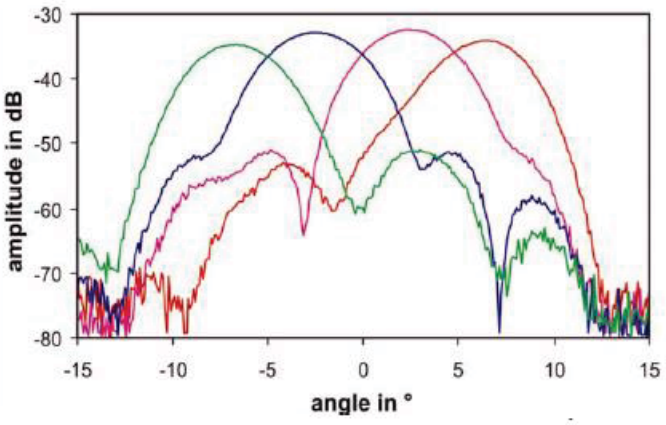


Figure 7 – Multiple beam procedure graph.
Source: Radar Tutorial [36].

The FMCW has a block diagram different from the pulse radar. It consists essentially of the transceiver and a control unit with a microprocessor. Figure 8 presents its diagram.

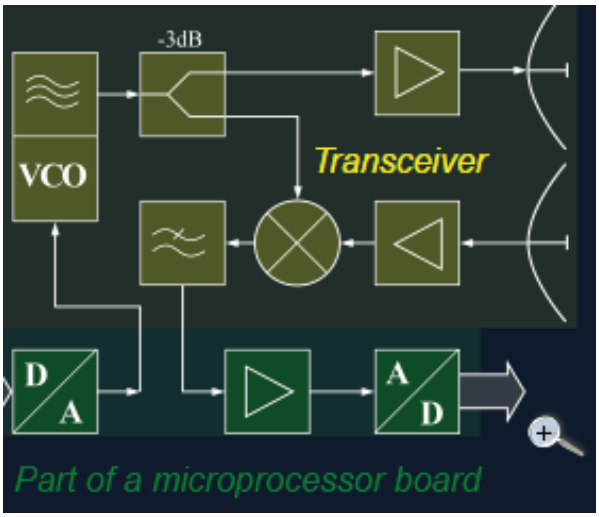


Figure 8 – FMCW block diagram.
Source: Radar Tutorial [36].

Connected to each antennas has an amplifier, for the transmitter antenna this amplifier amplifies the waveforms to the desired transmit power. Before the transmitter amplifier, the digital control passes throw a digital-analog (DA) converter, and then the analog signal changes its frequency in the Voltage Controlled Oscillator (VCO), based on the control command. At last the -3dB coupler branches off part of the of transmission power and then comes the amplifier.

In the receiver one, the signals are extremely weak, so they are amplified to be processed by the mixer. In the mixer, the echo signals are compared to the transmission frequency and as output is the difference between them. Then the signal passes throw a low pass filter to block the unwanted frequencies. The signal is again week so it is transform throw another amplifier and then is finally converted by an AD (Analog to digital) converter, transforming the signal into digital.

2.3. DOPPLER AND MICRO DOPPLER EFFECT

As commented before The Doppler- Effect is the apparent change in frequency when a wave source moves either toward or away from the target, or when the target moves either toward or away from the sound source [33].

There is an object called micro-Doppler signature, this signature occurs when μD signatures are generated by smaller objects, or part of objects in movement.

As an example the human body. Human motion involves the periodic movement of different body parts such as head, torso, two arms and two legs. All these movements have different velocities in time, and these different velocities result in many Doppler effects of smaller intensity, called micro-Doppler [35].

It has been observed that the periodic movement of torso and legs are the main contributions to the μD signature. Within the walking cycle, the torso is moving with a nearly constant mean velocity. However, the legs decelerate and accelerate depending on the actual stance or swing phase [35].

The extraction of these characteristics is very important in the safety automotive field. Because it is possible to distinguish using radars, people, cyclists and even animals from rigid bodies like cars. However, to extract these signatures, high separation abilities in the velocity domain are needed to improve the velocity resolution.

Figure 9 presents a comparison between two graphs. The first one is the velocity versus range of a pedestrian measured by one radar, and the other is the same for a car. In addition, figure 10 compares the sequential velocity along time of an inline skater with a car.

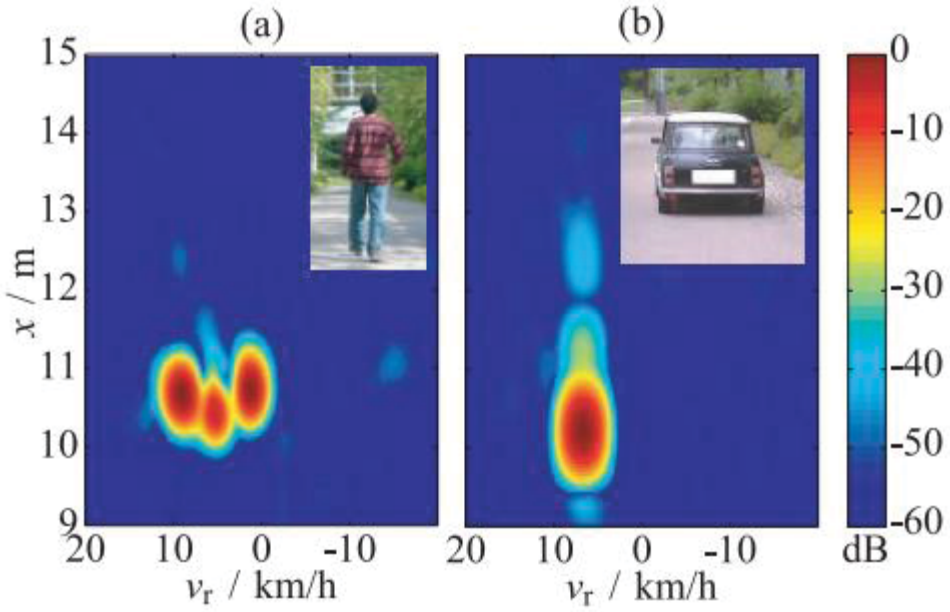


Figure 9 - Measurement of the Doppler profile of a slow moving (a) pedestrian and (b) car. Source: ADRES, M; ISHAK, K; MENZEL, W; BLOECHER, H. L (2016) [15].

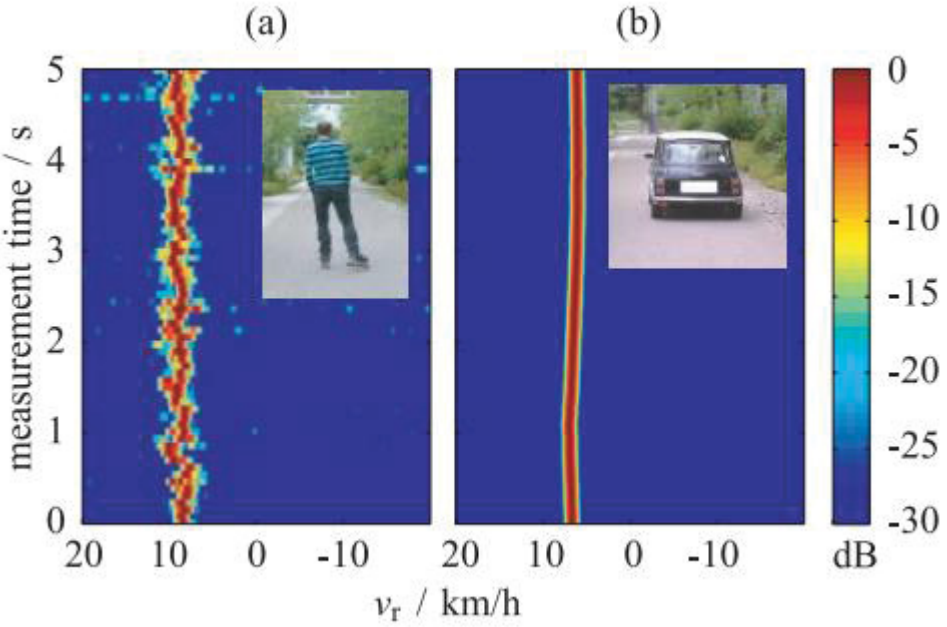


Figure 10 - Comparison on the sequential µD-signature of a (a) inline skater and (b) car. Source: ADRES, M; ISHAK, K; MENZEL, W; BLOECHER, H. L (2016) [15].

3. SUPPORT VECTORS MACHINE

In machine learning and pattern recognition, classification refers to the task of point out for one group of entry data, called characteristic vector, the class that it belongs. There are two major groups of learning, the supervised and the unsupervised.

Supervised learning is the task of inferring a function from labeled training data [37]. In supervised learning, the training set consists of input patterns as well as their correct results in the form of the precise activation of all outputs. Thus, for each training set that is fed into the method an output is generated. This output can directly be compared with the correct solution resulting in changes in the method, according to their difference [38]. Although this learning process is extremely effective and practicable, is not always biological plausible.

In contrast to supervised learning, the unsupervised learning has the task of inferring a function to describe hidden structure from "unlabeled" data. The training set only consists of input patterns, the method tries by itself to detect similarities and to generate pattern classes [38]. Opposite to the last task, this one is biological plausible, but is not suitable for all problems.

The aim of this chapter is to present the concepts about the supervised Support Vector Machine, the technique chosen to make the pedestrian recognition. The chapter is divided in three sub chapters, two sub-chapters explaining the soft and the hard margins and another one explaining the nonlinear SVM.

3.1. LINEAR SVM WITH HARD MARGINS

Two data sets are linearly separable if the patterns of their different classes can be separated with at least one hyperplane. However, for the linearly separable data, exists infinite hyperplanes capable of correctly separating them [40]. The hyperplane choice has directly consequences in the generalization capacity of the recognition system. Figure 11 presents two data sets linearly separable by one hyperplane.

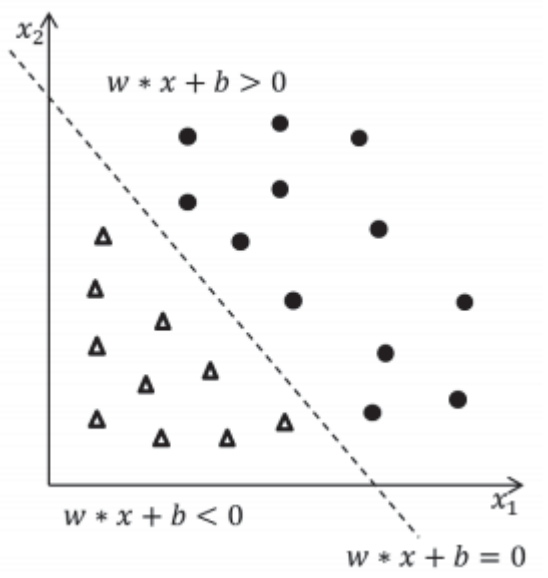


Figure 11 - Data set separable for one hyperplane.
Source: Adaptation of Burges (1998) [39].

Vapnik (1998) creates a method to choose which hyperplane to be used, maximizing the margins. The margin is defined with the maximum distance between the closest points of each class. This method defines the optimal hyperplane maximizing the system generalization, if used good representative's samples of the entire class. Figure 12 presents the maximum distance of a hyperplane between the closest points of each class [41].

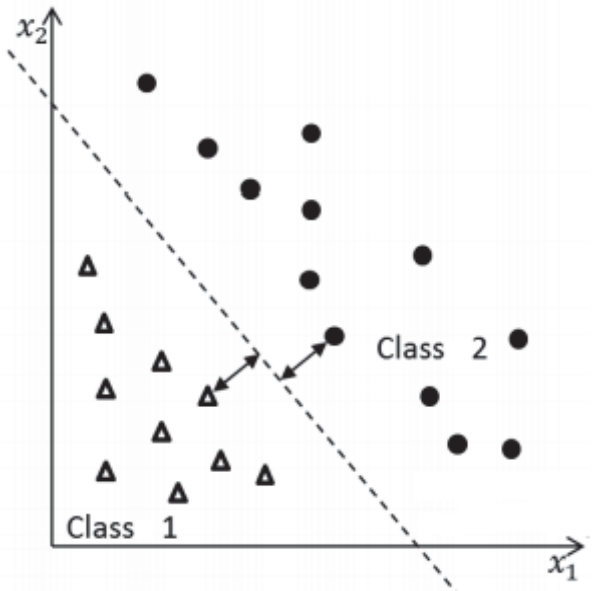


Figure 12 –Maximum distance hyperplane.
Source: Adaptation of Burges (1998) [39].

To determine the optimal linearly separable hyperplane, variables w and b are created, with the intention of making the closest points to the hyperplane, of each class, satisfy the hyperplane canonical representation equation, this equation is used along the work to determine the optimal hyperplanes (3.1) [39].

$$|w.x + b = 1| \ . \tag{3.1}$$

With this description, equation 3.2 is defined as a linear classifier that separates the given set with a positive margin. Analyzing this system, is observed that there are no points between the planes $w.x + b = 0$ and $|w.x + b = 1|$, thus, the distance between any point of one of the classes and the separation hyperplane, will be greater than or equal to the distance between the planes $w.x + b = 0$ and $|w.x + b = 1|$. Being the rigid margin SVMs then defined according to equation 3.2, where w is a *m*-dimensional vector and b is the compensating term.

$$w.x + b \begin{cases} \geq +1, & to \ \ y_i = +1 \\ \leq -1, & to \ y_i = -1 \end{cases} . \tag{3.2}$$

The values of the right side of the equation could be any constant a with $a > 0$ and $-a$, respectively. However, when both sides are divided by a, it goes back to equation 3.2 [39]. The equation 3.3 is a different representation of the previous one.

$$y_i(w.x + b) \geq 1 \ \ to \ \ i = 1,2, \dots M \ . \tag{3.3}$$

Now, considering $x_1$and x_2 above the hyperplanes $w.x + b = 1$ and $w.x + b = -1$, respectively, and considering x_1 intercepts the perpendicular line with x_2 , like in the figure 13, the equation 3.4 is defined. In addition, with the solution of this equation system the equation 3.5 is obtained.

$$\begin{cases} w.x_1 + b = -1 \\ w.x_2 + b = +1 \end{cases}, \tag{3.4}$$

$$w.(x_1 + x_2) = 2 \ . \tag{3.5}$$

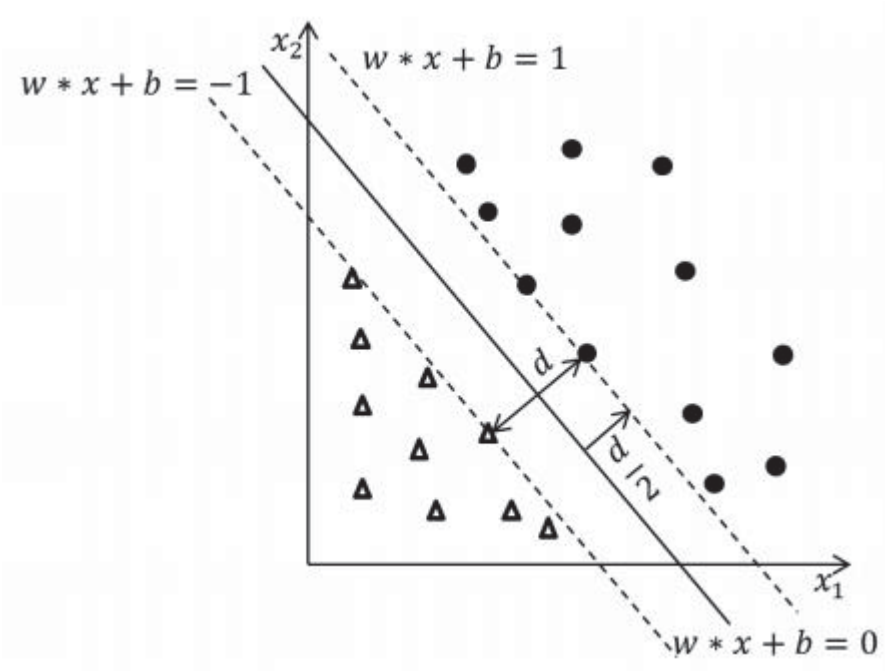


Figure 13 - Distance between hyperplanes
Source: Adaptation of Burges (1998) [39].

