POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Vehicle System Integration of an Adaptive Cruise Control for an Assisted Driving Prototype Vehicle

Supervisors

Prof. Andrea TONOLI

Prof. Nicola AMATI

Eng. Eugenio TRAMACERE

Candidate

Fabio BILLERO

December 2022

Abstract

The proposed thesis project refers to the Vehicle system integration of an Adaptive Cruise Control ACC) for an assisted driving prototype vehicle, the SCD22 race car of Squadra corse Driverless student team. Specifically, starting from the informations sent by a radar sensor, it has been developed a control strategy able to command the vehicle's longitudinal dynamic according to the position of the obstacle in front it. The developed ACC maintains the desired velocity set by the driver in case of absence of a target vehicle in front and guarantees a safe distance adjusting the longitudinal velocity according to that one of the preceding car. Furthermore, the controller has been tested on several driving conditions, with the objective of testing its action under real driving scenarios. Thus, it has been proven to react correctly when a lead vehicle rapidly exits from the radar sensor field of view (FoV), or a vehicle approaches our line becoming the new target obstacle for the radar sensor. Firstly, the information exiting from a Short Range Radar sensor (SRR), the AWR1642 have been decoded in MATLAB/Simulink environment. The designed Simulink model of our decoder has been deployed on the TI Launchpad board which receives the data via serial UART communication from the radar sensor and extracts from the data packets, the relative position and velocity of the vehicle in front of the sensor. These informations about the target obstacle are subsequently sent through serial CAN communication to a real time Electronic Control unit (ECU), the dSPACE MicroAutobox III used for performing fast in-vehicle function prototyping. The proposed ACC is deployed on the retained ECU, which is based on a custom state machine and the Constant time Gap (CTG) control law. The first one determines the ACC states, in particular the activation and deactivation of the system, which is switched on at the desired time by the driver through a button specifically inserted in the car's cockpit. When the button is pressed, the speed of the car is cruised, setting the reference speed to that specific time instant. The CTG instead is used to regulate the longitudinal behaviour of the car cruising the velocity of the vehicle according to the information received from the radar sensor. The CTG includes an extension of the conventional ACC system, in particular the ACC with Stop-and-Go features which is aimed to be used in urban scenarios where the lead vehicle can stop completely and restart afterwards. In this case the presented ACC has the capability to stop the ego vehicle completely at desired safe distance from the target vehicle. The control implementation has been integrated on a prototype car in a similar way to a common integration approach for road vehicles, in which the controller is initially designed by simulating the information extracted from the radar sensor in such a way to test its correct functioning and then verifying the accuracy of the information sent by the radar sensor in a static environment where the reliability and stability of these data checked. Ultimately, the ACC has been tested in different real driving conditions, based on which the goodness of the controller in terms of passenger comfort has been assessed. The results show that the ACC works fine in all the tested driving scenarios. The controller works correctly for values of relative distance that ranges between 7 and 50 meters and relative velocity values between -20 km/h and +20 km/h. Future work should focus on the controller side, designing an ECO ACC that features an advanced controller that bases the throttle pedal position on the efficiency map of the electric motors installed on the prototype vehicle.

Acknowledgements

ACKNOWLEDGMENTS

Forse questa è la parte più complicata da scrivere, il merito di questo percorso e la realizzazione di questo progetto va a molte persone e sarbebe impossibile elencarle qua tutte.

Ringrazio la mia famiglia che mi ha permesso di intraprendere questo percorso di studi, per avermi sempre sostenuo, supportato e sopportato in qualsiasi momento. Ringrazio tutti i miei aimici più stretti, che mi sono sempre stati accanto lungo tutti questi anni di studio, ma anche quelli che mi sono stati vicino per un breve periodo del mio percorso. Ringrazio tutte le persone che ho conosciuto in questi anni di università. Ringrazio le persone che ho conosciuto, e mi sono state accanto negli ultimi anni. Ringrazio tutti i membri del team Squadracorse driverless (SCD) per essere stati sempre disponibili per la realizzazione di questo progetto, per avermi fatto crescere personalmente e professionalmente. Ringrazio il LIM, in particolare i dottorandi Eugenio e Stefano, per il supporto durante l'intero progetto, per avermi insegnato molto come ingegnere e per aver creduto sempre nelle mie potenzialità. Ringrazio le persone che hanno sempre creduto in me ma anche quelle che non lo hanno mai fatto. Non ci sono parole per descrivere quanto tutti vuoi siete stati importanti lungo tutto questo periodo di studi.

Fabio, Billo, Birillo

Table of Contents

List of Tables

List of Figures

Acronyms

ASB

Autonomous Braking System

ACC

Adaptive Cruise Control

ADAS

Advance Driver Assistance System

ADC

Analog to Digital Converter

AoA

Angle of Arrival

API

Application Programming Interface

BPM

Binary Phase Modulation

CCS

Code Composer Studio

CTG

Costant Time Gap

DAS

Driving Assistance System

FMWC

Frequency Modulated Continuous Wave

LRR

Long Range Radar

MIMO

Multiple Input Multiple Output

MMIC

Monolithic Microwave Integrated Circuit

MRR

Medium Range Radar

RCS

Radar Cross Section of the Target

RRE

Radar Range Equation

SRR

Short Range Radar

TI

Texas Instrument

TLV

Type-Lenght-Value

UART

Universal Asynchronous Receiver-Transmitter

USRR

Ultra Short Range Radar

Chapter 1

Introduction

Since the born of the automobile, one of the aim of the automotive sector has been to interface with many innovative devices and systems able to increase the comfort of the drivers and reducing fatigue while driving. One of the main factors which cause the car accidents can be connected to driver distraction [1]. For this reason almost all cars which commonly use in our daily life, have different Driving Assistance System (DAS), such as the ABS, ESC, TCS and CC that help the drivers to maintain a correct control of the car in case of critical situation. These assistance system are based on proprioceptive sensors, so it is based on all the information that derives from the vehicle, such as velocity, acceleration and so on [2].

In the last years thanks to the improvement of the sensors available on the cars, the DAS system has been upgraded to the Advance Driver Assistance System (ADAS), which differently from the previous one is based on exteroperceptive sensors, so those sensors that take the information from the external environment, like road, vehicle, and traffic situations.

At the beginning of the 21st century, one of the main ADAS system implemented in automobiles is the Adaptive Cruise Control (ACC) [3], this system was initially implemented only on the manufacturer's top car. One of the goals that the automotive market has set itself is to expand the ACC system not only for luxury vehicles but also for mid-range vehicles [4].

This system helps to reduce the number of accidents on the roadways. The ACC has been developed for the driver's safety in possible dangerous conditions, such as the sudden cut in of a vehicle or the sudden braking condition of the vehicle located in front of our cars.

To have a correct working principle of the ACC, the cars must be equipped by various sensors which give information on the environment surrounding the car. In particular in modern and more technologies cars it is possible to have different sensors such as Radar, stereo camera and LIDAR, which, in the case of radar and lidar, are able to identify objects positioned in a certain range, determine their position and relative velocity, instead in the case of stereo camera it is able to identify the objects located in its Field of View, it is able to identify cars, pedestrian and road signs.

In this thesis we focus on the physical implementation of the radar on an assisted driving prototype vehicle and how an ACC system based on the Hardware can be implemented on the vehicle. Secondary this study explain how safety system, in particular the ACC is able to decrease the mortality rate and to avoid conditions of possible critical situations. The implementation on the vehicle starts from a DAS system, the Cruise Control (CC) is a servomechanism system that is used to maintain constant the speed of the car when it is activated by the driver. In this way the throttle value is not taken over by the throttle pedal but by the system which activate an actuator able to control the acceleration and speed of the car. The common CC it is updated to the ACC which uses a radar sensor data to collect the information on the environment surrounding the car, in particular is able to identify longitudinal position, angular position and relative velocity of the cars located in front of it and uses these data to avoid unnecessary acceleration or deceleration, moderate the velocity of the host car, and to determine the space needed to avoid forward collisions. With the implementation of the Adaptive Cruise Control system the car is able to accelerate and decelerate in an autonomous way according to the distance between the host vehicle and the lead vehicle.

1.1 AWR1642 ODS Evaluation Mode (EVM)

For the this study the has been used a device manufactured by Texas Instruments (TI), it is an integrated single-chirp FMWC radar, capable to operate at in the band of 76-81 GHz. The device have a monolithic implementation of a 2 Transmit Antenna (TX) and 4 Receive Antenna (RX). It integrates the DSP subsystem, which contains TI's high performance C674x-DSP, able for the radar signal processing. The radar includes two programmable ARM R4F-based processor, which is responsible for radio configuration, control and calibration. The Hardware has by default the following setting:

- 76-to-81 GHz Frequency Band
- Four receive Channels and Two transmit channels.
- Two processors, one dedicated to the advanced signal processing (DSS), the second one dedicated to the overall operation of the device (MSS)
- TX power: 12dBm

- I/O support dual voltage 3.3 V/1.8 V
- Maximum range resolution: 80 m for cars and trucks, 20 m for small objects
- Maximum speed resolution: 1.87 km/s
- Maximum unambiguous relative speed: 90 km/h for cars and truck up to 80 m, 18km/h for small objects detected up to 20 m
- Field of View (FOV): covers ±55°

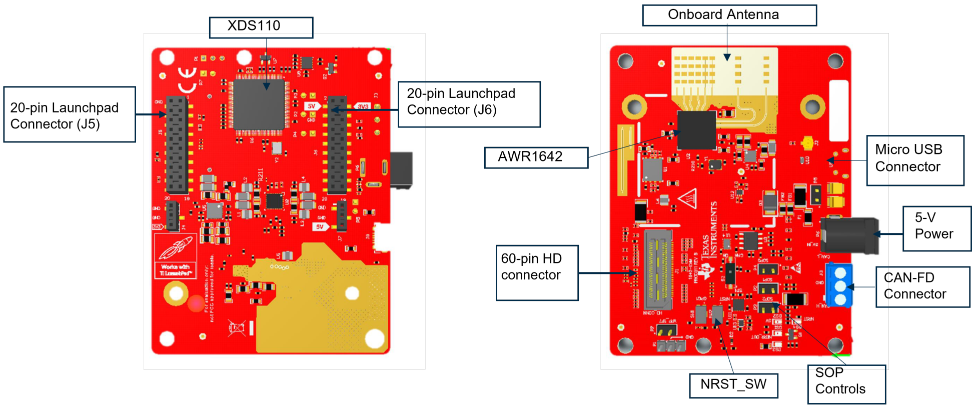


Figure 1.1: AWR1642 radar

Chapter 2

State of Art

2.1 Radar Fundamentals

Before diving into the discussion of the proposed method, this chapter is dedicated to the theoretical background considered during this work. Radar stands for [5] "Radio Detection and Ranging" and its working principle is to detect range, velocity, and angle of arrival of objects in front of it. For this project, the used radar exploits the principle of Frequency Modulated Continuous Wave (FMWC), which helps in measuring the detected objects range and velocity.