Knowing that w and $x_1 - x_2$ are orthogonal to the hyperplane and parallel with each other, also knowing the vector's norm is $\| . \|$, the equation 3.6 is obtained. Also replacing the equation 3.5 in 3.6 reaches the equation 3.7.

$$|w.(x_1 + x_2)| = \|w\| \times \|x_1 + x_2\| , \tag{3.6}$$

$$\|x_1 + x_2\| = \frac{2}{\|w\|} . \tag{3.7}$$

Burges (1998) concludes that the distance between the planes $w.x_1 + b = -1$ and $w.x_1 + b = +1$ is $\frac{2}{\|w\|}$, since it is the same distance of x_1 and x_2 given in $\|x_1 + x_2\|$. In addition, the distance between the optimal plane $w.x_1 + b = 0$ and both planes can also be defined as $\frac{1}{\|w\|}$, the size of the margin. Once the objective is to maximize the value of the margin, that is, minimize the value of $\|w\|$, the optimization system is defined in equation 3.8 [39].

$$Minimizing \ \|w\|^2 \tag{3.8}$$
$$Subject\ to: y_i(w.x_i + b) \geq 1 \ \ to \ \ i = 1,2, \dots m .$$

This optimization can be solved using the Lagrangian function, as presented in equation 3.9, defined in terms of w and b , where α are Lagrangian multipliers vector. The optimal values of this optimization can be discovered in the equations 3.10 and 3.11.

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i\,(w.x_i + b) - 1)\ , \tag{3.9}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0\ , \tag{3.10}$$

$$\frac{\partial L}{\partial w} = w = \sum_{i=1}^{n} \alpha_i y_i\, x_i\ . \tag{3.11}$$

Replacing in the equation 3.9 the equations 3.10 and 3.11 is defined the dual maximization problem, presented in equation 3.12.

$$Maximizing \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j x_i\,.\,x_j$$
$$Subject\ to: \begin{cases} \alpha_i \geq 0, i = 1,2 \ldots mn \\ \sum_{j=1}^{n} \alpha_i y_i = 0 \end{cases}, \tag{3.12}$$

Obtaining the maximum value of α, called α^*, and replacing it on the equation 3.11 is possible to find the value of w^* (optimal w) With the values find for w^*, b^* can be defined as the equation 3.13. Where x_{class1} and x_{class2} are the training patterns of class 1 and class 2 respectively.

$$b^* = -\frac{1}{2}[\max(w^*.x_{class1}) + \min(w^*.x_{class2})]\ . \tag{3.13}$$

Throw the equations previously demonstrated the optimal values $w^*, \alpha^*$and b^* are obtained, that is, the training of the SVM. When the value of α^* is positive the point in question is above the margin ($\frac{1}{\|w\|}$ units from the optimal hyperplane). These points are called support vectors, and them are responsible for the determination of the optimal plane. In this way, w^* can be defined as in equation 3.14.

$$w^* = \sum_{x_i \in SV} \alpha_i^* y_i x_i \ . \tag{3.14}$$

For every other point α^* is equal to zero. Finally with the optimal values of the three training variables a new pattern called x_{new} can be classified (C) throw the equation 3.15.

$$C(x_{new}) = sign\ (\sum_{x_i \in SV} \alpha_i^* y_i x_i \ . x_{new} + b^*) \ . \tag{3.15}$$

3.2. LINEAR SVM WITH SOFT MARGINS

When it is defined soft margins for the support vector machine, the idea that there is no point between the margins, $w.x_1 + b = -1$ and $w.x_1 + b = +1$, and the hyperplane, $w.x_1 + b = 0$, is negleted. The reason of this assumption is that in real problems, the situation with no points in the margins rarely occurs, mostly due the noise in the data or in its measures [40]. Figure 14 shows one case of linearly inseparable classes.

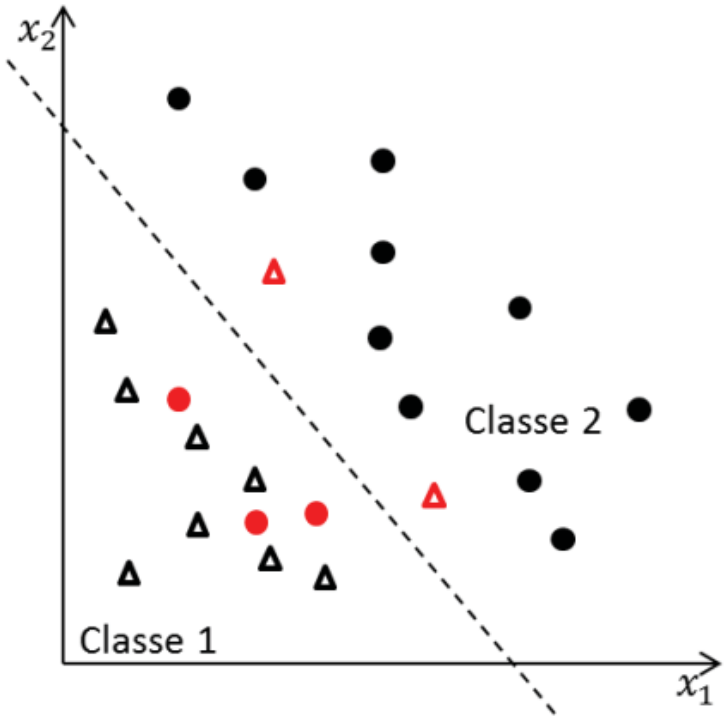


Figure 14 – Data linearly non-separable.
Source: Adaptation of Burges (1998) [39].

Shilton (2006) proposed a SVM adaptation of the hard margins. He introduces the concept of relaxing, in a way to permit some classification errors, softening the restrictions in the determination of the optimal hyperplane. Variables of softening were introduced in the problem formalization. The equation 3.16 introduces this new variable in the problem [40].

$$y_i(w.x_i + b) \geq 1 - \xi_i \;\; to \;\; i = 1,2,\dots M \; . \qquad (3.16)$$
$$\xi_i \;\geq 0$$

When $\xi_i > 1$ the ith inequality is violated in comparison with the inequality in the hard margins SVM. Therefore, the optimization problem can be defined now as equation 3.17, where c is a real positive constant. In addition, the Lagrangian function must be consider as in equation 3.18.

$$minimizing \; \|w\|^2 + c\sum_{i=1}^{n}\xi_i \; , \qquad (3.17)$$
$$Subject\ to: \begin{cases} y_i(w\,.x_i + b) \geq 1 - \xi_i \\ \xi_i \;\geq 0 \end{cases} ,$$

$$L(w,b,\xi,\alpha,\gamma) = \|w\|^2 + c\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i\,(w.x_i + b) - 1 + \xi_i) - \qquad (3.18)$$
$$\sum_{i=1}^{n}\gamma_i\xi_i \; .$$

With $\alpha > 0$ and $\gamma > 0$, the late is the conjunct necessary for the new relaxing variables ξ_i. With this definition, the equations 3.19, 3.20 and 3.21 are obtained.

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n}\alpha_i y_i = 0 \; , \qquad (3.19)$$

$$\frac{\partial L}{\partial w} = w = \sum_{i=1}^{n}\alpha_i y_i \, x_i \; , \qquad (3.20)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \;\rightarrow 0 \leq \alpha_i \leq c, i = 1,2,\dots,m \; . \qquad (3.21)$$

Replacing the equations 3.19, 3.20 and 3.21 in the equation 3.18 the dual problem is reached, which can be resolved with the quadratic programming as shown in equation 3.22.

$$Maximizing\ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i\ .\ x_j\ , \qquad (3.22)$$
$$Subject\ to: \begin{cases} 0 \le \alpha_i \le c, i = 1,2 \dots m \\ \sum_{j=1}^{n}\alpha_i y_i = 0 \end{cases}.$$

Only the restriction about the superior limit of α is modified in comparison with the hard margins. Solving this optimization problem the values of α^* are obtained. The w^* and b^* are obtained in the same way as the hard margins SVM, as well the new classification of new patterns. The position of the pattern i in relation to the margins and to the optimal hyperplane can be calculated as the equation 3.23, where the ς_i specifies the position. If $\varsigma_i = 0$, the pattern was classified correctly, if the $0 < \varsigma_i \le 1$, the pattern was correctly classified, but the distance to the hyperplane is less than the margin. And if $\varsigma_i > 1$, the pattern was classified incorrectly. The positions of ς_i are presented in the figure 15.

$$\varsigma_i(w,b) = \begin{cases} 0, & y_i = +1\ and\ w\ .x_i + b \ge 1 \\ 1 - w\ .x_i + b, & y_i = +1\ and\ w\ .x_i + b < 1 \\ 0, & y_i = -1\ and\ w\ .x_i + b \le -1 \\ 1 + w\ .x_i + b & y_i = -1\ and\ w\ .x_i + b > -1 \end{cases}. \qquad (3.23)$$

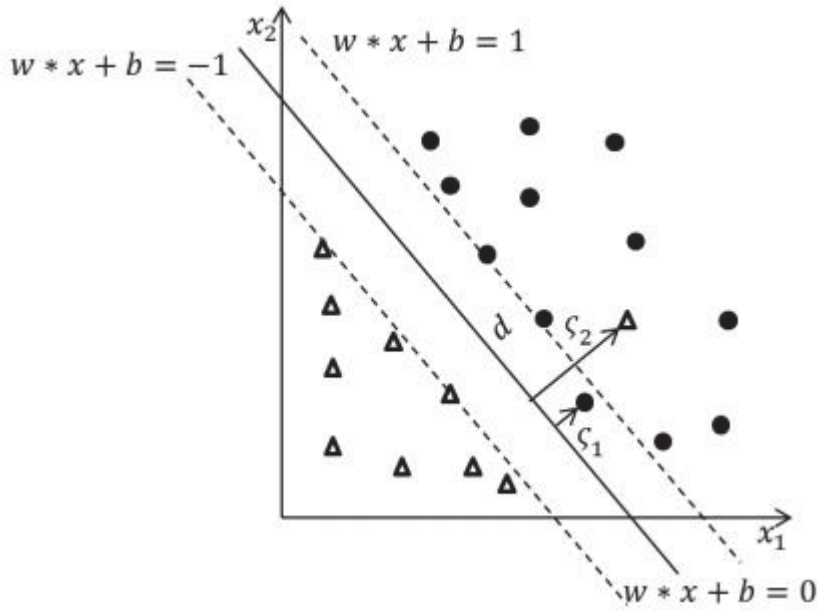


Figure 15 – Positions of the points in a soft margin SVM.
Source: Adaptation of Burges (1998) [39].

Shilton (2016) showed that both hard and soft margins SVMs have the same objectives, maximize the margin between the optimal hyperplane and the patterns

closer and to clear the classification errors. When is not possible to clear them, then is sought to minimize the errors and the patterns inside de margin. With this purpose, these values are mixed in one equation to after minimize (3.24), where C is a constant that defines the relaxing weight [40].

$$\varepsilon(w,b) = \|w\|^2 + C\sum_{i=1}^{n}\varsigma_i(w,b)\ . \tag{3.24}$$

3.3. NONLINEAR SVM

Sometimes, the hard or soft margins can classify one data set. There are cases where the classes cannot be separated by a hyperplane in a satisfactory way. To deal with this problem, a mapping called characteristics space is made, transforming the data in linearly separable again, rising its dimensionality. Figure 16 provides an example of it.

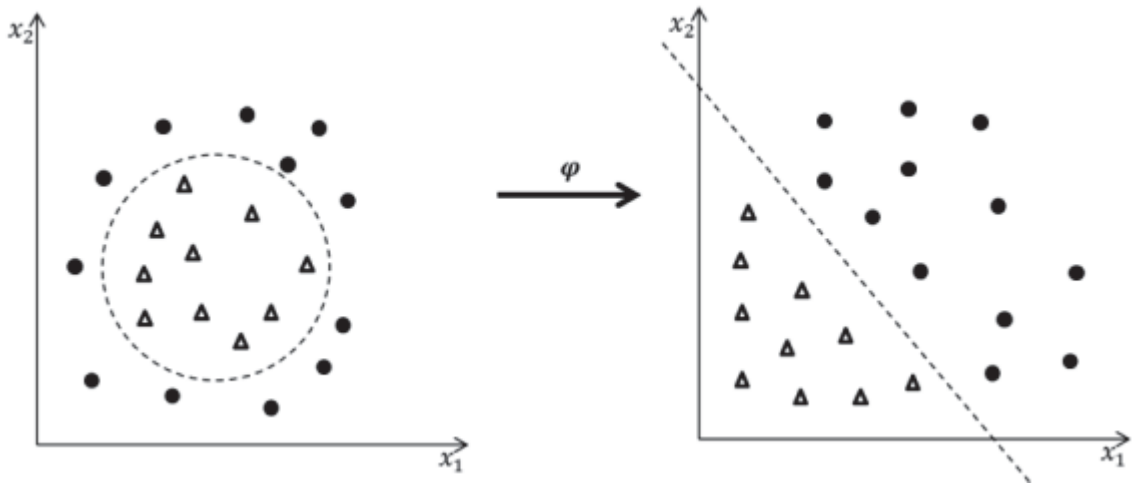


Figure 16 – Transformation of the data from a spatial space to the characteristics space.
Source: Adaptation of Burges (1998) [39].

The transformation occurs using one vector of nonlinear functions, $\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_l(x))$, which maps the previously seen *m*-dimensional entry vector x , to the *l*-dimensional characteristic space. The separation is made by the hyperplane shown in equation 3.25, where w is an *l*-dimensional vector and b is the compensating term.

$$w \,.\, \phi(x) + b = 0 \,. \tag{3.25}$$

Because of the probably high number of dimensions that a characteristic space could have, the internal product between two patterns in this space can be very computationally expensive, sometimes there is a large number of dimensions in the problem, making the transformation impossible. To deal with this problem, the method called *kernel trick* is used, where the patterns are not treated explicitly in the characteristics space.

This method is based on calculating the internal product between two patterns entries, x_1 and x_2, but without calculating its respective mappings in the characteristics space. Instead, a *kernel* function $K(x_1, x_2)$ is defined, which is equal to the two patterns internal product in some characteristic space $\phi_1(x) \,.\, \phi_2(x)$. The *kernel* is defined without calculating $\phi_1(x)$, $\phi_2(x)$ or even knowing which characteristic space $\phi(\,.\,)$ is.

Defining one kernel, the Mercer conditions must be satisfied. These condition ensures one specific kernel indeed represents $\phi_1(x) \,.\, \phi_2(x)$ in some characteristics space ϕ. Equation 3.26 presents the Mercer's conditions, where M is a natural number and h is a real one. In addition, the equation 3.27 presents the optimization dual problem.

$$\sum_{i,j=1}^{M} h_i h_j K(x_i, x_j) = \left(\sum_{j=1}^{M} h_j \phi^T(x_j)\right)\left(\sum_{j=1}^{M} h_j \, \phi(x_j)\right) \geq 0 \,, \tag{3.26}$$

$$Maximize \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i x_j) \,, \tag{3.27}$$

$$Subject\ to: \begin{cases} 0 \leq \alpha_1 \leq c, i = 1,2, \ldots, m \\ \sum_{j=1}^{n} \alpha_i y_i = 0 \end{cases} .$$

After obtaining the optimal value α, called α^*, in the equation , the w^* and the b^* can be obtained the same way as in the hard and soft margins method. With equation 3.28, a new pattern x_{new} can be classified.

$$C(x_{new}) = sign\left(\sum_{x_i \in SV} \alpha_i^* y_i K(x_i, x_{new}) + b^*\right) \,. \tag{3.28}$$

Any function that attends to the Mercer conditions can be called a kernel, but there are some that are used more frequently. They are the linear kernel, polynomial

kernel, radial basis function kernel and sigmoidal kernel. The linear is represented in the equation 3.29.

$$K(x_i, x_j) = x_i \,.\, x_j \,. \tag{3.29}$$

In the polynomial kernel, the mapping functions ϕ are also polynomials and its complexity rises as the grad d of the polynomial. The equation 3.30 shows the polynomial kernel.

$$K(x_i, x_j) = (\tau + x_i \,.\, x_j)^d \,. \tag{3.30}$$

The RBF kernel represents an infinite characteristics space dimension, where almost every mapping can be implemented by this function. A neural network of type RBF can be defined, the number of functions of the radial basis should be the same as the number of support vectors (SV) and their center equal to the Lagrangian multipliers of each SV. Equation 3.31 presents the radial basis function kernel.

$$K(x_i, x_j) = \exp(-\frac{\|x_i + x_j\|^2}{2\sigma^2}) \,. \tag{3.31}$$

The sigmoidal kernel can explicit a perceptron multi-layer neural network, being the number of SVs equals the neurons in the intermediated layer and the Lagrangian multipliers equal to the bias vectors. Equation 3.32 presents a regular sigmoidal kernel function.

$$K(x_i, x_j) = \tanh(\beta_0 x_i \;.\; x_j + \beta_1) \,. \tag{3.32}$$

The capability of one SVM to have a good generalization as much its complexity depends on the values of the kernels parameters as well the constant of relaxing c. Some techniques were developed to this parameters selection, like the *span* rule, the *Jaakola and Haussler* and the rule proposed by Chapelle (2002) [42].

4. DEVELOPMENT

The main goal of the project is to early predict the presence of a pedestrian in a traffic environment, only using one radar and signal processing techniques to extract the micro Doppler effect. This data was used after, with different methods of support vector machine to detect the pedestrians. Then the results of time and performance were compared.

This chapter presents the work developed in the year of 2017 in the CARISSMA laboratories, in the city of Ingolstadt, Germany. For doing that, the execution of this project was divided into four major parts: The first part was an introduction to the specific used radar and its parameters. The second one was the optimization of the radar parameters in order to get the best values for velocity range and angle. The third one was to definition of test scenarios and gather big data. The last part was to organize the data, training different types of support vector machine models to recognize the pedestrians' data inside the data gathered, and extract the results and time of performance of each one.

4.1. RADAR PARAMETERS

To get the data, a radar called radarlog [43] was used. The supplier (Inras GmbH) sent together libraries, some code and configurations of the RADAR. Along with the radar was used one computer using Matlab R2015a with USB 3.0 communication between Matlab and radarlog. For the software, two algorithms and two libraries from the supplier were used as basis to generate an unified algorithm capable to capture all the needed data [44, 45].

Five algorithms were provided with the radarlog by Inras. Three of them, libraries with functions to configure the radar as sets and gets, the other two are where the processing and calculus were performed. One gathered data of the objects angle and range and the other gathered the velocity and range.

This chapter is devoted to the software development, detail about the radar hardware structure can be found in the Appendix A.

4.1.1. Angle processing

This algorithm uses two-dimensional Fourier transform of the IF signals to get the data from the angle and range. It also uses all sixteen channels and a chirp frequency modulation continuous wave with just one chirp per frame.

The algorithm begins setting the power of the transmit antenna to its maximum, and then the parameters of the frequency modulation wave are set. Each frame has only one chirp, where the frequency goes from 76 to 78 GHz in 256 µs and then returns to the first frequency in 64 µs. This was made one time per frame, and the entire frame lasts 200 ms. The code gather 512 measures during one chirp, and it captures 100 frames. Figure 17 represents the frequency modulation and table 1 shows the values of the parameters.

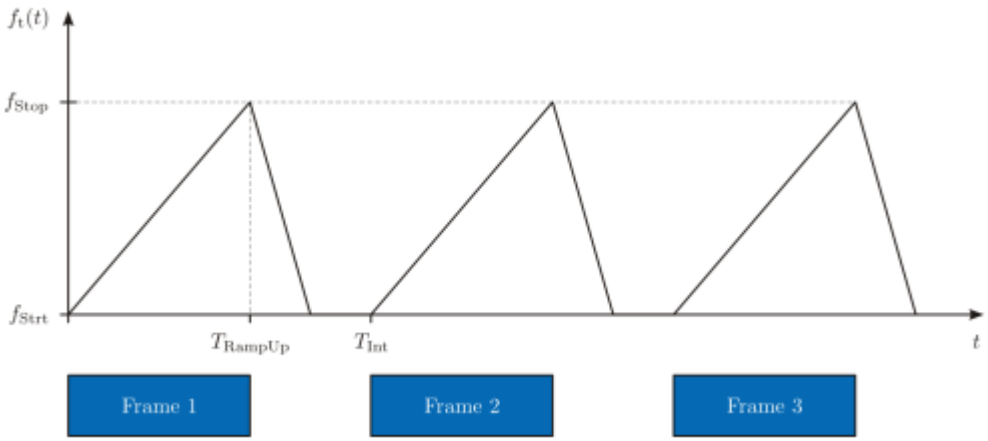


Figure 17 – Frequency modulation of algorithm ANGLE x RANGE.
Source: Inras GmbH (2017) [44].

Table 1 – Frequency modulation parameters ANGLE x RANGE.

Variables Values	
Start Frequency (*Fstart*)	76 GHz
Stop Frequency (*Fstop*)	78 GHz
Ramp up Time (*TRampUp*)	256 µs
Ramp down Time (*TRampDown*)	64 µs
Frame Time (*Tint*)	200 ms
Number of measurements (*N*)	512
Number of frames (*NrFrame*)	100

After the wave parametrization, the preprocessing begins. At first, using the equation (2.11) mentioned in chapter 2, the range is calculated. A variable called $NFFT$ must be defined with the desired size of data. Then, a vector with size equivalent to this variable is created with values equally spaced between 0 and 1. This vector is replaced in T in the equation (2.11). The result of the equation is a vector of values between 0 and the maximum possible range. This vector is limited for two variables defined by the user, R_{max} and R_{min}. Which defines the limits of the distance.

Next, the same was performed for the angle. Another variable called $NFFTAng$ is defined. A vector of size $NFFTAng$ between 0 and *180* was created. Meaning the representation of the possible degrees of the radar. This vector and the vector before will define the *x*- and *y*-axis of the resulting graph.

The code measures the data from the radar using a function called $BrdGetData$. The function returns the IF signals measured by the radar. The size of these signals is $(N * NrFrame * 16)$. To get the information of the position and angle two fast Fourier transform are applied, one in each direction of the signal matrix (column and row). The size of the first transform is $NFFT$ and the second $NFFTAng$. At last, a logarithmic function of basis 10 of this signal is multiplied per twenty in order to make the intensity of the signal in the dB scale.

The data, in dB, is used as a color map with the information and the vector of angle and the vector of range are used for scaling max in the axis x and y at the same graph. Figure 18 illustrates a graph generated from this algorithm.

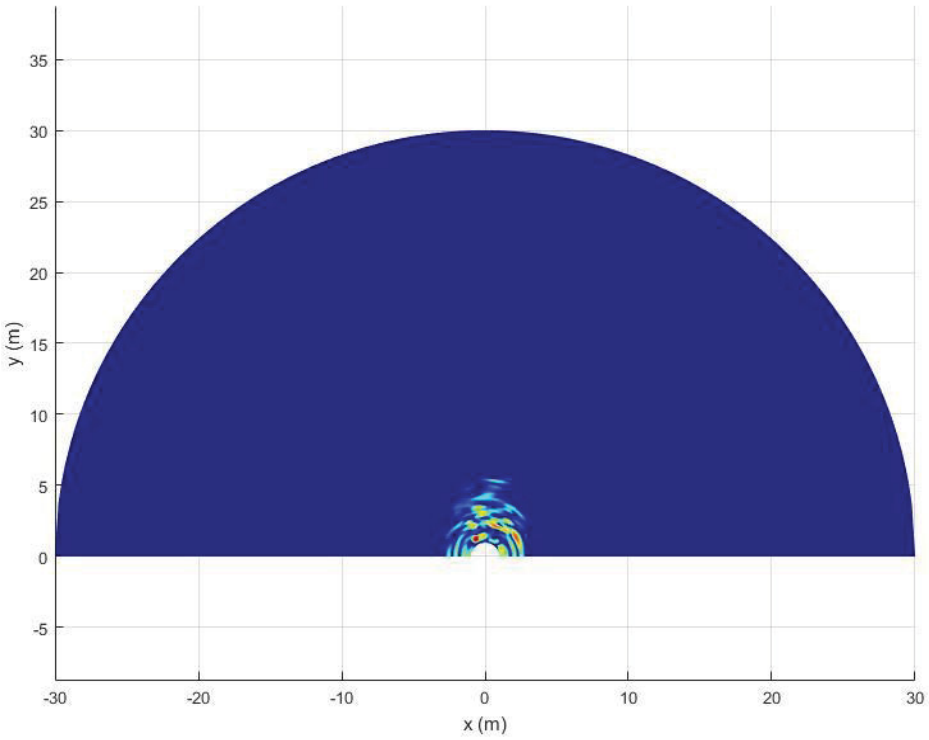


Figure 18 – Angle x Range, supplier algorithm
Source: The author (2017).

4.1.2. Velocity processing

To calculate the velocity with the radarlog the algorithm uses the two-dimensional Fourier transform of the IF signals. In this code, only two channels of the radar are used and a chirp frequency modulation continuous wave with multiple chirps generates the Range-Doppler map.

Firstly, the power of one antenna is set to the maximum. Then the frequency modulation is configure like the figure 19, called *chirp modulation*, where the frequency goes from 76 to 77 GHz in a time of 51.2 µs, and then goes back to 76 GHz in a time of 20 µs. The sum of these two times is the time for one chirp. The code repeat the chirp 64 times in each one of the 100 frames. During each chirp, the radar measure 512 values of the IF signals. All the variables from the wave parameters are presented in the table 2.

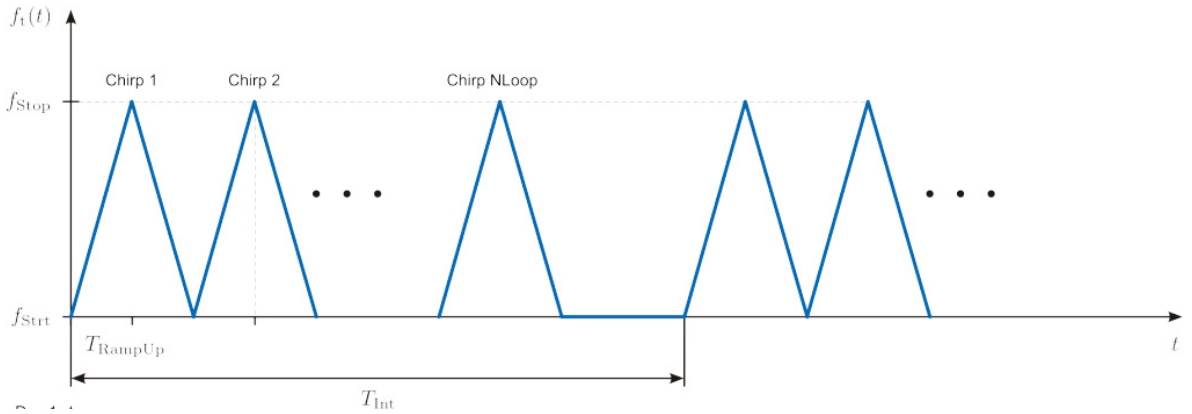


Figure 19 - Frequency modulation of algorithm VELOCITY x RANGE.
Source: Inras GmbH (2017) [45].

Table 2 – Frequency modulation parameters VELOCITY x RANGE.

Variables Values	
Start Frequency (*Fstart*)	76 GHz
Stop Frequency (*Fstop*)	77 GHz
Ramp up Time (*TRampUp*)	51.2 µs
Ramp down Time (*TRampDown*)	20 µs
Frame Time (*Tint*)	200 ms
Number of measurements (*N*)	512
Number of frames (*NrFrame*)	100
Number of Chirps (*NLoop*)	64

Secondly, a range vector is calculated using the variable $NFFT$ and the limitations R_{max} and R_{min}, exactly the same as explained in the subchapter 4.1.1.

After that, a velocity vector is created, using the equation 2.13, where $v_{r,rel}$ is the relative velocity of the object, $f_0$is the average frequency and f_D is the Doppler frequency. Again, for the velocity be a vector at least one variable must be a vector. The only one which is not constant is the Doppler frequency f_D. Therefore, another variable is created, $NFFTVel$. This variable generates a vector, of size $NFFTVel$, equally spaced between 0.5 and -0.5, because the velocity can be negative depending if the object approaches or departs the radar. This new vector multiplies the inverse of the time of one chirp ($TRampUp + TRampDown$), to correspond to the f_D and then multiplies the rest of the velocity equation, generating a velocity vector $v_{r,rel}$.

In the sequence, a function, which receives the IF signals of all sixteen channels from the radar, is called. The size of this data is $N * NLoop * NrFrame * 16$ per frame. The algorithm then divides the IF signals per frame, and uses only two channels

information. In the sequence, a simple normalization is made in this data and the result multiply a Hanning window. With this treated signal a fast Fourier transform with size $NFFT$ is performed in the first dimension of the data matrix, the results is limited by the maximum and minimum range set before. A new fast Fourier transform is applied again in the previous result, but this time with size $NFFTVel$ and in the second dimension of the data matrix. In the result is applied a logarithmic function of basis ten, in order to get the intensity in the dB scale.

The result show the intensity of the IF signals, the vector of velocity and range are used for scaling the axis, x and y, at the same graph. Figure 20 presents one frame of the velocity per range graph.

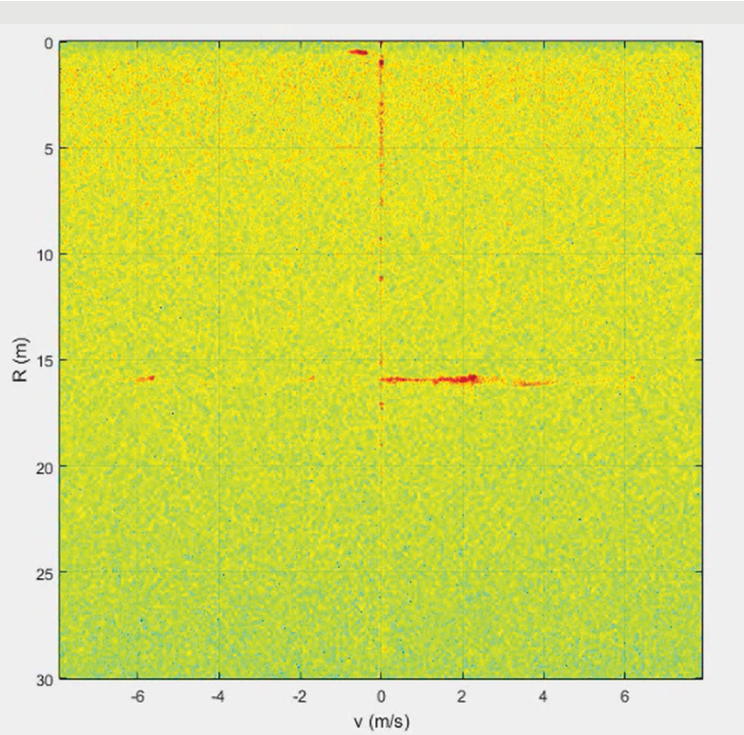


Figure 20 - Velocity x Range, supplier algorithm.
Source: The author (2017).