At the heart of the FMWC radar there is a signal called Chirp. A Chirp is a sinusoidal or sine wave whose frequency increase linearly with time. The chirp can starts with a frequency f_c and gradually increase its frequency ending up with a frequency f_c plus B, where B is the bandwidth of the chirp. Another conventional way to represent the chirp is the frequency versus time plot, In this representation the chirp is characterized by a start frequency f_c, bandwidth B, duration T_c, and by a slope S which defines the rates at which the chirps ramp up.

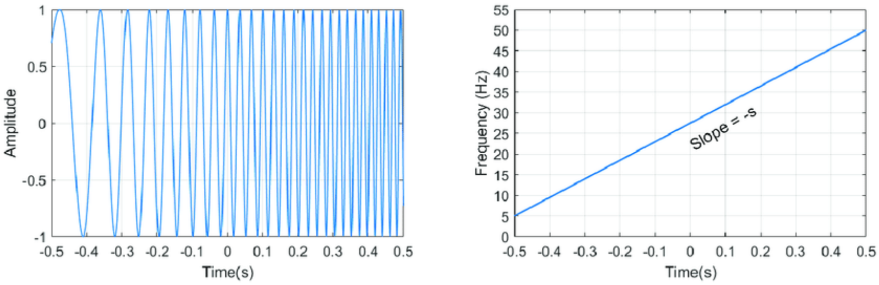


Figure 2.1: Amplitude versus time and frequency versus time representation of the chirp

For example, in the figure above, the chirp starts at a frequency of 5 GHz, sweeping a bandwidth of 45GHz with a time period of 1s, thus ending up at a frequency of 50GHz with a slope of 45MHz/us. As will discuss later the bandwidth B and the slope S are important parameters which defines system performance.

After the explanation on what a chirp is, it is possible to understand how an FMWC radar works. Supposing to have a simplified block diagram of an FMWC radar with a single transmitting antenna Tx and a single receiving antenna Rx, the radar operates as follow. The synthesizer generates a chirp, this chirp is transmitted by the Tx antenna, the chirp is reflected off an object and the reflected chirp is received at the Rx antenna. The Rx and the Tx signal are mixed and the resulting signal is called Intermediate Frequency signal (IF signal).

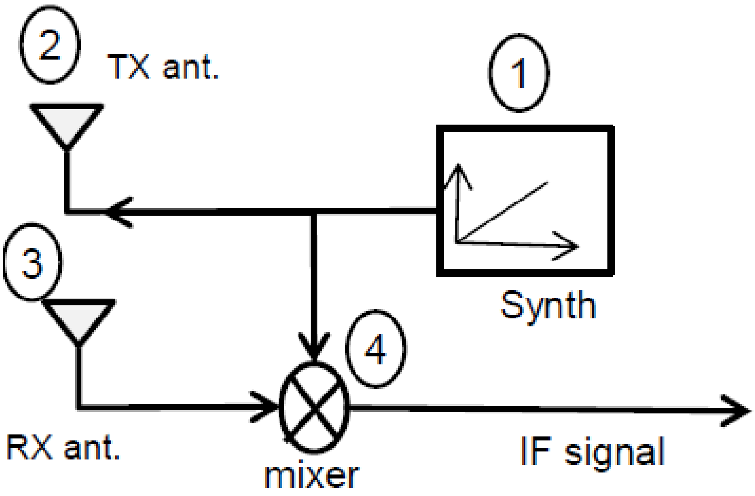


Figure 2.2: FMWC block diagram

The mixer which mix the Rx and the Tx signal is a three port device with two inputs and one output. From the theoretical point of view, it is possible to suppose to have two sinusoids in the input ports and as the output of the mixer a sinusoid with the following two properties:

- The instantaneous frequency equal to the difference of the instantaneous frequencies of the two input sinusoids

- The starting phase of the output sinusoid is equal to the difference of the starting phases of the two inputs sinusoids.

From the formulation point of view it the mixer can be expressed by:

$$x_1 = sin(w_1 t + \phi_1)$$

$$x_2 = sin(w_2 t + \phi_2)$$

$$x_{out} = sin[(w_1 + w_2)t + (\phi_1 + \phi_2)] \tag{2.1}$$

In order to understand better the working principle of the mixer it is possible to represent it in the frequency-time plot. In this plot it is possible to see that the received chirp is a time delay replica of the transmitted chirp, and in order to generate the IF signal, must be subtract the received signal from the transmitted one.

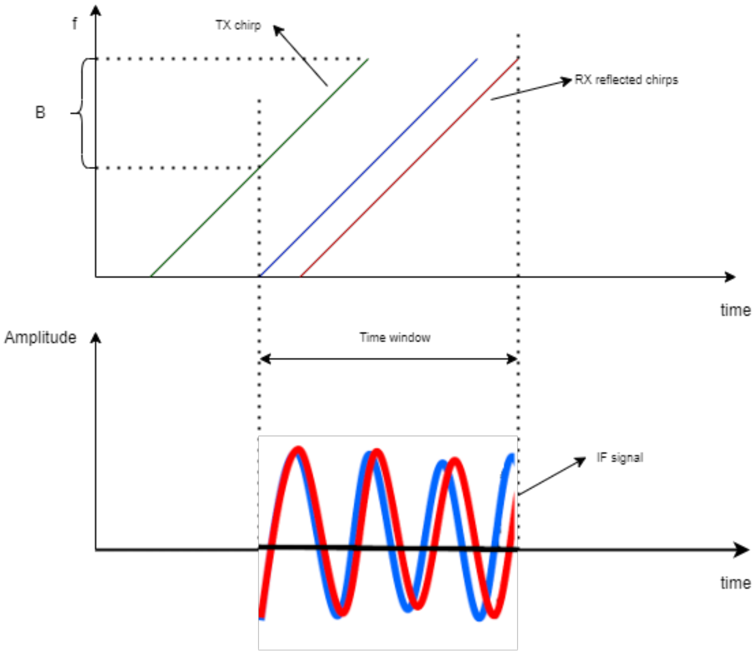


Figure 2.3: Amplitude-time chirp and IF signal

As it possible to see, the two signals are a fixed distance from each other. And the fixed distance is given by the slope of the chirp times the round trip delay. So a single object in front of the radar produces an IF signal that is a constant frequency tone. The frequency of this tone is S_τ = S2d/c where d is the distance of the object and c is the speed light. Another point worth noting is that the round trip delay τ is usually a small fraction of the total chirp time.

2.1.1 Fourier Transform

A Fourier transform converts a time domain signal into a frequency domain. So a single wave in the time domain produce a single peak in the frequency domain.

Likewise two waves in the time domain should result in two peaks in the frequency domain. However this is not always the case, but in some cases it is possible to have one single peak for both the waves like the Fourier transform of the figure 2.4

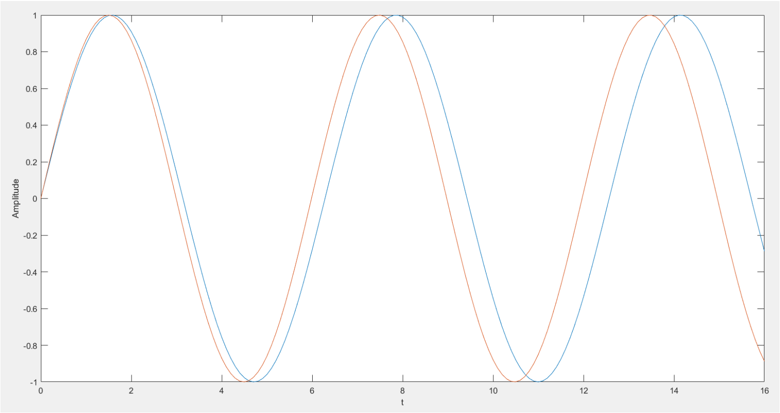


Figure 2.4: Time domain signal

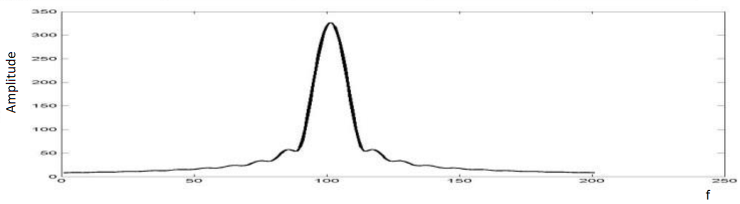


Figure 2.5: Fourier transform

In order to overcome this problem. it is possible to see that doubling the observation window, the two waves that previously corresponded in one single peak, now they corresponds at two peaks. So longer id the observation period and better is the resolution.