In the graph presented before, the red color represents objects, its velocities, and range. In the velocity zero there are a vertical line in red, the reason is that there are objects in different distances standing still, with no velocity. In the range, near fifteen meters there is movement, as it can be seen on the horizontal red line, and the velocity is positive, between zero and four m/s.

4.1.3. Velocity and angle unification

The radar is capable of measuring the velocity, the range and the angle of objects, and with that information more features can be presumed, like the micro Doppler effects with the velocity, and a pattern of movement with angle and range. All this features are important for the training and learning of the SVM method, with more features is easier to the model learn how to separate the pedestrian from other objects, also to predict the pedestrian trajectory. Because of that, one unified algorithm was made to gather these three information at the same time and feed the SVM with them.

To reach that goal, the unified algorithm was based on the velocity processing, because of its complexity compared with the previous one presented on the sub-chapter 4.1.2.

In order to adapt the angle information in the velocity algorithm, changes were made. In the angle processing algorithm the modulation used just one chirp per frame, while the velocity one used many. For the unified algorithm, one unique modulation was created. The solution was to use the first chirp in each frame for the angle calculation and the rest $(NLoop - 1)$ for the velocity, exactly like the figure 19, but with one less chirp for velocity.

The parameters values was maintained as the table 2, the only change was the number of chirps $Nloop$. Now this parameter became two, $NloopVel$ and $NloopAng$, the earlier is equal to 63 and the later always equal to one. Figures 21 and 22 shows the result of the unification.

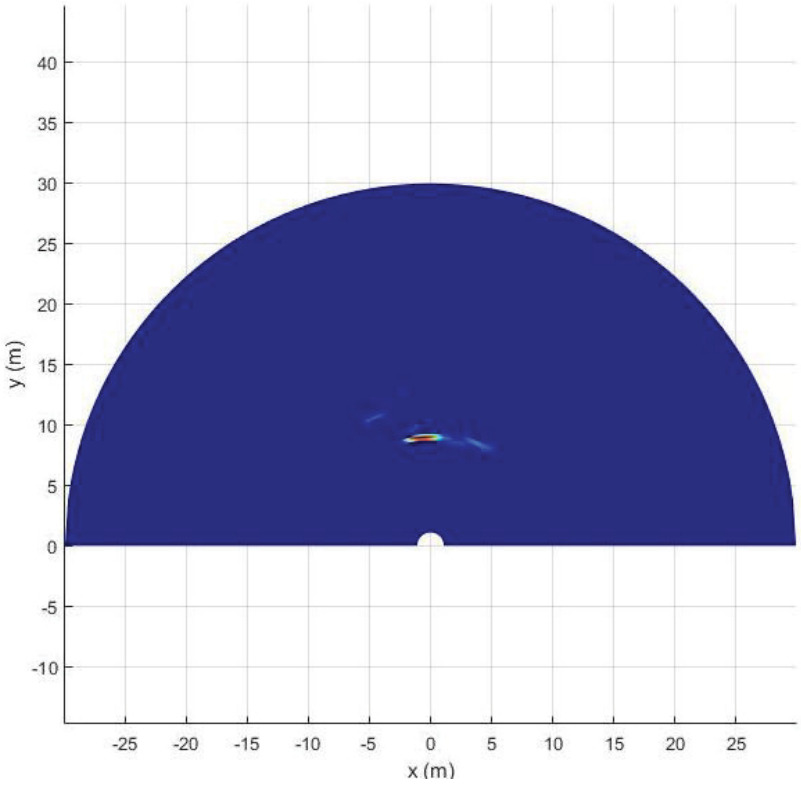


Figure 21– Angle x Range graph, unified algorithm.
Source: The author (2017)

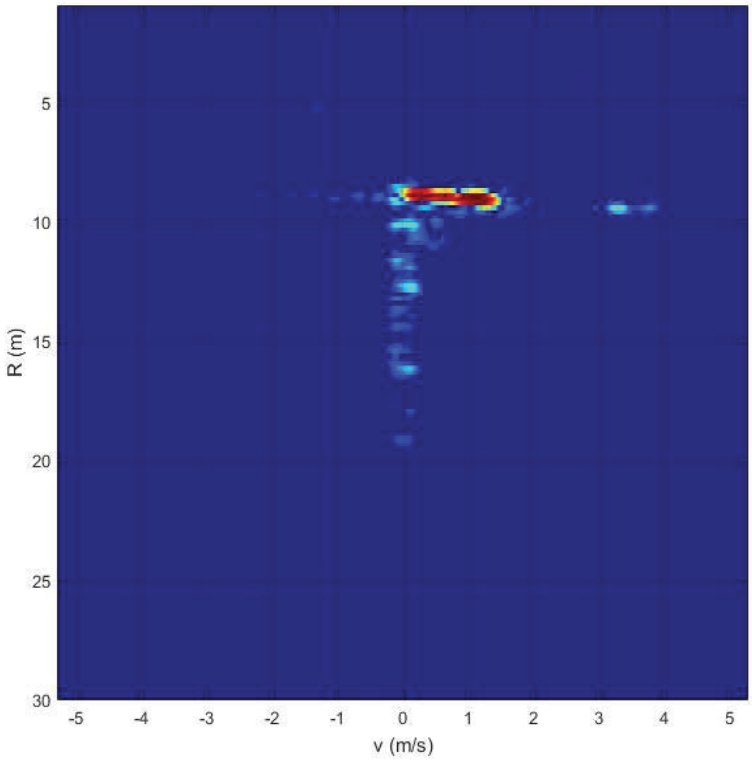


Figure 22 - Velocity x Range graph, unified algorithm.
Source: The author (2017).

In the figure 22 and 23 was the same test running. Is possible to see the person in the red area in almost ten meters in figure 22, and the maximum velocity is around two meters per second, in figure 23. More velocities are presented in the same graph, because of the many different velocities expected in a person movement.

4.2. OPTIMIZATION

The most important feature of the signal is the presence of micro Doppler intensities, which can define a person movement instead of movements of rigid bodies. Only with very accurate velocity resolution (Δv), these intensities can be distinguished from noise. The major problem when improving the velocity resolution is to not worsen the other characteristics of the radar, like the maximum detectable relative velocity ($v_{r,max}$), range resolution (Δr) and maximum detectable range (r_{max}).

To deal with this problem a technique called multiobjective optimization was chosen. The advantage of this technique is that the calculation is made just once, before the use of the RADAR, avoiding many calculations in each frame. It uses the RADAR equations found in the literature and a mathematical optimization to find the best wave parameters. After the optimization, the RADAR is configured with the found parameters, and them the tests are made.

4.2.1. Multi-objective problem definition (MOP)

The first step of a MOOD is to define the problem. To define the objective functions to be optimize, the variables which will be find and the constraints of the problem.

Firstly was defined the objective functions J_i, five functions based on the Radar literature [15] and presented in the chapter 2, were defined to the optimization. Equations 4.1 to 4.4 represents the objective functions of the problem.

$$J_1 = \min_{\Delta v \, \epsilon \, R^+} \Delta v \,, \qquad where \; \Delta v = \frac{c}{2 f_c . T_{RRI} . K} \,, \tag{4.1}$$

$$J_2 = \max_{v_{r,max,1} \in \mathbb{R}} v_{r,max,1}, \qquad where \; \left| v_{r,max} \right| = \frac{c}{2 T_{RRI}} . \frac{1}{2 f_c}, \tag{4.2}$$

$$J_3 = \max_{r_{max} \in \mathbb{R}^+} r_{max}, \qquad where\ \ r_{max} = \frac{c}{2B} T \frac{f_s}{2}, \tag{4.3}$$

$$J_4 = \min_{\Delta r\, \epsilon\, R^+} \Delta r, \qquad where\ \ \Delta r = \frac{c}{2B}. \tag{4.4}$$

Where J_1 and J_2 are measured in *m/s* and J_3 and J_4 are measured in *m*. The maximization equations were transformed in minimization problems, and then constraints R_j were defined. Equations 4.5 and 4.6 explains the problem constraints.

$$R_1 = T_{frame} = T_{RRI}.K < 10^{-1} \tag{4.5}$$

$$R_2 = D_{rate} = \frac{N.K.N_{chn}.16}{T_{frame}} < 15.10^8 \tag{4.6}$$

The time to measure a single frame is represented in the inequality R_1. For real safety automotive applications, this time should be around 20 ms, but because of the computer hardware used in this project only 100ms could be achieved. The second inequality represents the amount of data in each frame. The radar has a transmission limit of data per frame of 1,5Gb, when the data is bigger, the information between frames become lost and desynchronized.

In the equations and inequalities were only six variables to be optimized. The parameters c and f_s, were constants and the variable f_c varies depending on B. The variables are defined as x_k in equations 4.7 to 4.12.

$$x_1 = T_{rampup}, \tag{4.7}$$

$$x_2 = T_{RRI} = T_{rampup} + T_{rampdown}, \tag{4.8}$$

$$x_3 = \mathrm{B} = f_{stop} - f_{start}, \tag{4.9}$$

$$x_4 = \mathrm{K}, \tag{4.10}$$

$$x_5 = N_{Chn}, \tag{4.11}$$

$$x_6 = \mathrm{N}. \tag{4.12}$$

4.2.2. Multi-objective optimization process (MOO)

The second step in a MOOD problem is the optimization process. In this phase, some methods are apply to generate multiple optimal solutions to the objective functions. Two techniques were used to generate possible solutions for this problem, the random search method (RS) [46] and genetic algorithm (GA) [47]. The first generates many possible solutions randomly, and then a dominance filter finds the Pareto Front. While GA, is used to generate fewer possibilities, also randomly, but in every iteration they are transform by an evolutionary process, using mutation, crossing over, heredity and natural selection reaching the Pareto front. The parameters of the random search and the genetic algorithm are present in tables 3 and 4 respectively.

Table 3 - Random search parameters.

RS parameters	
Population	10000
Distribution	Uniform

Table 4 – Genetic algorithm parameters.

GA parameters	
Population	10000
Crossing over	0.8
Generations	1200
Migration	0.2

4.2.3. Multi-criteria decision making step (MCDM)

The last step of the MOOD problem is the decision making, when one possibility must be chosen of the many generated in the last step. In order to facilitate the choosing, two ranking method were used in this work. The technique for order of preference by similarity to ideal solution (TOPSIS) [48] and the preference ranking organization method for enrichment of evaluations (PROMETHEE) [49]. Both methods need the user knowledge of the problem to make a ranking. In the first, it is needed to set a positive ideal solution (PIS) and a negative ideal solution (NIS) for J_i, then the possible solution closer to PIS and farther from NIS is considered the best. In the PROMETHEE it is necessary to set ideal values for x_k, then the technique compare the ideal values with the possible solutions to find the best. Table 5 shows the best three possibilities for each technique.

Table 5 – Ranking optimization results.

	TOPSIS Ranking									
	J_i				x_k					
Position	r_{max}	$v_{r,max}$	Δv	Δr	T_{rampup}	B	T_{RRI}	K	N	Nr_{Chn}
1	33,91	7,56	0,12	0,13	1,01E-04	1,13E+09	1,30E-04	128	512	6
2	38,20	7,87	0,12	0,15	9,34E-05	1,01E+09	1,25E-04	128	512	5
3	32,84	9,14	0,14	0,13	6,41E-05	1,17E+09	1,07E-04	128	512	5
	PROMETHEE Ranking									
	J_i				x_k					
Position	r_{max}	$v_{r,max}$	Δv	Δr	T_{rampup}	B	T_{RRI}	K	N	Nr_{Chn}
1	33,91	7,56	0,12	0,13	1,01E-04	1,13E+09	1,30E-04	128	512	6
2	32,84	9,14	0,14	0,13	6,41E-05	1,17E+09	1,07E-04	128	512	5
3	31,15	7,36	0,03	0,12	1,00E-04	1,23E+09	1,34E-04	128	512	5

The ranking had some differences between them, but the first position were equal for both, the first positon was assumed as the best. In the table 6 are the new parameters. Figure 23 and 24 presents the graphs obtained from the optimization procedure.

Table 6 – Optimized parameters.

Optimized Algorithm Parameters	
Start Frequency (Hz)	75,87e+09
Stop Frequency (Hz)	77e+09
Time Ramp Up (s)	101e-06
Time Ramp Down (s)	28,5e-06
Time of one Frame (s)	100e-03
Number of measurements	512
Number of Chirps	128
Number of channels	6

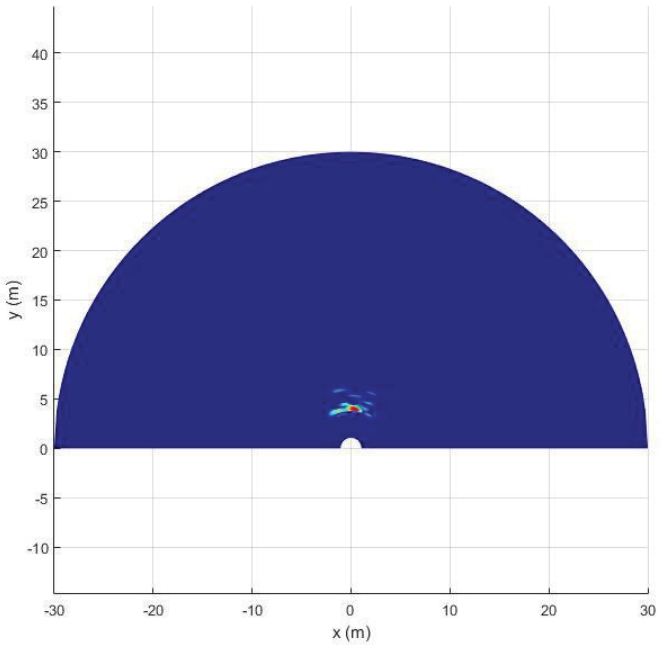


Figure 23 - Angle x Range graph, optimized algorithm.
Source: The author (2017).

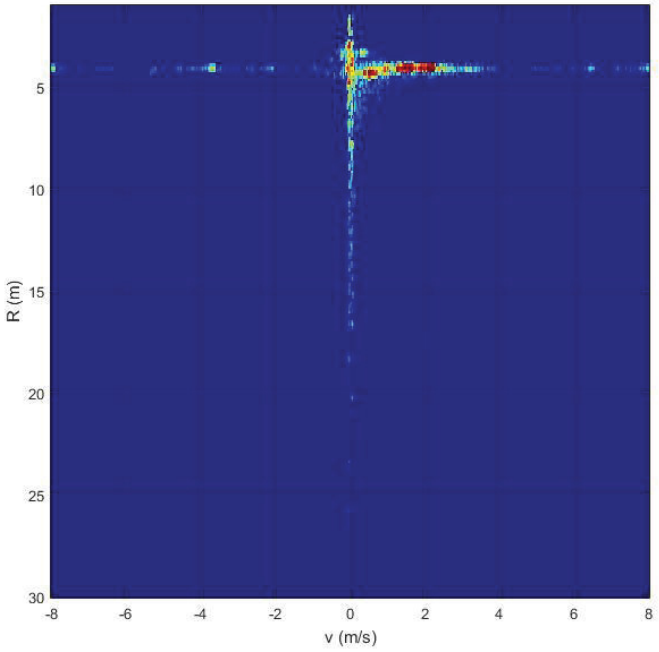


Figure 24 – Velocity x Range graph, optimized algorithm.
Source: The author (2017).

Again, there was a person in the red intensity near five meters in the figure 23, now the noise in the same graph was improved but the object was still clear. Figure 24 now presents clearer colors; the smaller intensities can be seen easier.