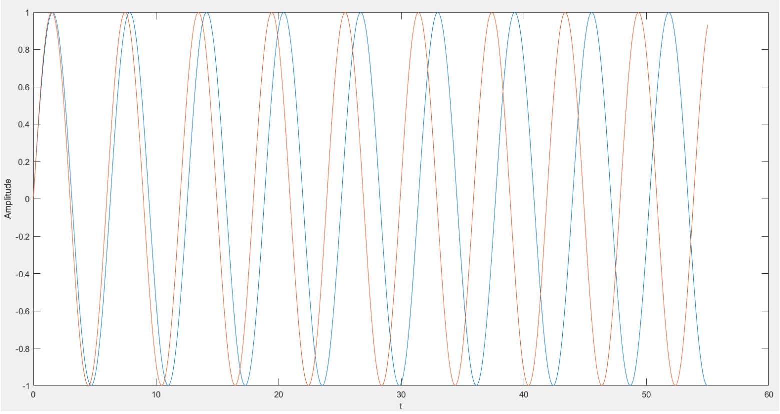


Figure 2.6: Time domain signal with longer observation period

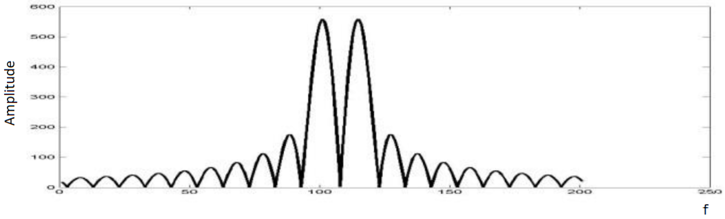


Figure 2.7: Fourier transform of the longer observation period

2.1.2 Multiple objects detection

Until now has been treated the single object detection. It is possible to extend the same principle to where there are multiple objects in front of the radar. So, the radar transmitting a single chirp and it gets multiple reflected chirps from different objects. Each delayed by a different amount of time which depending on the distance to the object. So the IF signal will have tones corresponding to each of these reflections, with frequency that is directly proportional to the range. So the IF signal with lowest frequency corresponds to the closest object, instead the the IF signal with highest frequency correspond to the farthest one.

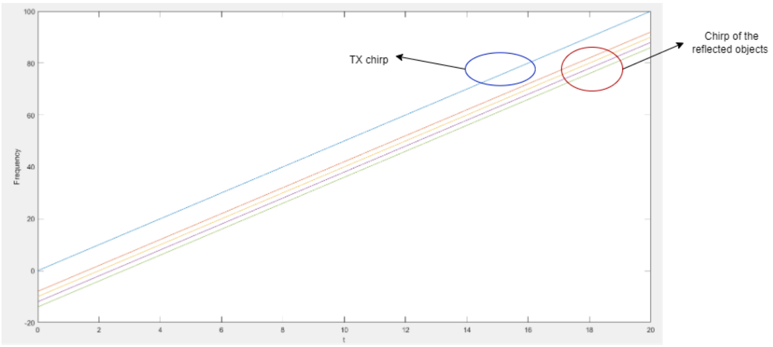


Figure 2.8: Multiple tones signal

A Fourier transform of this IF signal will reveal multiple tones, and the frequency of each peak will be directly proportional to the range of the corresponding object.

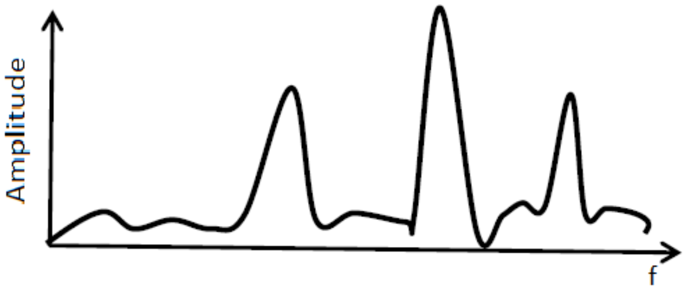


Figure 2.9: Fourier transform of the Multiple tones

2.1.3 Range resolution

The range resolution of the radar is the ability to distinguish between object that are very closely to each other. In some cases it is possible to have two sine waves in the Amplitude-frequency plot, but if the two objects are very closely, they are represented by one single peak in the frequency spectrum like in the Figure 2.4 Taking a cue from one recap of Fourier Transforms, one option is to extend the observation window of these two sine waves by increasing the length of the IF signal. In this way the chirp is extended which then extends the duration of the IF signal. It should be noted that increasing the duration of the IF signal proportionally increase the bandwidth of the chirp. So this means that increasing the bandwidth corresponds to a better range resolution.

So two objects at a distance Δd will have the IF frequencies separated by δf given by:

$$\Delta f = \frac{S2\Delta d}{c} \tag{2.2}$$

For these two frequencies to show up as as distinct peaks in the IF frequency spectrum, the frequency separation delta, must be greater than 1 by the duration of the IF signal, which is virtually equal to the duration of the chirp T_c; in formula:

$$\Delta f > 1/T_c$$

Knowing that the slope times the duration of the chirp is actually the Bandwidth of the chirp, after some rearrangements of the formulas above, it is possible to obtain the following expression:

$$\Delta d > \frac{c}{2B} \tag{2.3}$$

Which says that two objects can be separated in the IF frequency spectrum as long as the distance (separation) between them is greater than the ratio of the speed of light to twice the bandwidth of the chirp.

So the main point here is that the range resolution depends only on the bandwidth swept by the chirp, and is given by the expression over here, speed of light divided by twice the bandwidth.

2.1.4 Digitizing the IF signal

In most radar the IF signal is digitized for subsequent processing. So it's first low pass filterd and then digitized by an ADC, and sent to a suitable processor such ad a DSP. Whenever we digitizing a signal, we need to know what is the bandwidth of interest so that the low pass filter and the ADC sampling rate can appropriately set. It is important to underline that the bandwidth of the interest of the IF signal depends on the desired maximum distance:

$$f_{IFMAX} = \frac{S2d_{max}}{c}$$

And correspondingly, the bandwidth of interest is going to be from zero to this maximum IF frequency which means that the low pass filter should have a cut-off frequency, which is beyond this IF max. And also the ADC should have a sampling rate which is greater than the same value. Supposing a complex baseband signal (hence half the Nyquist rate of a real signal) IF bandwidth is thus limited by the ADC sampling rate (F_s)

$$F_s = \frac{S2d_{max}}{c}$$

Reversing the formula above it is possible to obtain:

$$d_{max} = \frac{F_s c}{2S} \tag{2.4}$$

In practice and according to the datasheet, two IF frequencies are used in detection process based on current sample rate mode.However only the following one will be used in complex 1x sampling mode:

$$IF_{MAX} = 0.9 \cdot ADC_{Samp} \tag{2.5}$$

In complex 2x sampling mode:

$$IF_{MAX} = 0.9 \cdot \frac{ADC_{Samp}}{2} \tag{2.6}$$

2.1.5 The Phase of the IF signal

In the previous chapter has been seen the relationship between the frequency of the IF signal and the distance to the object. In this chapter will be analyzed the relationship between the phase of the IF signal and the distance of the object.

A sinusoid in the time domain produces a peak in the frequency domain, the location of the peak corresponding to the frequency of the sinusoid. The signal in the frequency domain is a complex number with an amplitude and a phase, A complex number can be mathematically represented in the $Ae^{j\Theta}$ form, where A is the amplitude and Θ is the phase. Alternatively, it can also be pictorially represented as a phasor, which is a vector with a length corresponding to the amplitude A and a direction corresponding to the phase Θ.

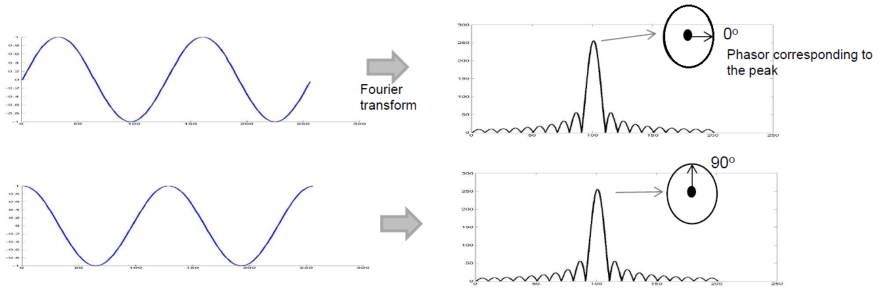


Figure 2.10: Phase of the receiving sinusoid

An important property of the Fourier transform is that the phase of the peak corresponds to the initial phase of the sinusoid. So a sine wave starts with a certain initial phase. And that phase is reflected in the phase of the peak of Fourier transform. So if we take two sine waves with the same frequency but with the second one with an offset of 90 degrees, the Fourier transform will have a peak which is at the same location of the first sine wave but with a phase that is offset by 90 degrees.