4.3. TEST PROCEDURES

In test procedures, the amount of data was too big for the computer memory capability, jeopardizing the duration time of each frame. Considering the range and velocity information enough for the training, the number of channels was reduced to only two and the angle information was not measured anymore.

The test procedures explained in the next sub-chapters was the first one of a bigger work. It has only one pedestrian crossing at each time, and has too few rigid objects in the street. In the sequence of the work developed in Germany more pedestrians, cars and bicycles will be used.

This subchapter is divided in two parts. In the first, is explained the six test scenarios and how they were simulated in the automotive track. In addition, the second one, the methodology of labelling data and the final data structure is presented.

4.3.1. Test scenarios and simulation

According to German In-Depth Accident Study (GIDAS) [50] more than eighty percent of the accidents involving pedestrians happens in six scenarios. The first one is when a car is moving forward and the pedestrian crosses the street in front of it. The second and third scenarios are the same situation as the first, but in these cases, the pedestrian is hidden behind an object. In the fourth, the pedestrian is standing in front of a moving car. In the last two, the car is turning, left or right, and the pedestrian is crossing the street. Figure 25 presents all most possible accident scenarios. In scenarios 2 and 3 there is an object blocking the drivers view of the pedestrian. Except these scenarios there is no other object, than a pedestrian, in the tests. More objects, like cars and bikers, will be consider in a future work.

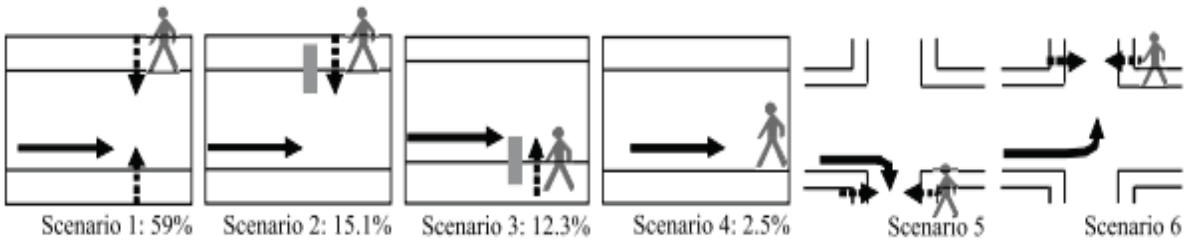


Figure 25 – Test scenarios.
Source: HEUEL, S.; ROHLING, H. (2011) [50].

These scenarios were created inside the automotive track, in CARISSMA lab. Unfortunately, there was no time for installing the radar in a real car for the test making, so a moving structure, where the radar and a digital camera were fixed, was assembled. The camera was only used for helping in the labelling step, there was no intention to use the camera information to train the classifier. Figure 26 illustrates the radar test structure.

Figure 26 – Radar test structure.
Source: The author (2017).

The track was divided in two lanes of 20 meter and the radar structure was positioned in the right lane. Two individuals represented the pedestrians in the simulations, using the same clothes, one male with 1,85m high and 0,54m wide, and other male 1,78m high and 0,60m wide. All the tests were made for the both crossing direction of the street.

In the first scenario, the structure was still, and for simulating the car forward movement, the crossing trajectory of the pedestrian was in diagonal. Three crossing diagonals with different angles were measure to simulate different velocities of the car, and for each one, the pedestrian crossed with different velocities: walking (~1,5m/s), slow running (~2,5m/s) and fast running (~4m/s). It was also defined three crossing starting points, to simulate the distance variation between the car and the crossing. Figure 27 presents a schematic of scenario 1.

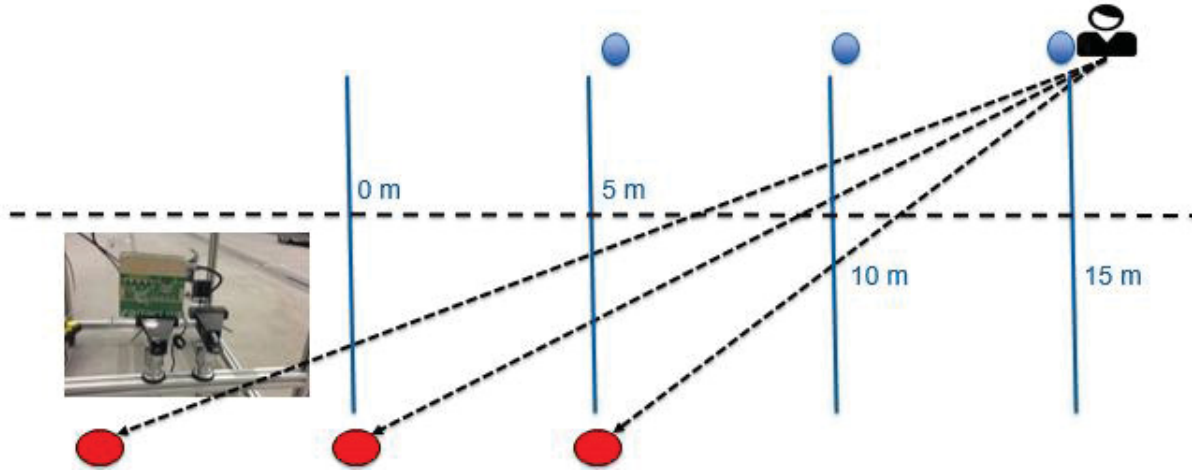


Figure 27 – First scenario representation.
Source: The author (2017).

In the second and third scenarios was repeated the methodology of the first one, however an object hid the pedestrian from the radar. A car prototype was used to do so. Figure 28 shows the car prototype used to hide the pedestrians.

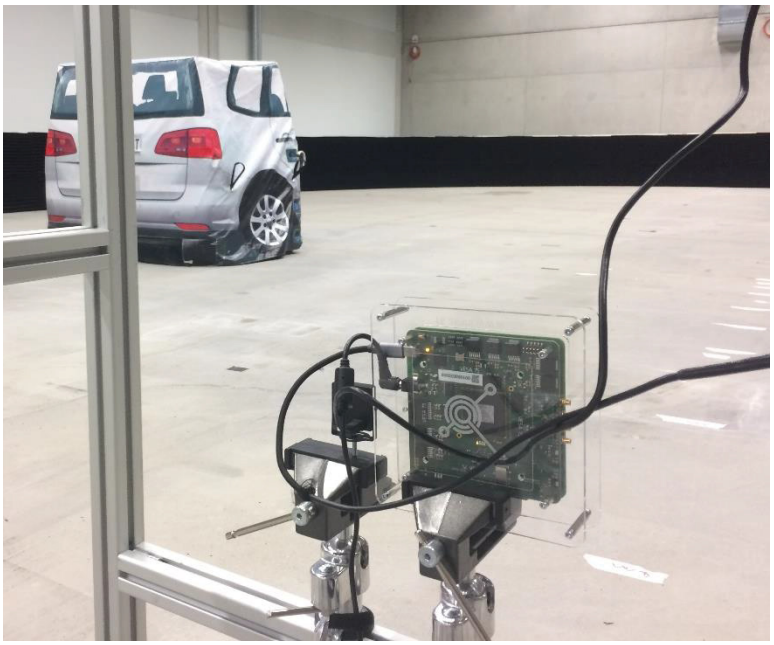


Figure 28 – Car prototype in the second scenario.
Source: The author (2017).

For the fourth scenario, the moving structure was pushed towards the pedestrian in one test, and then the pedestrian moved towards the structure with three different velocities, for the other three tests.

The last two scenarios the same methodology of the first was adopted, but the structure were positioned as if it was in a perpendicular street, and when the pedestrian had started moving the structure was pushed, simulating a curve.

In the first scenario 54 tests were captured, in the second one 27, the third also 27, the fourth only 4 and the fifth and the sixth 54 each, resulting in 220 tests gathered. Unfortunately, because of the project confidentiality the data is not available for independent verification.

4.3.2. Data labelling and search method

In the 220 test data, 16.160 frames were measured and saved, being 100*ms* the time of one frame. Considering the model to be trained as a supervised method, the data should be labelled before training. A small script was created, where the signal from the radar was plotted next to the images of the camera, this way if there was any doubts about which signal was the pedestrian, the digital images could help. Figure 29 represents the plot.

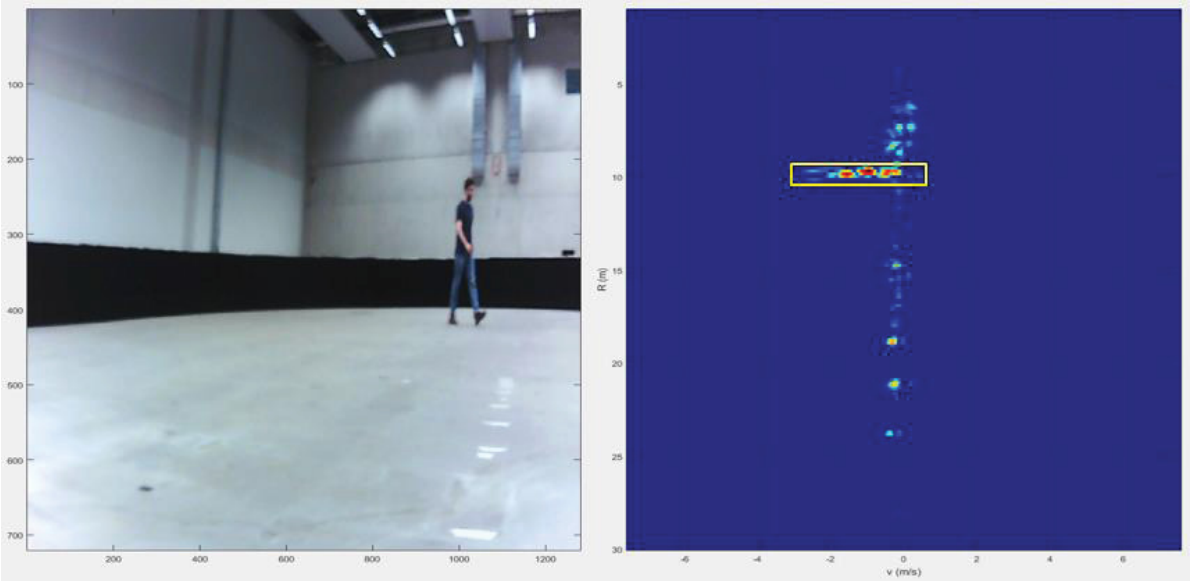


Figure 29 – Positive manual labelling.
Source: The author (2017).

In the script, for each frame, a searching method was responsible to found objects that may be a pedestrian. This method was necessary to reduce the processing time of the classifier. In the subchapter 4.4.3 has an explanation of the purpose of the searching method.

In this previous mentioned method, two techniques were used: The first one was the matrix connected neighbor labelling [51], and the second one the k-means clustering [52]. First, the frame matrix was transformed to logical values, being the values greater than -125 dB transformed to 1 and the smaller transformed to 0. The ones are the interesting objects, since intensities smaller than -125 dB are considered noise.

Next, the matrix connected neighbor labelling using eight neighbors clustered the connected ones into small groups. All these single groups were considered objects of interest. Figure 30 represents the smaller groups.

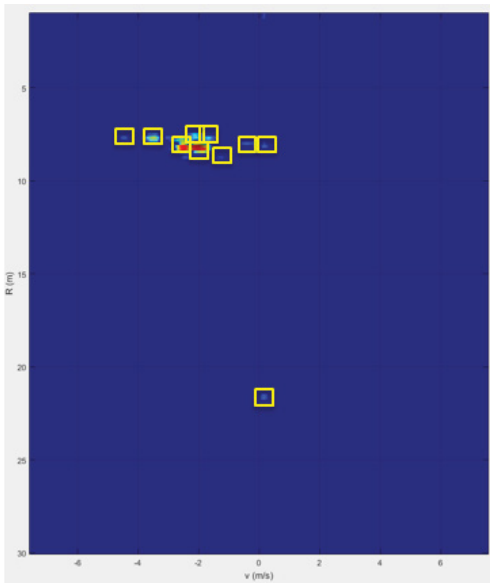


Figure 30 – Example of detection of smaller groups.
Source: The author (2017).

A pedestrian signal in the data frame could be a combination of many points disconnected, depending on the speed, because of the different velocities of legs and arms. For this reason, a k-mean clustering algorithm was used to group the smaller groups in bigger ones. The number of clusters was determined depending on the number of smaller groups. For example: If there were 10 smaller groups, would be 10 minus 1 groups to be clustered. For each one of these groups the value of *k* would change from 2 until 10. All the clustered groups are consider objects of interest. Figure 31 illustrates an example of clustering.

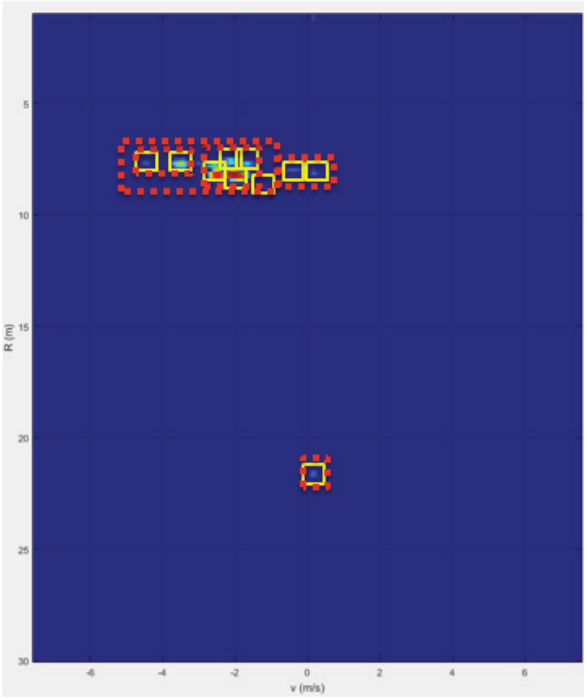


Figure 31 – Example of the smaller groups being clustered.
Source: The author (2017).

In the script, each one of the rectangles appears one at time to the user, and if the user pressed the computer space bar the signal inside the rectangle was saved as a pedestrian. If not, the signal was saved as non-pedestrian. In image 29, it is possible to see an example of a pedestrian rectangle while in figure 32 all the rectangles are non-pedestrians.

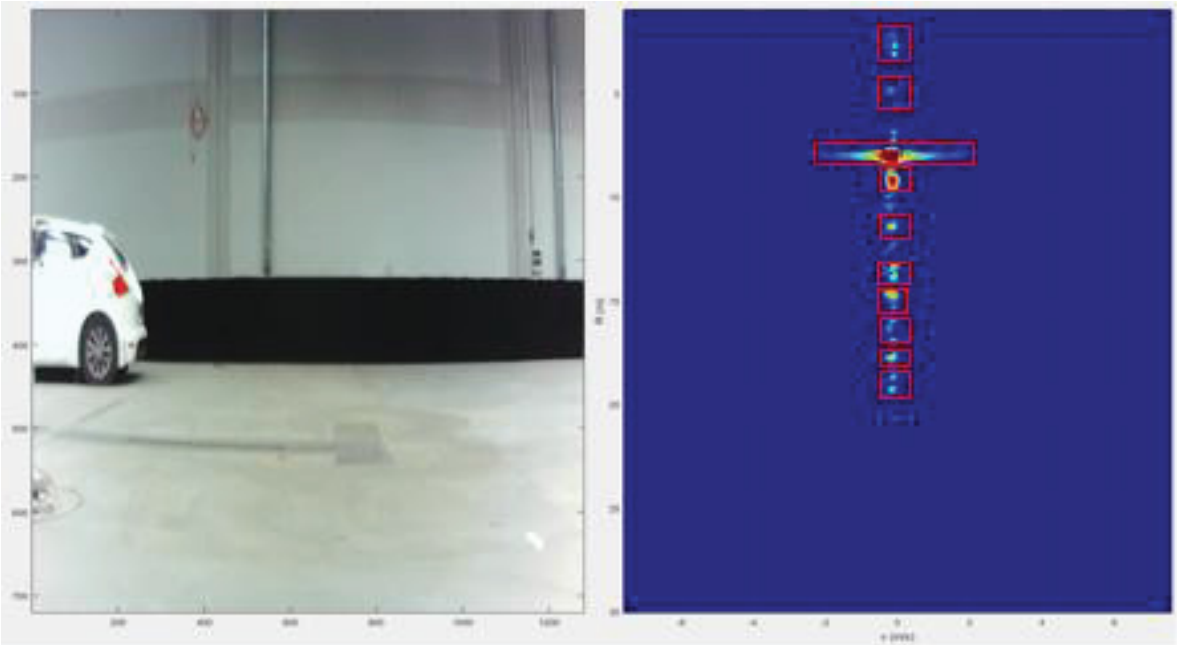


Figure 32 – Negative manual labelling.
Source: The author (2017).

In general, the negative instance were most commonly found than the positive, most because of the noise in some tests. The number of positive instances was 4.832, and the negative 246.990. It was approximately 30 Gb of data.

The amount of negative data was reduced randomly by a quarter of the total. It was necessary because the computer memory and processing were not capable of dealing with a data of this size in a reasonable time. In the end, 61.748 negative instances were used (quarter of the total). Adding the negatives and positives 66.580 instances were gathered. The data is unbalanced between negatives and positives instances. Being more than 92% negative. The sequence of the work used these data without any corrections, but for the sequence of the work the quantity of positive and negative data should be balanced.

For each rectangle, in the script, the velocity and distance was calculated. First, the area of maximum intensity of the signal was found, and then the center of this area. The corresponding value in the x-axis was the average velocity, and in the y-axis was the distance. These two parameters were used to train the SVM model. Figure 33 presents 19 lines of the data matrix after labelling.

	1 rect				2 Image	3 Velocity	4 Distance
1	-1.2751	15.1390	1.2626	0.5148	5x82 double	-1.1188	15.2977
2	-1.8860	14.8816	1.7513	0.7721	6x113 double	-0.5891	15.0403
3	-1.4685	14.6564	0.6924	0.8043	6x45 double	-1.0154	14.9739
4	-1.6315	14.4312	1.3135	0.8043	6x85 double	-1.2408	14.7487
5	-1.7842	14.4312	1.6088	0.6435	5x104 double	-1.0342	14.5899
6	-1.2344	14.0773	0.6720	0.8043	6x44 double	-0.9531	14.2360
7	-1.9980	13.7877	1.7310	0.8043	6x112 double	-1.0918	13.9465
8	-1.9369	13.7877	1.7310	0.6756	5x112 double	-0.7963	13.9465
9	-2.7719	13.6591	2.5048	0.6435	5x161 double	-1.0219	13.8178
10	-1.8046	13.3373	1.7412	0.7078	5x112 double	-1.2108	13.6548
11	-1.8555	13.2086	1.8226	0.8043	6x118 double	-1.4805	13.5261
12	-1.3565	12.9191	1.0488	0.7078	6x68 double	-0.9815	13.0778
13	-1.8351	12.7904	1.6291	0.6756	5x105 double	-1.0851	12.9491
14	-1.7740	12.6617	1.7921	0.6435	5x116 double	-0.5240	12.8204
15	-1.2344	12.4043	1.2422	0.7721	6x81 double	-0.7344	12.5631
16	-1.4787	12.4043	1.4561	0.4504	4x95 double	-0.8537	12.5631
17	-1.8656	12.2756	1.8837	0.4826	4x121 double	-1.7875	12.4344
18	-1.4074	11.8574	1.0182	0.8043	6x66 double	-0.7199	12.0161
19	-1.8555	11.7609	1.5171	0.6756	5x98 double	-1.1055	11.9196

Figure 33 – Data matrix after labeling.
Source: The author (2017).

4.4. SUPPORT VECTOR MACHINE TRAINING AND TESTING

Considering the training and testing of the SVM model, the data still needed some transformations. In addition, the training methods should be created. And after the training, it is needed a method to search as fast as possible in the data matrix for objects, which can be pedestrians.

In the three next subchapters is presented the data configuration, the SVM training and the used search method.

4.4.1. Data configuration

For the data configuration, the “rect” values, the coordinates of the labelling rectangle, were erased. This data were not important for training because they are better represented in velocity and distance values. Second, one more column called instances was add, with ones for the positive instances (pedestrians), and zero for the negative ones. After, the lines of the matrix were shuffled, to mixture the last column ones and zeros.

In the sequence was made the feature extraction with the matrices in the column “image”, using the video temporal gradient [25]. The last task was to normalize the size of the feature extraction resulting matrix. To use the SVM methods the signal should be the same size, but in the frames, when velocity changed, the size of the objects varied. To fix the problem the biggest size matrix was chosen as the ideal one, then all the others signal matrix were transformed, without losing its pattern, to this size, *15 x 541*. Finally, they were transformed to line vectors of size *1 x 8115*. Figure 34 presents 19 lines of the final data matrix.

	1 Velocity	2 Distance	3 Image	4 Instances
1	-1.1188	15.2977	*1x8115 double*	1
2	-0.5891	15.0403	*1x8115 double*	1
3	-1.0154	14.9739	*1x8115 double*	0
4	-1.2408	14.7487	*1x8115 double*	1
5	-1.0342	14.5899	*1x8115 double*	1
6	-0.9531	14.2360	*1x8115 double*	0
7	-1.0918	13.9465	*1x8115 double*	0
8	-0.7963	13.9465	*1x8115 double*	1
9	-1.0219	13.8178	*1x8115 double*	1
10	-1.2108	13.6548	*1x8115 double*	1
11	-1.4805	13.5261	*1x8115 double*	0
12	-0.9815	13.0778	*1x8115 double*	1
13	-1.0851	12.9491	*1x8115 double*	1
14	-0.5240	12.8204	*1x8115 double*	0
15	-0.7344	12.5631	*1x8115 double*	1
16	-0.8537	12.5631	*1x8115 double*	0
17	-1.7875	12.4344	*1x8115 double*	0
18	-0.7199	12.0161	*1x8115 double*	1
19	-1.1055	11.9196	*1x8115 double*	1

Figure 34 – Final data matrix.
Source: The author (2017).

After gathering the final data, four different SVM models were trained, each one organizing the data inside the column signal in different forms. For all methods the variables *velocity*, *distance* and *signal* were the inputs of the machine learning method, and the variable *instances* was the output.

It was used the *hold out* and the *cross validation* concepts for training. The data is divided randomly in five groups: *s1, s2, s3, s4* and *s5*. One of these groups is selected each time to be the testing group, while all the others become the training

group. So each time 25% of the data is used for testing and 75% for training. This is called *cross validation*, and in figure 35 is presented a representation of that.

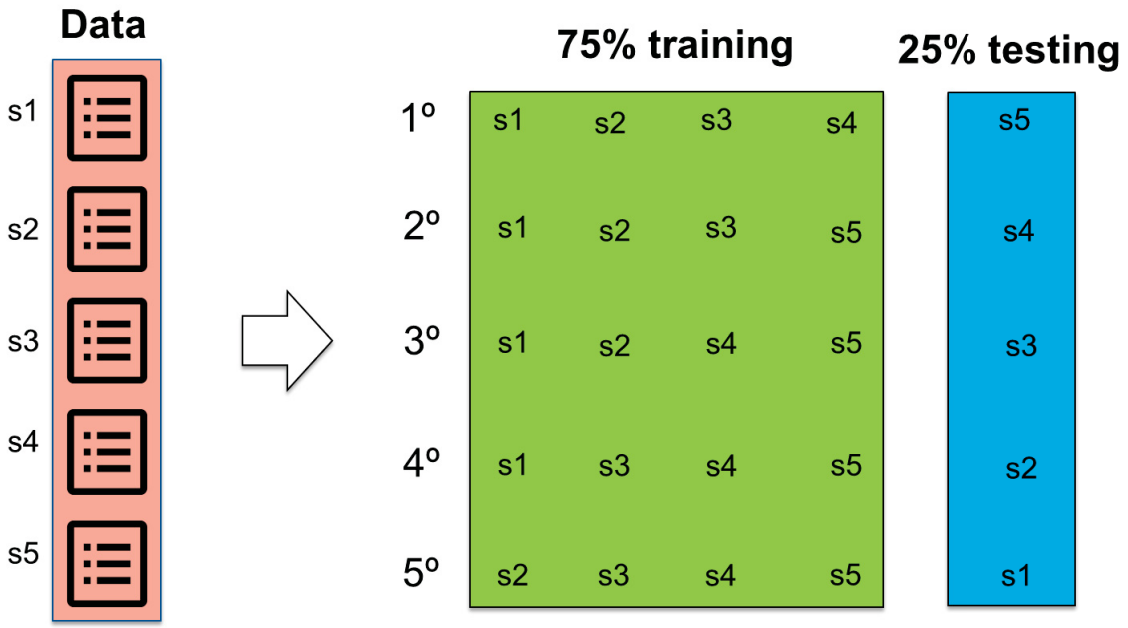


Figure 35 – Cross validation.
Source: The author (2017).

The supervised training is made with the inputs and outputs of the training group. This process generates a model, and with the inputs of the test group, a prediction is made. The result of the prediction is a column vector of same size as the group test output, and finally both prediction and test output are compared to calculate the model accuracy. Figure 36 illustrates a schematic of the *hold out* concept.

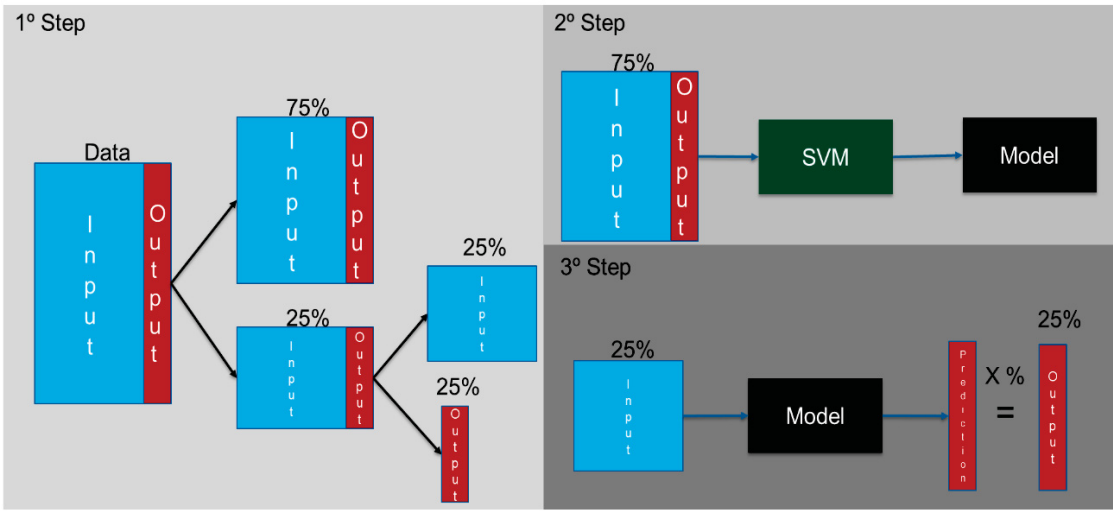


Figure 36 – Hold out concept.
Source: The author (2017).

4.4.2. SVM methods

As said in the sub chapter before, four SVM were trained, for each of them were adopted three different Kernel functions, mentioned in chapter 3. They were the linear, polynomial and RBF kernel functions. The kernel parameters used were the Matlab default. With the SVM cost defined as 1, the polynomial order as 3 and the RBF gamma as 1. In addition, each SVM method was trained and tested 15 times, and the result are the average of the result of the 15 times.

The first SVM method was trained with the original signal from the data configuration. The second SVM, used the same data but normalized between 0 and 1.

For the third and fourth SVM methods, a deeper analysis was made. When labelling the data, it was noticed that the pattern of the signal changes when the velocity and/or the distance changes. Figure 37 compares the signal in velocity and distance.

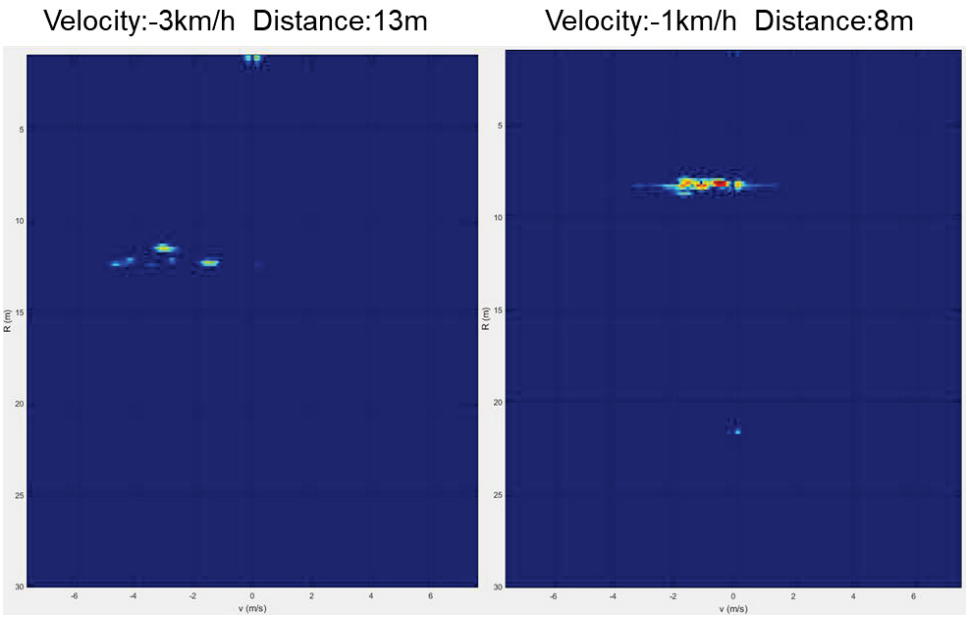


Figure 37 – Comparison of the signal in velocity and distance.
Source: The author (2017).

For velocity, the many micro Doppler effects become more scattered when the object is faster. As for distance, the signal intensity decreases losing micro intensities when the object is farer. Because of these facts, for the third and fourth models, the data were divided in three *t1*, *t2* and *t3*, before the *hold out* and *cross validation.* With each one, a different model was trained, which resulted in three predictions. At last,

the predictions were compared to the respectively test output and all the results were merged into one. Figure 38 presents an example of each one.

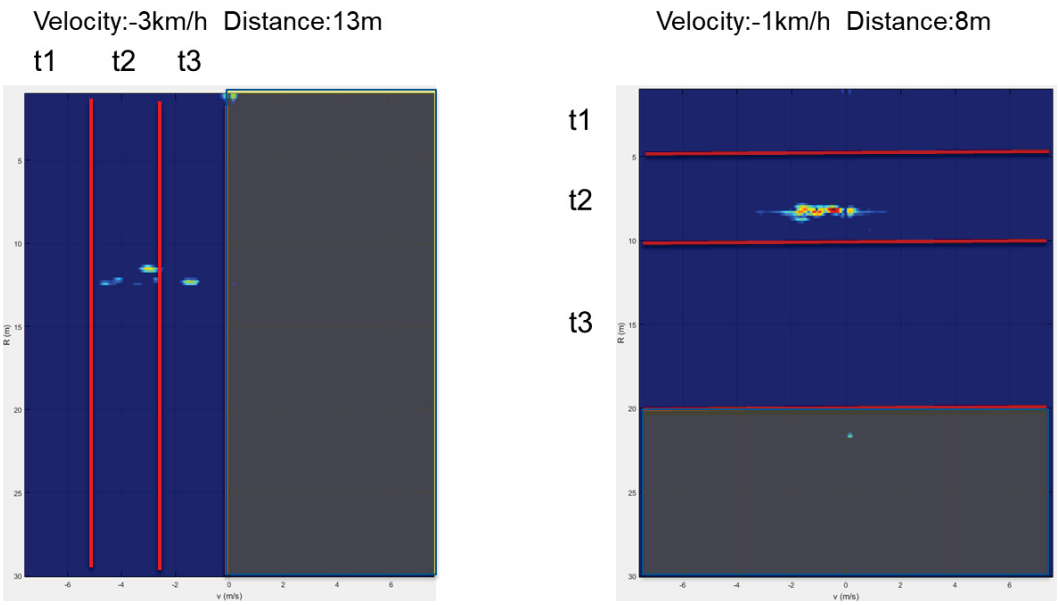


Figure 38 – SVM methods a) Velocity division; b) Distance division.
Source: The Author (2017).