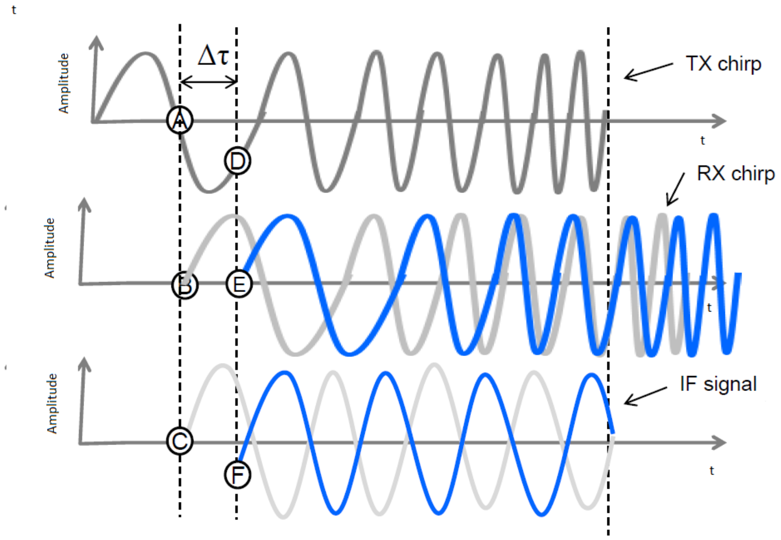


Figure 2.11: Phase of the IF signal with a delay

The figure 2.11 represent the Amplitude-time plot of the Transmitted, received and IF signal. As mentioned in the previous chapters, the IF signal of a single object is a constant frequency signal, in other word a signle sinusoid which can be represented as:

$$Asin(2\pi ft + \phi_0)$$

where f is the the frequency given by $f = \frac{S2d}{c}$ and the phase ϕ_0 is the phase of the IF signal at the C point in the figure.

It is important to re underline that the initial phase of the IF signal ϕ_0 is the difference of the initial phase of the TX and RX signal. i.e the phase of the TX signal at point A and the phase of the Rx signal at point B Let's considering now a small amount of displacement, in the forward direction, made by the object. From the figure it is possible to note that both the received signal and the IF signal are shifted by the small amount $\Delta\tau$ (blue curves). Now the starting phase of the IF signal that is the phase of the IF signal at point F, is going to be difference of the phase at D and the phase at E. The phase of the RX chirp at E is going to be the same as the phase of the earlier RX signal at B. But the phase of the TX chirp at D is going to be the earlier phase A with an additional offset given by:

$$\Delta\Phi = 2\pi f_c \Delta\tau = \frac{4\pi\Delta d}{\Lambda} \tag{2.7}$$

So for a single object in front of the radar the IF signal is a tone with a frequency that is proportional to the distance of the object and also has starting phase which has the property that it changes linearly with small changes Δd in the distance of the object.

2.1.6 Velocity Estimation

The measurement of the velocity is an important tool of the FMWC radar.The measurement principle is the following:

- It transmits two chirps separated by a time of T_c.
- The range-FFTs corresponding to each of these chirps will have peaks in the same location, but with different phase.
- The measured phase difference ω between the phases of these two peaks will directly correspond to the motion of the object. And note that if the velocity of the object is v, the object would have moved a distance $d = v \cdot T_c$ during this time period T_c.

$$\omega = \frac{4\pi v T_c}{\lambda} \tag{2.8}$$

$$v = \frac{\Lambda \omega}{4\pi T_c} \tag{2.9}$$

Note that the method described above relies on a phase difference measurement, which is unambiguous, only as long as this difference is within plus or minus 180 degrees or plus or minus pi radians. If it is analyzed the phasor motion across two chirps, for positive velocity the phasor moves anticlockwise, instead for negative velocities the phasor moves clockwise. If the movement in che clockwise or anticlockwise direction is more than 180 degrees, there would be an ambiguity. Therefore the maximum relative velocity V_{max} that can be measured by two chirps spaced T_c apart is given by:

$$|\omega| < \pi$$

$$\frac{4\pi V T_c}{\lambda} < \pi$$

$$v_{max} = \frac{\lambda}{4T_c} \tag{2.10}$$

It is important to note that T_c depends on the frequency sweep, so faster is the ramp of the frequency ramp the lower the T_c and therefore the greater the ambiguity about the speed.

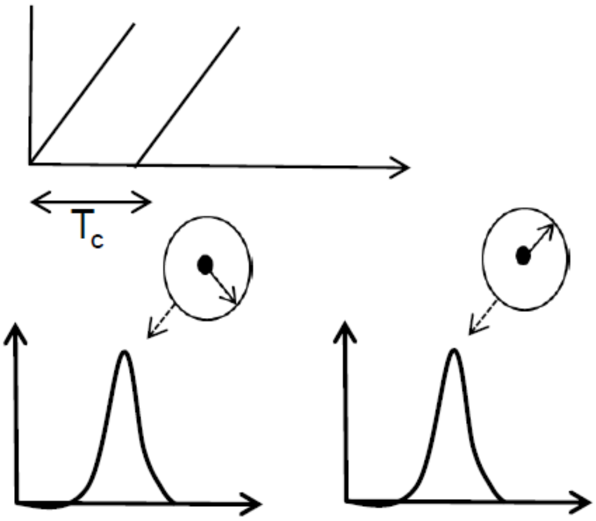


Figure 2.12: Speed estimation

2.1.7 Doppler FFT

This technique is used when there are multiple object in front of the radar with different speeds. In case of multiple objects, the phasor at the peak is going to have components from the velocity from both objects, so the simple phase comparison technique does not work here. In order to avoid this problem, it is possible to sent a series of equi-spaced chirps instead of only two chirps. And so there will be N equi-spaced chirps called "Frame", where the range FFTs corresponding to each of these chirps would have peaks in identical locations.

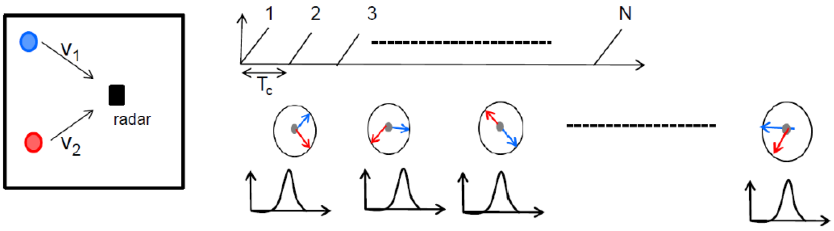


Figure 2.13: FFT's on a complex sequence

But the discrete sequence corresponding to the phasor of these peaks would have two rotating phasors rotating at frequencies of ω_1 and ω_2, corresponding to the two vwlocities V_1 and V_2.

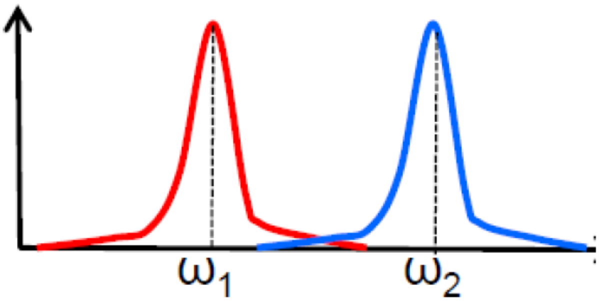


Figure 2.14: Speed of two objects

So a range FFT on this discrete sequence would then show two peaks corresponding to discrete angle of frequencies of ω_1 and ω_2, respectively. And then having measured ω_1 and ω_2, we can then back-calculate the velocities.

2.1.8 2D FFT

Has been expressed that objects at different ranges can be resolved using a range-FFT. And then Doppler FFT's done across subsequent chirps in a frame, resolves objects which may be at the same range but have different velocities with respect to the radar.

The ADC samples corresponding to each of these jobs can be visualized as being stored as the rows of the matrix.

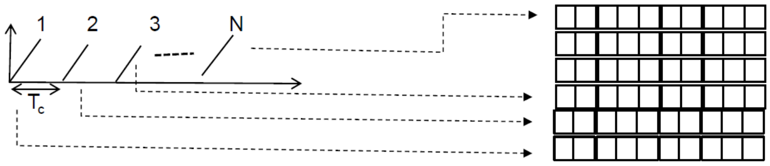


Figure 2.15: 2D FFT processing in a matrix

Each row on the figure, 2.15, corresponds to a sample of a specific chirp. and the range-FFT is then performed on each row. And this range-FFT resolves objects in range. Note that the X-axis is the frequency corresponding to the range FFT bins, but since range is proportional to the IF frequency, it is possible to plot, equivalently this axis as the range axis. Subsequently an FFT called, doppler-FFT is performed along the columns of these range-FFT results, and this resolve objects in the velocity dimension.

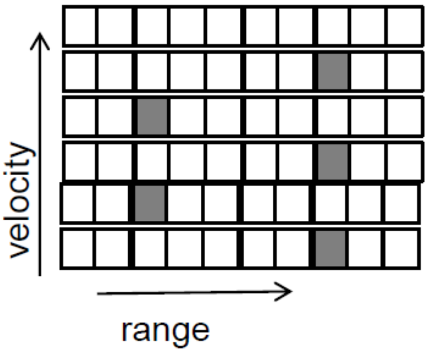


Figure 2.16: Objects Identification on the Matrix

So in figure 2.16 it is possible to note that the third range-bin has two objects at different velocity and the ninth range-bin has three objects at different velocities. Also in this case the y-axis is the doppler, is the discrete angular frequencies corresponding to the Doppler-FFT, but due to the fact that those discrete angular frequencies are proportional to the velocity, it is possible to represent the y-axis as velocity axis.

2.1.9 Velocity resolution

The velocity resolution is the smallest speed difference between two moving objects. It is the capability to represent two separate objects at velocity v_1 and v_2 when these are very closely each other. The velocity resolution of the radar is inversely proportional to the frame time T_f, and is given by:

$$v_{res} = \frac{\lambda}{2T_f} \tag{2.11}$$

Where $T_f = \mathrm{N}T_c$ is the frame time, N is the number of equi-spaced chirps that has been transmitted, and T_c is the total chirp time, which is made by the chirp time plus idle time between two consecutive chirps.

$$T_c = T_{CD} + T_{ID} \tag{2.12}$$

2.1.10 Angle of arrival estimation (AoA)

As has been mentioned in the previous chapter, the phase of the IF signal is very sensitive to small displacement of the objects that are situated in front of the radar. Angle of arrival estimation exploits a similar concept. Indeed the angle estimation requires at least two RX antennas.

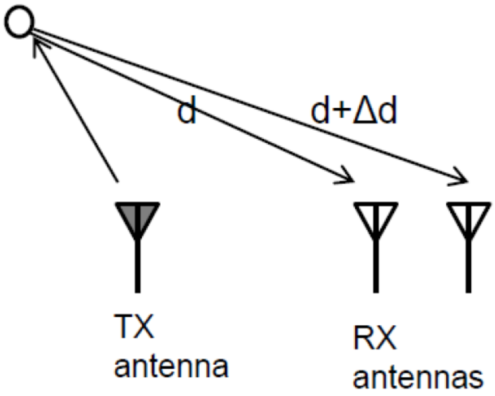


Figure 2.17: Reflection of the object on 2 RX antennas

In figure 2.17 it is possible to see that the transmit antenna, transmits a chirp, two rays are reflected by the object, one to the first RX antenna, with distance "d" and the second one to the second RX antenna with a distance "d+Δd". This additional distance results in an additional phase of ω equal to:

$$\omega = \frac{2\pi\Delta d}{\lambda}$$

which change the overall phase in 2D-FFT.

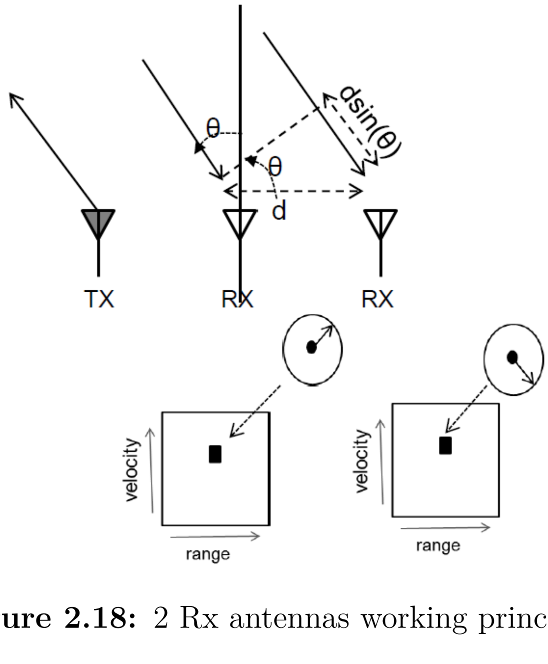


Figure 2.18: 2 Rx antennas working principle

The figure 2.18 explain how to estimate the AoA

- The TX antenna transmits a frame of chirps (series of chirps)
- Supposing to have one TX antenna and two RX antennas, and an object far enough compared to the distance between the two receiving antennas "d", the chirp is transmitted and reflected by the object on the RX antennas, The first antenna receive the chirp with an angle of arrival of Θ, while the second RX antenna receive the chirp with an higher distance with respect the first one, that is given by dsin(Θ)
- The transmitted antenna transmits a frame chirps, and the data is received at each antenna which processes the data to create a 2D-FFT matrix with a peak corresponding to the range and velocity of the object. This peaks are at the same location in the frequency spectrum but with different phases.
- The measured phase difference ω can be used to estimate the angle of arrival of the object.

$$\omega = \frac{2\pi d sin(\Theta)}{\lambda} \tag{2.13}$$

$$\Theta = arcsin(\frac{\lambda\omega}{2\pi d}) \tag{2.14}$$

Observing the formula 2.13 it is possible to note that the relationship between the measurement Θ and the measured phase difference ω is a nonlinear one due to the sin of Theta. This means that when $\Theta = 0°$ ω is most sensitive to the changes, instead with the increasing of the value of Theta, omega become less sensitive until becoming zero when $\Theta = 90°$. So in other words, the radar has the best measurement when the object is located in front of the sensor. Supposing to have a theoretical Field of View (FOV) of the radar of 55°, in both sides, and supposing to have three objects in front of the radar with the same size, surface and material, the objects have three different angle with respect to the sensor, in particular, Θ_1, Θ_2 and Θ_3, with Θ_1 and Θ_2 inside of the FOV and Θ_3 out of the FOV. Due to their position Object 1 and Object 2 reflect a certain amount of energy back to the radar, instead the Object 3 does not reflect energy. Between the 2 objects that are inside the FOV of the radar, the Object 1 has more probability to be detected by the radar, this is due to the fact that that the absolute value of Θ_1 is less than the absolute value of Θ_2.

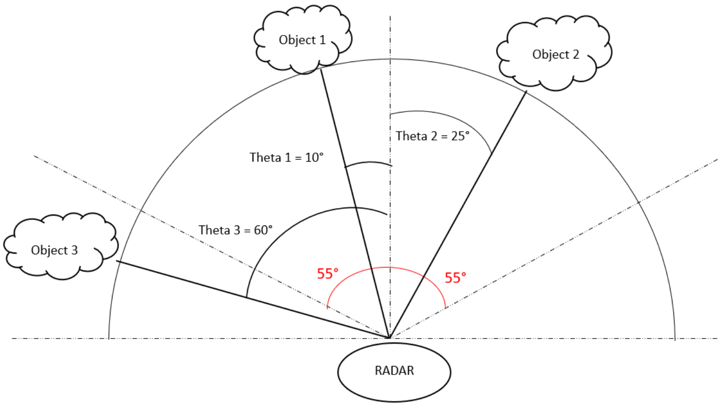


Figure 2.19: AoA accuracy with respect to angle

As in the velocity measurement the phasor moves in clockwise or anticlockwise direction, but in this case the rotating movement is given by the position of the object with respect to the radar instead of the velocity. In particular if the object is located on the left of the radar, the phasor moves anticlockwise when you go from the first RX antenna to the second one. Likewise, for an object to the right of the radar, the phasor moves clockwise. Also in this measurement there is an ambiguity when the phase difference between two antennas is in the range [-180°, 180°] so the absoluta value of omega is less than 180°. Therefore the maximum Theta that con be measured is given by:

$$|\omega| = \pi$$

$$\frac{2\pi d sin(\Theta)}{\lambda} < \pi$$

$$\Theta_{MAX} = arcsin(\frac{\lambda}{2s})$$

Noting that if there is a spacing d= $\frac{\lambda}{2}$ between two RX antennas, the biggest possible FOV is ± 90°. For the AWR1642 the distance between two consecutive antennas id 2.1 mm and so the wavelenght is:

$$\lambda = \frac{c}{f} = \frac{3x10^8 m/s}{77x10^9 Hz} = 3.9mm$$

$$\Theta_{MAX} = arcsin(\frac{3.9x10^{-3}m}{2x2.1x10^{-3}m}) = 68.2°$$

2.1.11 Angle FFT

The same technique used for the Doppler FFT can be used when there are two or more objects in front of the radar at the same range and relative velocity. In the 2D-FFT there is a single peak at the same location for both antennas, and the signal at the peak will have contributions from phasor corresponding to the objects. So the simple phase contribution that was explained for the single object does not work properly.

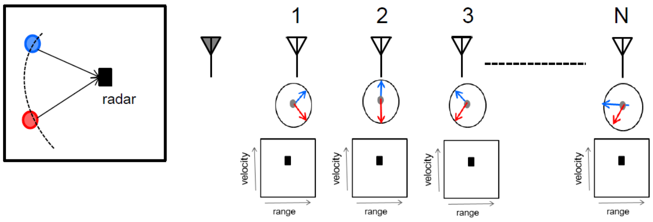


Figure 2.20: AoA FFT

The solution of this problem is analogue to what was done for the speed, so have several RX antennas as shown in figure 2.20. In this way the 2D-FFT has a peak in the same location, and the signal at this series of peaks is going to create a discrete sequence consisting to the two phasor related to the two objects.

FFT of these phasor on the sequence show up as two distinct peak ω_1 and ω_2, where ω_1,ω_2 are the rates of rotation in radians per sample for the two objects. So, the AoA of the two objects can be calculated using the formula 2.14 in order to obtain the frequency components.

2.1.12 Angle resolution

Radar angular resolution is the minimum distance between two equally large targets at the same range which radar is able to distinguish and separate to each other.

Supposing to have two objects at angle Θ and $\Theta + \Delta\Theta$ relative to the radar. The angle resolution is the ability of the radar to identify the two objects, separated by an angle of $\Delta\Theta$ as separate objects and process them as two distinct peaks in the frequency spectrum. The angle resolution can be estimated by:

$$\Theta_{res} = \frac{\lambda}{Ndcos\Theta} \tag{2.15}$$

Where λ is the wavelenght, N is the number of RX antennas, d is the distance between two consecutive RX antennas, and Θ is the angle of arrival. It is important to note that the resolution depends on the distance between two RX antennas, but this in the radar is fixed, like the number of RX antennas. So the angle resolutions depends only on the AoA.