4.4.3. Search method

After training and testing the SVM classifiers, it is needed to test them in real time. However, for doing that the computer should know which peace of the signal is an object.

The problem was that the data was labelled manually before training, this means that a human decided which data was going to be a pedestrian and which is not. Now the computer should do this choice alone.

The frame signal has a size of *1024 x 849,* and the smaller object found in the labelling had a size of *55 x 20*. Therefore, dividing the entire signal matrix into smaller parts would create more than seven hundred matrices to be tested each frame, without considering matrices overlapping. This way, would cost much time to execute each frame. Figure 39 shows an example of matrix division with overlap.

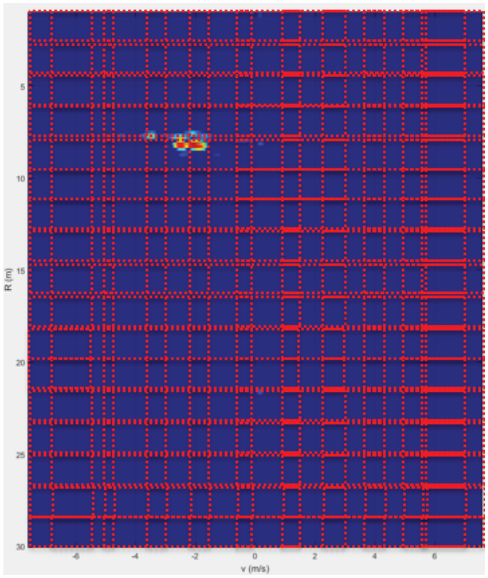


Figure 39 – Example of matrix division.
Source: The author (2017).

To deal with the problem the searching method explained in the subchapter 4.3.2 was used. As a result, instead of testing more than seven hundred matrices into the SVM classifier less than 120 possibilities was tested. This way the classification time was reduced.

Every rectangle found by the searching method is considered an object of interest. For all the objects of interest the velocity and the distance were calculated, the same way they were in the manually labelling (subchapter 4.3.2). The data were then merged in a matrix with velocity, distance and signal information. The signal information was converted to a matrix of size *15 x 541* and then to a line vector, as mentioned in sub chapter 4.4.1. At last, each one of these lines was tested into the classifiers.

5. RESULTS

The configuration of the radar's wave modulation was based on the multiobjective optimization results. As commented in sub-chapter 4.2, the main objective was to improve the velocity resolution in order to extract the micro Doppler effects of the human body. Table 7 compares the variables and objective functions, defined in 4.2.1, of the original algorithm and the optimized one. Next, figure 40 *a* and *b* presents two graphs with pedestrians trajectories. Both show how the velocity measurement of the pedestrian in each instant was; 40a was before optimization and 40b after.

Table 7 – Comparison between optimized and the original configurations.

	Objective Functions				**Variables**					
Config.	r_{max} (m)	$v_{r,max}$ (m/s)	Δv (m/s)	Δr (m)	T_{RampUp} (s)	B (Hz)	T (s)	K	N	N_{chn}
Optm.	33.9	7.56	0.12	0.13	1.01E-04	1.13E+09	1.2E-04	128	512	6
Orig.	33.4	13.77	0.43	0.15	0.51E-04	1.00E+09	0.7E-04	64	512	16

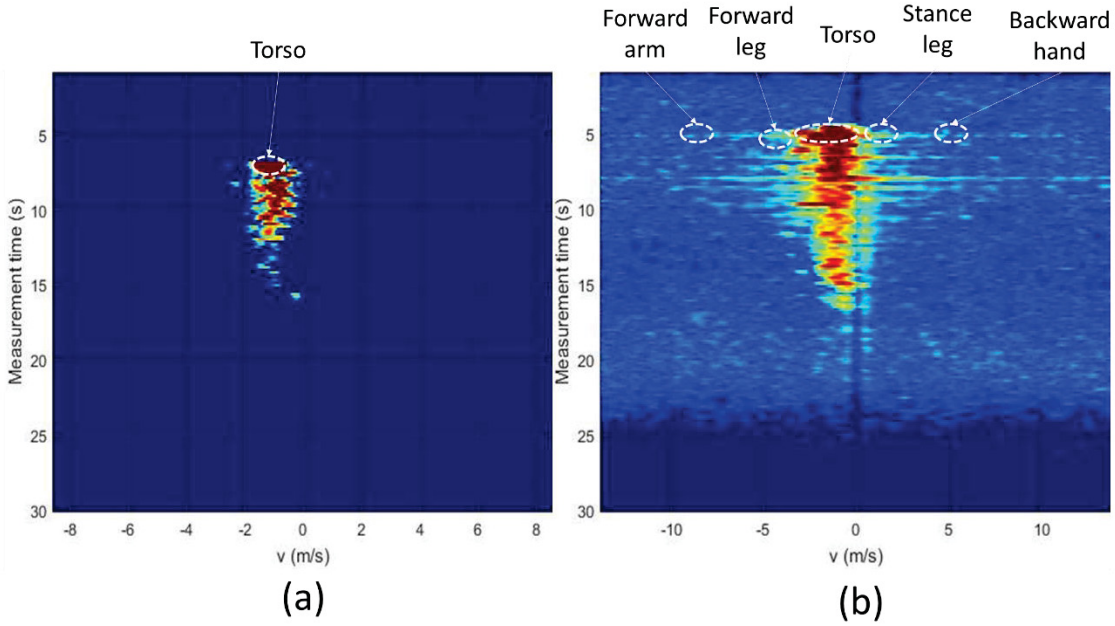


Figure 40 – Velocity measurement in each time for a pedestrian: a) before optimization; b) after optimization.
Source: The author (2017).

With the results of the multiobjective optimization, the data tests were gathered, and then, four support vector machines models were trained, as commented in sub-

chapter 4.4.2. Before showing the results of each model, a brief explanation of the performance and time indicators.

First, the confusion matrix, which is the most important indicator of performance in machine learning. With this matrix is possible to see how many positive and negative values were used in tests, how many positives and negatives values were predict correctly, called true positives (TP) and true negatives (TN), respectively. In addition, how many positives and negative instances were predict wrongly, called false negatives (FN) and false positives (FP), respectively. Figure 41 illustrates an example of confusion matrix.

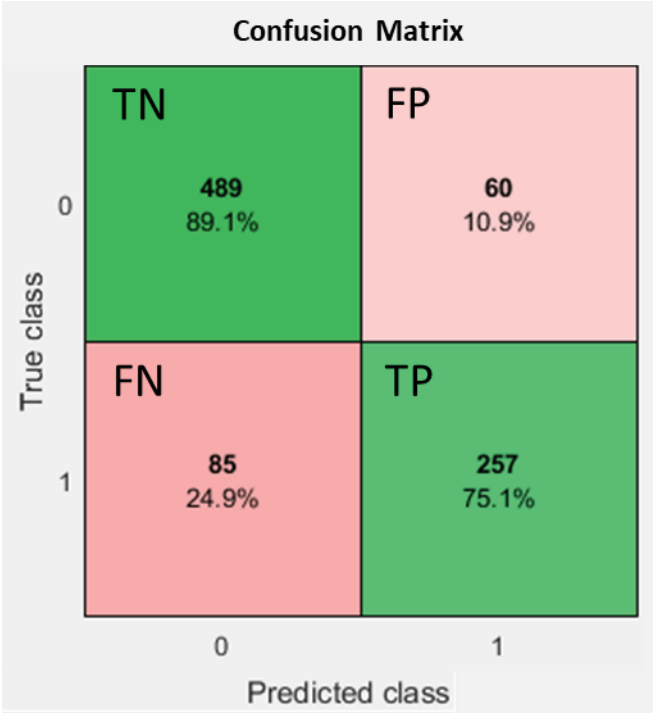


Figure 41 – Example of confusion matrix.
Source: Author (2017).

With the values of this table is possible to calculate the others indicators, like the confusability ($conf$), accuracy (acc), sensibility ($sens$), and error (err). In equations 5.1 to 5.4 the indicators respectively.

$$conf = (TP * TN) - (FP * FN), \tag{5.1}$$

$$acc = \frac{TN + TP}{TN + TP + FN + FP}, \tag{5.2}$$

$$sens = \frac{\mathrm{TP}}{TP + FP}, \tag{5.3}$$

$$err = \frac{\mathrm{FN} + \mathrm{FP}}{TN + TP + FP + FN}. \tag{5.4}$$

Considering time comparison, three times were measured. The training time, which is the duration of the training of each model. The searching time, which was the duration of the search method described in sub-chapter 4.4.3. In addition, the SVM execution time, which was the average time, which the support vector machine takes to predict an entire frame.

The tables 8 to 11 show the results of each individual methods comparing the different Kernel functions. For last, table 12 compares the best of each model.

Table 8 – Kernels functions comparison for the first training model.

	First model		
	Linear	**Polynomial**	**RBF**
Number sets	15	15	15
Total tests	19974	19974	19974
Negative instances	1487	1487	1487
Positive instances	18487	18487	18487
False Negative (FN)	1,16%	0,17%	0,44%
False Positive (FP)	1,91%	0,31%	0,54%
True Negative (TN)	5,54%	7,13%	6,91%
True Positive (TP)	91,40%	92,39%	92,12%
Accuracy	96,9%	99,5%	99,0%
Confusability	20103125	26294904	25382691
Error	3,10%	0,50%	1,00%
Sensibility	98,0%	99,7%	99,4%
Training time (s)	1099,76	657.986	920
Searching time (s)	0,1166	0,0253	0,0242
SVM execution time (s)	0,4913	0,0476	0,0714

Table 9 – Kernels functions comparison for the second training model

	Second model		
	Linear	**Polynomial**	**RBF**
Number sets	15	15	15
Total tests	19974	19974	19974
Negative instances	1487	1487	1487
Positive instances	18487	18487	18487
False Negative (FN)	1,45%	0,74%	0,81%
False Positive (FP)	1,84%	0,47%	0,56%
True Negative (TN)	5,60%	6,98%	6,89%
True Positive (TP)	91,11%	91,82%	91,75%
Accuracy	96,7%	98,8%	98,6%
Confusability	20257210	25552289	25198705
Error	3,3%	1,2%	1,4%
Sensibility	98,0%	99,5%	99,4%
Training time (s)	1215,2	796,56	1038,56
Searching time (s)	0,0253	0,0264	0,0264
SVM execution time (s)	0,1054	0,068	0,09265

Table 10 – Kernels functions comparison for the third training model

	Third model		
	Linear	**Polynomial**	**RBF**
Number sets	15	15	15
Total tests	19974	19974	19974
Negative instances	1487	1487	1487
Positive instances	18487	18487	18487
False Negative (FN)	1,22%	0,19%	0,27%
False Positive (FP)	1,66%	0,43%	0,45%
True Negative (TN)	5,79%	7,01%	7,00%
True Positive (TP)	91,34%	92,37%	92,29%
Accuracy	97,1%	99,4%	99,3%
Confusability	21009631	25843781	25766015
Error	2,9%	0,6%	0,7%
Sensibility	98,2%	99,5%	99,5%
Training time (s)	856,632	513.083	713,928
Searching time (s)	0,0341	0,033	0,0319
SVM execution time (s)	0,1054	0,068	0,0391

Table 11 – Kernels functions comparison for the fourth training model

	Fourth model		
	Linear	**Polynomial**	**RBF**
Number sets	15	15	15
Total tests	19974	19974	19974
Negative instances	1487	1487	1487
Positive instances	18487	18487	18487
False Negative (FN)	1,46%	0,31%	0,39%
False Positive (FP)	1,11%	0,21%	0,30%
True Negative (TN)	6,34%	7,24%	7,15%
True Positive (TP)	91,10%	92,24%	92,16%
Accuracy	97,4%	99,5%	99,3%
Confusability	22971825	26640008	26283450
Error	2,6%	0,5%	0,7%
Sensibility	98,8%	99,8%	99,7%
Training time (s)	576,4	352	434,8
Searching time (s)	0,0341	0,0319	0,0319
SVM execution time (s)	0,1071	0,0663	0,04165

Table 12 – Comparison between the best results in each training model.

	First model	**Second model**	**Third model**	**Fourth model**
	Polynomial	**Polynomial**	**Polynomial**	**Polynomial**
Number sets	15	15	15	15
Total tests	19974	19974	19974	19974
Negative instances	1487	1487	1487	1487
Positive instances	18487	18487	18487	18487
False Negative (FN)	0,17%	0,74%	0,19%	0,31%
False Positive (FP)	0,31%	0,47%	0,43%	0,21%
True Negative (TN)	7,13%	6,98%	7,01%	7,24%
True Positive (TP)	92,39%	91,82%	92,37%	92,24%
Accuracy	99,50%	98,80%	99,40%	99,50%
Confusability	26294904	25552289	25843781	26640008
Error	0,50%	1,20%	0,60%	0,50%
Sensibility	99,70%	99,50%	99,50%	99,80%
Training time (s)	657.986	796,56	513.083	352
Searching time (s)	0,0253	0,0264	0,033	0,0319
SVM execution time (s)	0,0476	0,068	0,068	0,0663

6. CONCLUSION

The present work dealt with the extraction of the micro Doppler effects of the human body, in order to perform a pedestrian recognition in the near field (0-20 m) of traffic scenarios. A novel multiobjective optimization was made for radar parameters, in order to improve the velocity resolution, in addition, four SVM models were trained to compare the results.

Discussing the optimization, the multiobjective technique was able to improve the velocity resolution, consequently, improving the visualization of the micro Doppler effects, as seen in figure 40. Analyzing the other objectives functions of the optimization, it was also capable of a slight improvement of the range resolution and the maximum detectable range. However, the maximum detectable velocity decreased its value, becoming worse than before. The optimization showed great results, but only for velocities under 7.5 *m/s*, which was enough for a pedestrian recognition under the scope of this study.

About the support vector machines performance, for the four models, the polynomial Kernel, in general, was capable to better classify the pedestrians, while the linear Kernel, presented the worst results, as observed in tables 8 - 11. In addition, the fourth model, which divided the data by distance, was able to generate the best results, as presented in table 12.

However, the support vector machines failed in execute the classification in enough time. For real applications, the searching time plus the SVM execution time should be as less as possible. The best result was achieved with the first training model using polynomial Kernel, but the execution plus searching time was 72 ms, almost the same time as the measurement time, which was 100 ms. For concluding, in each frame the system takes 172 ms to recognize the pedestrian.

For the sequence of this work is recommended to apply the measurement code in a microcontroller, this way the system would be capable to reduce the measurement time from 100 ms to at least 20 ms, and would improve the hardware memory capability. With this, it would be possible to consider the angle information in the classifier training, and to delete one of the optimizations constraints, improving the objective functions.

For the improvement of the velocity resolution, is advisable to use linear prediction in order to improve, even more, the micro Doppler effects [15]. In addition,

would be advisable to try different machine learning methods to improve the classification time. In literature there are different other searching methods faster than the choose in this work, which were not implemented because of time. For reducing the searching time these other methods should be implemented and tested.