In particular with the increase of theta between 0° and 90°, cos(Θ) decreases and so Θ_{Res} decreases:

$$\Theta \uparrow cos(\Theta) \downarrow \Theta_{Res} \uparrow$$

2.2 Adaptive Cruise Control

The Cruise Control (CC) is one of the common automotive Drive Assistance System (DAS). The driver sets a reference velocity and the car tends to maintain this velocity under the influence of external loads, such road slope and wind. The aim of the DAS implemented on the car, is to help the driver to maintain the control of the car. These control systems are based on the information related to the car, like acceleration, velocity and so on. In the last years these control system have been upgraded in Advanced Driver Assistance System (ADAS) which are born to improve the driving comfort and help and support the driver in critical traffic scenario. Another important feature of these system is to prevent a possible accident situations. The main reason these devices were created is the statistical consequence of road accidents. In 2002, approximately 6,977 people in Germany, 7,720 people in France, 3,450 in UK and 6,682 in Italy died due to traffic accidents. [6]. Consequently the automotive sectors introduced different driving systems which can help the drivers in possible dangerous scenarios. These system are based on information from the external environment. The Adaptive Cruise Control (ACC) is an ADAS system, capable of regulate the car's speed on the basis of the traffic conditions. The main feature of the ACC is to maintain a safety distance from the vehicle located in front without any driver interventions. To do this the system needs several information on the environment that surround our car. These information come from the different sensors mounted on the cars like radar, stereo camera an lidar. The control system can be activated directly from the driver when it prefers. When the system is activated the actual speeds becomes the cruising speed which the vehicle tends to maintain. This speed is then subject to a possible autonomous variation based on the information imposed of the traffic conditions coming from the sensors. For example if the car located in front slows down, the ACC reduce the velocity of the car to maintain the safety distance. Primarily to be a safety system the ACC is marked as a comfort system. Since the braking capacity is usually limited to - 3 m/s^2 the ACC can not be regarded as collision avoidance system. Collision avoidance systems need higher deceleration, in the order of -8 m/s^2. In emergency situations the driver must remains fully responsible for the vehicle manoeuvring [7]. By means of smoother accelerations and deceleration profiles the system contributes to a more stable traffic flow which increase roadway capacity [8]. The activation and the deactivation of the control system is commonly commanded with appropriate buttons on the steering wheel.The driver thanks the suitable buttons is able to set the distance to keep from the vehicle located in front of it (lead vehicle) and the speed that he wants to maintain.

The working principle of the ACC is directly correlated to the sensor's information. The main sensor on which the ADAS system is based is the radar. This sensor on the road cars is commonly located under the bumper. As mentioned it is able to detect when one vehicle is in the same lane and gives the information on the relative distance and relative velocity of the car located in front of it. On the basis of these information the control system regulate the velocity of our car (Host/Ego vehicle) in order to maintain a safety distance from the vehicle in front of us (Lead vehicle).

In Figure 2.21 on the left vehicle is the host or ego vehicle equipped with the ACC, on the right there is the lead or target vehicle.

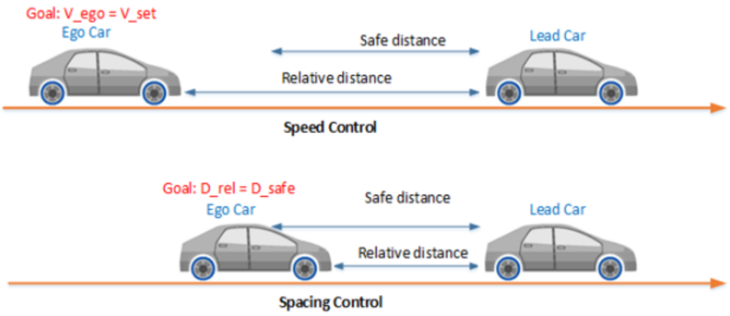


Figure 2.21: Adaptive Cruise control

The relative distance d_r between the lead vehicle and the ego vehicle is defied as

$$d_r = x_l - x_h \tag{2.16}$$

where x_l i the position of the lead vehicle, and x_h is the position of the host vehicle.

The relative velocity is the velocity between the lead vehicle v_l and the host vehicle v_h.

$$v_r = v_l - v_h \tag{2.17}$$

The driver can over-ride the system in any time by a pressure on the braking or acceleration pedal.

The conventional ACC system work normally with velocity which are greater than 30 km/h in a highway. The Stop-and-go functionality is a feature of the ACC which has been implemented to improve the driver comfort also for velocity below 30 km/h in urban scenario. The ACC stop-and-go is able to follow the lead vehicle and automatically stops the car if it stops. ACC stop-and-go automatically accelerate the host vehicle when the target vehicle moves again.

2.2.1 General Architecture of the ACC

The ACC is an extension of the Cruise Control. The ACC architecture can be divided into two states: speed control and spacing control. Speed control is a traditional cruise control which aim is to maintain the velocity set by the driver. The spacing control instead, must regulate the longitudinal behaviour of the car according to the traffic condition. The longitudinal behaviour of the ACC is typically hierarchical with an upper-level control/model and lower-level control/model [9]. The upper level model receive as input the actual longitudinal velocity of the car, and based on the information received by the sensors (Radar in our case) impose the desired acceleration to the plant. The lower level control instead realizes the actual longitudinal control of the vehicle based on the acceleration received by the upper level control.

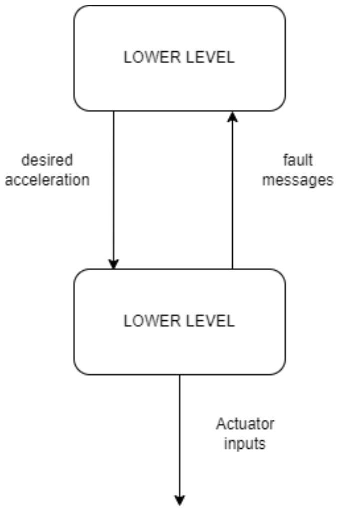


Figure 2.22: Structure of ACC control

2.3 Features and Requirement

The ACC is activated directly by the driver when he prefers. For this reason it is possible to divide the control system in two states:

- **Manual Mode:** The ACC system is not activated. The behaviour of the car like speed, acceleration and deceleration are directly controlled by the driver with the appropriate pedal, without any intervention of the ADAS.
- **ACC mode:** The activation of the ACC system is done directly by the driver with an appropriate button located on the steering wheel. The system allows the car to maintain the speed and distance set by the pilot from the car located in front, without any intervention of the driver on the throttle or braking pedal. The driver first of all must activate the ACC subsequently he can set the desired speed and safety distance to keep. There are also a buttons on the controller able to increase and decrease the initial cruising velocity.

The ACC has different customers features with its implementation on the road cars, which are figured in Figure 2.23

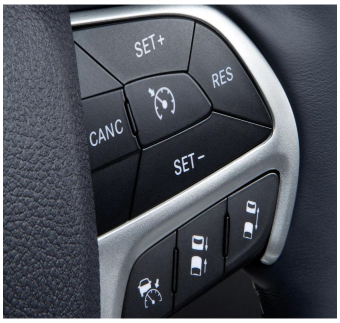


Figure 2.23: ACC controller on the steering wheel

- **Activation/deactivation ACC:** The ACC is activated only when the driver press the appropriate button on the steering wheel but its deactivation can occurs with different factors. The deactivation of the driver assistance happen when the driver turn off the system through the button or when the driver exerts some pressure on the brake pedal. The latter condition is implemented in order to guarantee the driver to have the entire control of the vehicle in case of an emergency situation which is not recognized by the autonomous system.
- **Movement:** The acceleration or deceleration of the car can occurs in a two standard situations: either when the driver press the throttle pedal whit the ACC disable or when the ACC wants to accelerate. However when the ACC system is activated the acceleration and deceleration can occurs also in different situation like:
  - The driver presses the appropriate button to increase or decrease cruising speed (SET + or SET -). The speed can be increased by 1 km/h with a

single click by the driver or by 10 km/h with a long press of the button. (Acceleration/Deceleration)

- The driver wants to overtake another car, and so he wants to increase the speed of the car. In this situation the ACC is temporally "frozen" until the velocity imposed by the throttle pedal is equal or lower than the cruising velocity imposed. (Acceleration)
- The driver wants to increase or decrease the safety distance. If the driver press the "- button" the distance decreased and consequently the speed of the ego vehicle will be increased. If the driver press the "+ button" the distance increased and so the speed of the host vehicle decrease. (Acceleration/Deceleration)

Chapter 3

Chirp programming and configuration

3.1 Chirp structure

It is one of the most relative aspect that determine the overall detecting working principle of the radar, defining the configurations and several parameters of the chirp. With the modification of the chirp's parameters the radar can work in different ways, in particular it can works as Short Range Radar (SRR), Medium Range Radar (MRR), Long Range Raadar (LRR)... etc. One way to have the maximum flexibility by the radar is to programming the radar as Monolithic Microwave Integrated Circuit (MMIC), in which the radar sends several chirps with different parameters. Each chirp is responsible to detect different object located closed or far away to the radar.

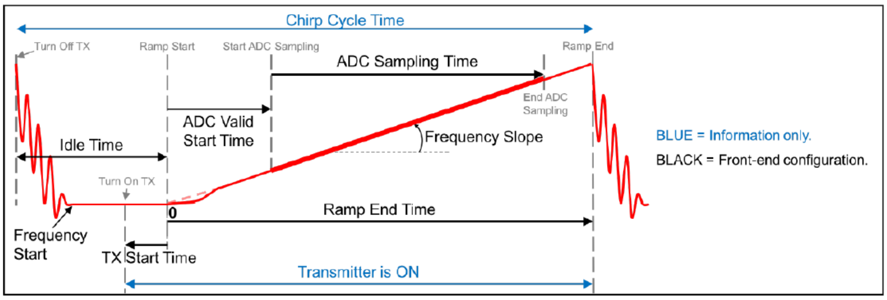


Figure 3.1: Typical FMWC chirp signal

Figure 3.1 represent a single chirp with the several associated timing parameters.

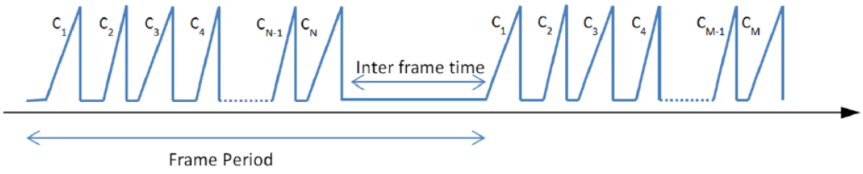


Figure 3.2: Typical Frame structure

Figure 3.2 instead, shows frame structure that consists of a series of chirps followed by inter frame time. This represents 'Fast FMCW' modulation, where each chirp is typically 10's of μs in duration.

3.1.1 Chirp Parameters

The radar sends a Chirp and when it meets an object it is reflected back, and received by the RX antennas. However the performance of the radar and the correct detection of the object located in the FOV of the radar depends on several parameters that can be tuned. The most important configurable parameters of chirps are:

- **Stat Frequency:** It is expressed in [Hz], and it represents the frequency at which the radar starts to transmit the signal. For the AWR1642BOOST the maximum bandwidth is up to 4GHZ within [77-81] GHz.
- **Frequency slope** It is represented in [MHμs] and it defines the speed at which the chirp sweeps within the frequency band. The AWR1642BOOST support until 100 MHz/μs. This parameter has a direct impact on the maximm detected range, and indirect impact on the speed parameters. In particular with the decreasing of the frequency slope, the maximum range increases.

$$\downarrow f_{slope} \Rightarrow \uparrow range_{MAX}$$

- **Idle Time** It is expressed in [μs] and defines the amount of time that passes between two consecutive chirps sent by the radar, it has a direcct impact on the speed parameters.

$$\downarrow idle_{Time} \Rightarrow \uparrow Speed_{MAX} \Rightarrow \uparrow Speed_{Res}$$

- **ADC start time** It is expressed in μs, and it is the time at which the ramp begin when the ADC starting to sample the data, it is directly correlated to the velocity parameters.

$$\downarrow ADC_{StartTime} \Rightarrow \uparrow Speed_{MAX} \Rightarrow \uparrow Speed_{Res}$$

- **Ramp end time** it is expressed in chirp suration and so in μs, it the time that passes from when the ramp starts until it end. It is a critical parameters because it influence both range and speed parameters

$$\downarrow Ramp_{end} \Rightarrow \uparrow Speed_{parameter} \Rightarrow \downarrow Range_{parameter}$$

3.2 Range parameters

The application of the radar in the last year is very common in the automotive domain. In order to guarantee the correct behaviour of the autonomous system mounted on the car, there are two main parameters that are very important, the distance at which the radar is able to detect an object and the range resolution of the radar. However these parameters are inversely proportional, therefore the increase of one parameter involve the decrease of the other.
There is another important factor that can influence the detection of the object by the radar, it is the amount of energy that the object is able to reflect back to the radar. This factor is called the Radar Cross Section, RCS. However there is not a formula able to calculate the RCS, because it depends on many factor, such that surface, orientation, nature of the material, geometry and in some cases it can depends on the frequency. For this reason the calculation of this parameter requires measurements lab and tools. Table 3.1 show some typical value of RCS for some objects.

Detected Object	**RCS** $[m^2]$
Truck	100
Car	5
Motorcycle	3.2
Adult	1
Child	0.5

Table 3.1: RCS value for typical object

Nevertheless higher is the value of RCS and better is the detection of the object by the radar.

3.2.1 Radar range equation

In Chapter 2 have been discussed the main characteristic of the working principle of the radar.

As it was mentioned there is an important factor which determines the maximum distance, and it is the reflected chirp by the object on the radar, this must be sufficient strength to be detectable by the radar, and it depends on other factors like:

- P_t: Outoput power of the device
- $G_{TX/RX}$: TX and RX Antenna Gain
- σ: Radar cross section of thr target (RCS)
- A_{RX}: Effective aperture area of RX antenna
- SNR: It is the signal to noise ratio

All these parameters are grouped in the Radar Range Equation (RRE) whose equation is the following:

$$Range_{MAX} = \sqrt[4]{\frac{P_t \times G_{RX} \times G_{TX} \times c^2 \times \sigma \times NT_c}{f_c^2 \times (4\pi)^3 \times KT \times NF \times SNR_{min}}} \tag{3.1}$$

As it possible to see the equation 3.1 has many mathematics calculation, for this reason in order to determine the maximum range is better to use 2.4. However in the following table are expressed some parameters that are present in 3.1.

Symbol	Parameter involved	Value
P_t	TX output power	12 dBm
G_{RX}	RX gain	24dB
G_{TX}	TX gain	24dB
σ	Radar Cross Section	5 m^2 for car
NT_c	total frame time	6 ms
f_c^2	frequency start	77 GHz
KT	Boltzman constant	4.11×10^{-21} Joule at 298 K
NF	noise figure	15 dB
SNR_{min}	minimum SNR programmed	15 dB

Table 3.2: Radar Range equations parameters

3.2.2 Speed parameters

The speed resolution depends on how many chirps are transmitter for each single frame. The frame period is expressed as the number of transmitted chirp multiplied by the chirp period. The speed resolution increase with the increasing of the frame period. It is possible to distinguish between two frame period, the firs one that includes the idle time and the second one can be denominate as the active frame period which does not contain the idel time.

3.3 Chirp Configuration

The AWR1642 radar allow the user to configure the chirp setting, such as, slope, start frequency, idle time... etc according to the radar application field. It is possible to program the radar for four different chirp profiles, and up to 512 unique chirps can be pre-programmed and stored in the chirp configuration RAM. Each chirp definition entry in the RAM can belong to one of the four profiles. A frame is the sequence of chirp that are created in the chirp configuration RAM which can be looped up to 255 times.

3.3.1 Multi-Mode Radar Application

The AWR1642 allow an "Advanced frame configuration" which allow a large flexibility on the radar, in fact with this implementation is possible to have multiple chirp configuration in a single frame. The main frame can be constituted using a sequence of sub-frames (up to 4) which constituting different radar mode. Each sub-frame it is composed by multiple burst of chirps (up to 512 bursts), and each burst can consist of 512 unique chirps which are associated to one of the 4 profile. A set of bursts in a sub-frame can be further looped in software up to 64 times.

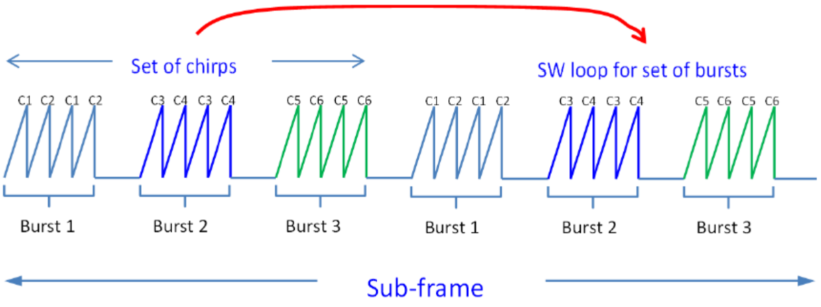


Figure 3.3: Frame structure in Multi mode application

The frame is formed by a maximum of 4 sub-frames that can have a different set of chirps. Each set of chirp can be used for each antenna that transmit the signal. Figure 3.3 shows an example of the different chirp profiles and sub frames.

However in order to have this kind of configuration, there some timing restriction that must be respected:

- Inter-burst time $\geq 50\ \mu s$
- Inter sub-frame $\geq 100\ \mu s$
- Inter frame $\geq 200\ \mu s$

The table 3.3 shows an advanced profile that contains two sub-frames able to detect objects up to 20 meters with an high resolution and up to 80 m with a less resolution.

Parameter	Units	Profile 1	Profile 2
Max unambiguous range	m	45	22.5
Sweep bandwidth	MHz	750	1500
Ramp slope	MHz/μs	15	30
Inter-chirp duration	μs	12	50
Numbers of chirps	-	128	128
Range resolution	m	0.20	0.1
Chirp duration	μs	50	50
Max unambiguous relative velocity	kmph	56.56	35.3
Max beat frequency	MHz	4.5	4.5
ADC sampling rate	Msps	5	5
Number of sample per chirp	-	250	250
Range FFT size	-	256	256
Frame time (total)	ms	7,94	12.8
Frame time (active)	ms	6.4	6.4
Radar data memory	KB	512	512

Table 3.3: Advanced Chirp Configuration

3.4 SSR - USRR configuration

As has been mentioned it is possible to have different configuration of the radar based on its application. For this study the radar was used as short range radar (SRR) and as Ultra Short Range Radar (USRR). In this chapter will be explained the configuration and the mathematical formulas that are used as configuration of the radar.
The AWR1642 is configured for short range application, it is able to detect as many as 200 objects up to a distance of 80 m and able to track up to 24 of them as fast 90 km/h. Thanks to the ability of the radar to be configured in the Multi Mode, a second profile of chirp is sent by the radar, that is able to detect object up to 20 m with an high range resolution, so that approaching vehicles and close small objects can be detected by the radar. The table 3.4 shows the two operation mode of the radar.

The AWR1642 has four RX antennas and two TX antennas, that are able to be set in two different chirp configuration. The first configuration SRR, uses a non multiple input and a multiple-output (MIMO) configuration, with only one

Parameter	**Values**
Maximum range	80 m (SRR), 20 m (USRR)
Range Resolution	36.6 cm (SRR), 4.3 cm (USRR)
Maximum velocity	90 km/h (SRR), 36 km/h (USRR)
Velocity resolution	0.52 m/s (SRR) 0.32 m/s (USRR)

Table 3.4: System specification

TX antenna. The second configuration, instead, the USRR uses a time division multiplex MIMO configuration, which alternate chirps in a frame transmit on TX1 and TX2 respectively.