The accuracy of the classification was very good for the purpose of this project, but for the continuation, more tests with more objects should be made to improve the complexity of the problem. Scenarios with more than one pedestrian and other objects, like cars and bikers, should be consider. Another point would be to install the radar in a car, to simulated more precisely the real traffic scenarios.

REFERENCES

[1] WORLD HEALTH ORGANIZATION. **Global status report on road safety 2013: supporting a decade of action**. Geneva: World Health Organization (WHO), 2013.

[2] VIRGINIA COMMONWEALTH UNIVERSITY. **VCU study finds cell phones are not the leading cause of distracted driving**. Richmond, 2013. Available at http://www.news.vcu.edu/article /VCU_study_finds_cell_phones_are_not_the_leading_cause_of_distracted.

[3] FESER, M.; MCCONNELL, D.; BRANDMEIER, T.; LAUERER, C. Advanced crash discrimination using crash impact sound sensing (CISS). **SAE Technical Paper**, 2006.

[4] TIGALDI, A.; GUJANATTI, R.; GONGHI, A. Advanced Driver Assistance Systems. **International Journal of Engineering Research and General Science**, Volume 4, Issue 3, May-June, 2016.

[5] SAE international. Automated Driver Levels of Driving. Automation are defined in new SAE international standard. **SAE international's J3016**. Warrendale, October 2014.

[6] TESLA. **Full Self-Driving Hardware in all Cars**. Available at <https://www.tesla.com/autopilot?redirect=no >. Accessed in September of 2017.

[7] WAYMO. **Journey**. Available at < https://waymo.com/ >. Accessed in September of 2017.

[8] BHUIYAN, J. **Uber's autonomous cars drove 20,354 miles and had to be taken over at every mile, according to documents**. Recode. Available at < https://www.recode.net/2017/3/16/14938116/uber-travis-kalanick-self-driving-internal-metrics-slow-progress >. Accessed in September of 2017.

[9] Instituto Parar. **Montadoras investem em segurança**. Brazil, 2016. Available at http://www.institutoparar.com.br/conteudo/278/montadoras-investem-em-seguranca.

[10] Vias seguras, Acidentes com pedestres. Brazil, 2012, Available at <http://www.vias-seguras.com/os_acidentes/acidentes_com_pedestres>

[11] CHEN, V. C.; LI, F.; HO, S. S.; WECHSLER H. Micro-Doppler effect in radar: phenomenon, model, and simulation study. **IEEE Transactions on Aerospace and Electronic Systems**, vol. 42, no. 1, pp. 2-21, Jan. 2006.

[12] DU, L.; LI L.; WANG, B.; XIAO, J. Micro-Doppler Feature Extraction Based on Time-Frequency Spectrogram for Ground Moving Targets Classification With Low-

Resolution Radar. **IEEE Sensors Journal,** vol. 16, no. 10, pp. 3756-3763, May15, 2016.

[13] BELGIOVANE, D.; CHEN, C. C. Micro-Doppler characteristics of pedestrians and bicycles for automotive radar sensors at 77 GHz. **11th European Conference on Antennas and Propagation (EUCAP)**, Paris, 2017, pp. 2912-2916.

[14] KIM, Y.; LING, H. Human Activity Classification Based on Micro-Doppler Signatures Using a Support Vector Machine. **IEEE Transactions on Geoscience and Remote Sensing**, vol. 47, no. 5, pp. 1328-1337, May 2009.

[15] ADRES, M.; ISHAK, K.; MENZEL, W.; BLOECHER, H. L. Extraction of Micro-Doppler Signatures using Automotive Radar Sensors. **Frequenz 2012**, 66(11-12): 371–377, November 2016.

[16] THAYAPARAN, T.; STANKOVIĆ, L.; DJUROVIĆ, I. Micro-Doppler-based target detection and feature extraction in indoor and outdoor environments. **Journal of the Franklin Institute**, Volume 345, Issue 6, September 2008, Pages 700-722.

[17] SAHOO, A. K.; PANDA, G. A multiobjective optimization approach to determine the parameters of stepped frequency pulse train. **Aerospace Science and Technology**, Volume 24, Issue 1, January–February 2013, Pages 101-110.

[18] LIU, Y.; LIAO, G.; YANG, Z.; XU, J. Multiobjective optimal waveform design for OFDM integrated radar and communication systems. **Signal Processing**, Volume 141, December 2017, Pages 331-342.

[19] BAGHEL, V.; PANDA, G.; SRIHARI, P.; RAJARAJESWARI, K.; MAJHI B. An efficient multi-objective pulse radar compression technique using RBF and NSGA-II. **2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)**, Coimbatore, 2009, pp. 1291-1296.

[20] SEN, S.; TANG, G.; NEHORAI, A. Multi-objective optimized OFDM radar waveform for target detection in multipath scenarios. **2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers**, Pacific Grove, CA, 2010, pp. 618-622.

[21] BILIK, I.; TABRIKIAN, J. Radar target classification using Doppler signatures of human locomotion models. **IEEE Transactions on Aerospace and Electronic Systems**, 43, 4 (2007), 1510–1522.

[22] TEKELI, B.; GURBUZ, S. Z.; YUSKEL, M.; GURBUZ, A. C.; GULDOGAN, M. B. Classification of human micro-Doppler in a radar network. **In Proceedings of the IEEE Radar Conference**, 2013, 1–6.

[23] BILIK, I.; TABRIKIAN, J.; COHEN, A. GMM-based target classification for ground surveillance Doppler radar. **IEEE Transactions on Aerospace and Electronic Systems**, 42, 1 (2006), 267–278.

[24] BILIK, I.; KHOMCHUK, P. Minimum divergence approaches for robust classification of ground moving targets. **IEEE Transactions on Aerospace and Electronic Systems**, 48, 1 (2012), 581–603.

[25] CASTRODAD, A.; SAPIRO, G. Sparse modeling of human actions from motion imagery. Springer, **International Journal of Computer Vision**, 100, 1 (2012), 1–15.

[26] HEUER, M.; AL-HAMADI, A.; RAIN, A.; MEINECKE, M. M. Detection and tracking approach using an automotive radar to increase active pedestrian safety. **IEEE Intelligent Vehicles Symposium Proceedings**, Dearborn, MI, 2014, pp. 890-893.

[27] ROHLING, H.; HEUEL, S.; RITTER, H. Pedestrian detection procedure integrated into an 24 GHz automotive radar, **IEEE Radar Conference**, Washington, DC, 2010, pp. 1229-1232

[28] FOELSTER, F.; ROHLING, H.; MEINECKE, M. M. Pedestrian recognition based on automotive radar sensors. **5th European Congress on Intelligent Transportation Systems and Services**, Hannover, Germany, June 2005.

[29] PARK, S.; HWANG, J. P.; KIM, E.; LEE, H.; JUNG, H. G. A neural network approach to target classification for active safety system using microwave radar. **Expert Systems with Applications**, Volume 37, Issue 3, 15 March 2010, Pages 2340-2346

[30] LEE, S.; YOON, Y. J.; LEE, J. E.; KIM, S. C. Human–vehicle classification using feature-based SVM in 77-GHz automotive FMCW radar. **IET Radar, Sonar & Navigation**, vol. 11, no. 10, pp. 1589-1596, October 2017.

[31] LIU, W. J.; KASAHARA, T.; YASUGI, M.; NAKAGAWA, Y. Pedestrian recognition using 79GHz radars for intersection surveillance. **European Radar Conference (EuRAD)**, London, 2016, pp. 233-236.

[32] HEUEL, S.; ROHLING, H. Pedestrian classification in automotive radar systems. **13th International Radar Symposium**, Warsaw, 2012, pp. 39-44.

[33] SKOLNIK, M**. Introduction to radar systems**. Second edition. Singapore. Mcgraw-Hill Book Company, 1981.

[34] BROOKNER, E. **Radar Technology**. First Edition. Artech House, 1977. 444p

[35] BOTSCH, M. Integrated Safety and Assistance Systems, Sumer Semester 17. Technische Hochschule Ingolstadt, 2017.

[36] WOLF, C. Radartutorial.eu. Germany, 1998, Available at < http://www.radartutorial.eu/index.html#this >

[37] MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of Machine Learning**. The MIT Press. Massachusetts, 2012.

[38] KRIESEL, D. **A brief introduction to neural networks**. available at < http://www.dkriesel.com > , Germany. May of 2005.

[39] BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. **Data Mining Knowledge Discovery**, v. 2, p. 121–167, Netherlands 1998.

[40] SHILTON, A. *Design and Training of Support Vector Machines*. 2006. Doctor of Philosophy Thesis (Department of Electrical and Electronic Engineering) – The University of Melbourne, Australia, 2006.

[41] VAPNIK, V.N. **Statistical Learning Theory**. New York: Wiley, September 1998.

[42] CHAPELLE, O.; VAPNIK, V.; BOUSQUET, O.; MUKHERJEE, S. Choosing Multiple Parameters for Support Vector Machines. **Machine Learning**, v. 46, n. 1-3, p. 131-159, 2002.

[43] INRAS GmbH. Radarlog User Manual. Linz, Austria: INRAS GmbH, April 2016, 6 pg.

[44] INRAS GmbH. IFX-TX2RX16-AN77-02 FMCW Basics. Linz, Austria: INRAS GmbH, March 2017, 5 pg.

[45] INRAS GmbH. IFX-TX2RX16-AN77-06 Range-Doppler Processing. Linz, Austria: INRAS GmbH, April 2017, 7 pg.

[46] RASTRIGIN, L. A. About Convergence of Random Search Method in Extremal Control of Multi-Parameter Systems", **Avtomat. i Telemekh**., 24:11 (1963), 1467–1473

[47] HOLLAND, J. H. **Adaptation in natural and artificial systems**, University of Michigan Press, 1975.

[48] HWANG, C. L.; YOON, K. **Multiple attribute decision making: methods and applications**, Springer-Verlag, New York 1981.

[49] BRANS, J. P.; VINCKE, P. H. A Preference ranking organization method, **Management Science**, 1985, Vol. 31, No. 6 (June 1985), pp. 647-656.

[50] HEUEL, S.; ROHLING, H. Two-stage pedestrian classification in automotive radar systems. **12th International Radar Symposium (IRS),** Leipzig, 2011, pp. 477-484.

[51] ROSENFELD, A.; PFALTZ, J. L.; Sequential Operations in Digital Picture Processing Journal of the ACM (JACM) ,Volume 13 Issue 4, Oct. 1966.

[52] MACQUEEN, J. Some methods for classification and analysis of multivariate observations, **Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability**, Volume 1: Statistics, 281--297, University of California Press, Berkeley, Calif., 1967.

APPENDIX A – RADARLOG HARDWARE

Inras GmbH provided the radarlog used to do the measures. The radarlog is a novel microwave radar evaluation platform for R&D laboratories, educational institutions, and for rapid prototyping, which is primarily designed to operate different continuous wave radar frontends from a single baseband system. Figure 42 shows a picture of the Radarlog.

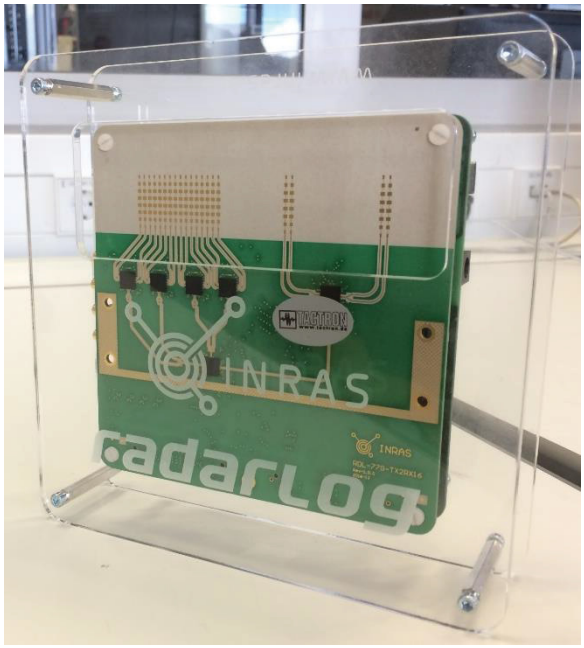


Figure 42 – Radarlog 77GHz.
Source: The author (2017).

The main features of the radarlog are: Up to sixteen input channels, two AFE5801 ADCs, processing with FPGA, onboard USB 3.0 that can be used to log data in real time, maximum transmit frequency of 79 GHz and synchronization of an external clock [43].

Figure 43 illustrates a simplified block diagram of the baseband board is sketched. The central processing element is built with an FPGA (Arria V) suited for real-time radar signal processing with up to 16 differential input channels in parallel. For processing the analog intermediary frequencies (IF) signals the board incorporates two AFE5801 (Radar receive path AFE) from Texas Instruments. Each AFE5801 provides a full receive path for eight differential input channels. Therefore, the baseband board is suited to directly process the IF signals of CW radars without the need for additional analog circuitry for signal conditioning. The AFE5801 input

channels are routed to the RF connector of the board. In addition, 24 + 64 digital IO signals are available at the RF connector, in order to configure and operate the radar frontends [43].

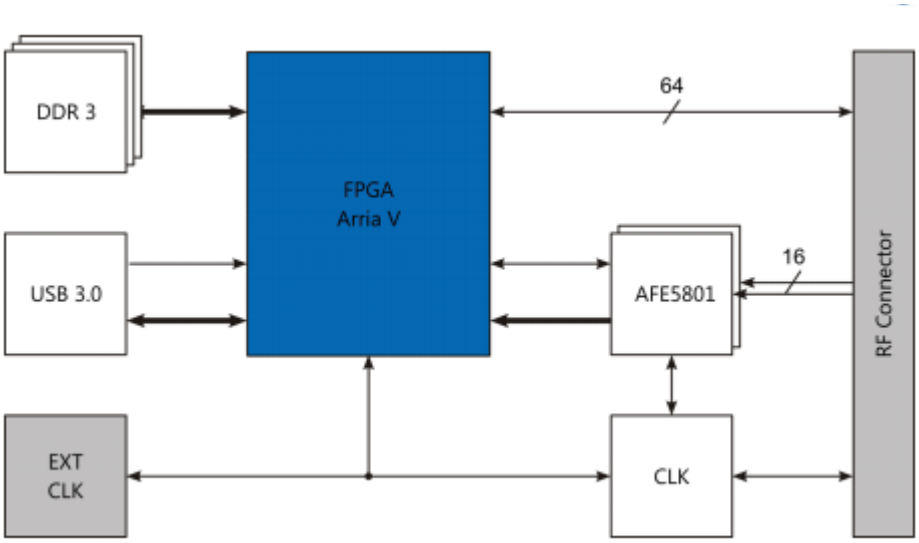


Figure 43 – Block diagram of the Radarlog with the main building blocks. Source: Inras GmbH (2016) [36].

The FPGA controls and configures the AFE5801 devices and in addition it is used to preprocess the discretized IF signals with the aim to reduce the overall data rate. The availability of powerful and re-configurable signal processing frameworks for the FPGA enables real-time signal processing without the need to deal with time consuming FPGA development. A flexible timing unit in the FPGA is used to implement measurement cycles with a precise timing. A synchronization of the ADC timing to the clock of the radar frontend can be achieved by means of an onboard clock distribution circuit (LMK4033). This device can also be used to synchronize the sampling and the RF circuit to an external LVDS clock signal. This enables the synchronization of multiple radar systems to an external clock. In addition, the timing unit allows to program different ramp scenarios. Hence, uniform and non-uniform ramp scenarios with different ramp durations and slopes can be programmed independent from the utilized PLL in software [43].

The computer used to make the sampling and calculations was an ASUS S46CB with an intel Core i7-3537U CPU processor of 2.0 - 2.5 GHz, 6 GB of RAM memory and 64 bits Windows 10 operational system.