3.4.1 USRR Profile (20 m)

This profile is used to detect objects up to 20 m that are in front of the radar. This sub frames are composed by two alternating chirps, which utilizes one of the available TX antenna present on the radar. The processing of this sub frames is able to guarantee a very high range resolution of 4.3 cm, with an angular resolution approximately to 14.3° and a maximum speed that is around 18 km/h.

The profile, the chirp and the frame parameters are the main parameters for the obstacle detection. Must be take high care in the interaction between these factors, and the radar must be programmed in the correct way, in order to have a good match between frame, profile and chirp, otherwise the correct detection of the objects does not happen. In the table 3.5 are shown the Profile parameter, instead in the table 3.6 are expressed the values of the chirp parameters.

Profile ID	1
Sampling rate	6222 Ksamp/s
Number of ADC sample	512
Start frequency	77 GHz
Frequency slope	44 MHz/s
Idle time	7 μs
Ramp end time	87.28 μs
TX start time	1 μs
ADC start time	4.8 μs

Table 3.5: USRR Profile parameters

	Chirp0	**Chirp1**
Profile ID	1	1
Start Index	128	129
End Index	128	129
Start Frequency variation	0	0
Frequency slope variation	0	0
Idle Time variation	0	0
ADC start time variation	0	0
TX antenna	1	2

Table 3.6: USRR frame parameters

Start Index	128
End index	129
Loop count	32
Periodicity	30 ms
Number of real ADC samples	1024
Number of complex ADC samples	512
Chirp 0 number	32
Chirp 1 number	32
Total chirp number	64
Transmitter number	2
Number of angle bin	32

Table 3.7: USRR sub frame parameters

Now, from the parameters available in the tables above and the formula expressed in Chapter 2 it is possible to calculate, mathematically, the main characteristic of the radar, and so the range, range resolution, maximum identifiable speed and so on.

- **Maximum range:** From the equations 2.4 and 2.5 it is possible to determine the maximum range

$$Range_{MAX} = \frac{IF_{MAX} \times c}{2S} = \frac{0.9 \times SamplingRate \times c}{2S} \tag{3.2}$$

$$Range_{MAX} = \frac{0.9 \times 6222 \times 10^3 \times 3 \times 10^8}{2 \times \frac{42 \times 10^6}{10^{-6}}}$$

$$Range_{MAX} = 19.99m$$

It is important to remember that S is the Slope that is given by the Bandwidth divided by the total chirp period, as expressed in the following formula:

$$S = \frac{B}{T_c} \tag{3.3}$$

Knowing the maximum number of ADC sample, T_c can be reformulated in the following way:

$$ADCSample = T_c \times SamplingRate \tag{3.4}$$

$$T_c = \frac{ADCSample}{SamplingRate}$$

- **Range resolution:**
  From the formula and the reformulation of the Total chirp period, expressed in 3.4 it is possible to determine the Range resolution:

$$Range_{RES} = \frac{c \times SamplingRate}{2S \times ADCSample} \tag{3.5}$$

$$Range_{RES} = \frac{c \times SamplingRate}{2000 \times Slope \times ADCSample}$$

$$Range_{RES} = \frac{3 \times 10^8 \times 6222 \times 10^6}{2000\frac{42 \times 10^6}{10^{-6}}}$$

$$Range_{RES} = 0.043m = 4.3cm$$

- **Period of repetition of chirps:**

$$T_{chirp} = ChirpIdleTime + ProfileIdleTime + RampEndTime \tag{3.6}$$

$$T_{chirp0} = 0 + 7 + 87.28 = 94.28\mu s$$

$$T_{chirp1} = 0 + 7 + 87.28 = 94.28\mu s$$

- **Speed resolution:** From the formula 2.11 it is possible to derive the following one:

$$Speed_{RES} = \frac{1000}{T_{chirp0} + T_{chirp1}} \times \frac{1}{NumberChirp} \times \frac{\lambda}{2} \tag{3.7}$$

  Where the NumberChirp is expressed as:

$$NumberChirp = (EndIndex - StartIndex + 1) \times LoopNumber$$

$$Number_{chirp0} = (128 - 128 + 1) \times 32 = 32$$

$$Number_{chirp1} = (129 - 129 + 1) \times 32 = 32$$

λ is expressed as:

$$\lambda = \frac{SpeedLight}{StartFrequency} = \frac{3 \times 10^8}{77 \times 10^9}$$

$$\lambda = 3.9 \times 10^{-3} = 3.9mm$$

Substituting all in the formula in 3.7 it is possible to obtain:

$$Speed_{RES} = \frac{1000}{94.28 + 94.28} \times \frac{1}{32} \times \frac{3.9}{2}$$

$$Speed_{RESchirp0} = 0.32m/s = 1.15km/h$$

$$Speed_{RESchirp0} = 0.32m/s = 1.15km/h$$

- **Maximum Speed:** Based on the equation 2.10 it is possible to reformulate the formula related to the maximum speed:

  $$Speed_{MAX} = \frac{\lambda}{4T_c}$$

  Multiplying and dividing the above formula by N it is possible to express the maximum speed in function of the speed resolution. and so obtain:

  $$Speed_{MAX} = \frac{Speed_{RES} \times NumberChirp}{2} \tag{3.8}$$

  It is possible to calculate the Maximum speed for both the chirp with the related parameters, and so obrain that:

  $$Speed_{MAXchirp0} = \frac{0.32 \times 32}{2}$$

  $$Speed_{MAXchirp1} = \frac{0.32 \times 32}{2}$$

  $$Speed_{MAXchirp0} = 5.16m/s = 18.6km/h$$

  $$Speed_{MAXchirp0} = 5.16m/s = 18.6km/h$$

3.4.2 SRR Profile (80 m)

The SRR configuration it is almost similar to the USRR, however here change the number of chirp which have an idle time different in the first 64 chirps with respect the last 64. In particular the chirps between 0 and 63 have an idle time of 3 μs, while the chirps between 64 and 127 have an idle time of 14.8 μs. As in the previous chapter the table 3.8 and 3.9 shown the frame and chirp parameters related to this application way.

Profile ID	0
Sampling rate	5000 Ksamp/s
Number of ADC sample	256
Start frequency	76 GHz
Frequency slope	8 MHz/s
Idle time	5 μs
Ramp end time	56 μs
TX start time	1 μs
ADC start time	4.8 μs

Table 3.8: USRR Profile parameters

	Chirp0	**Chirp1**
Profile ID	0	0
Start Index	0	64
End Index	63	127
Start Frequency variation	0	0
Frequency slope variation	0	0
Idle Time variation	0	11.8
ADC start time variation	0	0
TX antenna	1	1

Table 3.9: USRR frame parameters

Start Index	0
End index	127
Loop count	1
Periodicity	30 ms
Number of real ADC samples	512
Number of complex ADC samples	256
Chirp 0 number	64
Chirp 1 number	64
Total chirp number	128
Transmitter number	1
Number of angle bin	32

Table 3.10: USRR sub frame parameters

- **Maximum Range:** From 3.2 it is possible to obtain the maximum range:

$$Range_{MAX} = \frac{0.9 \times 5000 \times 10^3 \times 3 \times 10^8}{2000 \times \frac{8 \times 10^6}{10^{-6}}}$$

$$Range_{MAX} = 84m$$

- **Range resolution:** From 3.7 it is possible to obtain:

$$Range_{RES} = \frac{3 \times 10^8 \times 5000 \times 10^6}{2000 \frac{8 \times 10^6}{10^{-6}}}$$

$$Range_{RES} = 0.36m = 36cm$$

- **Period of repetition of chirp:** Based on equation 3.6 is is possible to calculate the period of repetition of the two chirps. As it possible to see from the following calculation the Repetition chirp of Chirp 1 is longer than Chirp 0, this is due to the fact that the ChirpIdleTime of Chirp 1 is different from zero differently from Chirp 0:

$$T_{chirp0} = 0 + 5 + 56 = 61\mu s$$

$$T_{chirp1} = 11.8 + 5 + 56 = 72.8\mu s$$

- **Speed Resolution:** Base on 3.7 we have:

$$Number_{chirp0} = (128 - 128 + 1) \times 32 = 32$$

$$Number_{chirp1} = (129 - 129 + 1) \times 32 = 32$$

$$Speed_{RES} = \frac{1000}{61} \times \frac{1}{64} \times \frac{3.9}{2}$$

$$Speed_{RESchirp0} = 0.49m/s = 1.76km/h$$

$$Speed_{RESchirp0} = 0.49m/s = 1.47km/h$$

- **Maximum Speed:** From 3.8 it is possible to obtain the maximum speed of the object, located at a distance between 20 m and 80 which are detectable by the radar. However it is important to underline that the maximum speed calculated is less than the real one identifiable, this is due to the Chinese reminder Theorem which give the real maxim speed that can be estimated by the radar, that is 90 km/h.

$$Speed_{MAXchirp0} = \frac{0.49 \times 64}{2}$$

$$Speed_{MAXchirp0} = 15.7m/s = 56.5km/h$$

$$Speed_{MAXchirp0} = 13.1m/s = 47.1km/h$$

Chapter 4

Implementation

4.1 Hardware

The device used for this thesis as mentioned in the Introduction is the AWR1642ODS Evaluation Mode (EVM) radar produced by Texas Instruments (TI). It is a FMWC radar able to operate in the 76-to-81 GHz band. Due to the presence of two programmable processors, it is responsible for the radio configuration, control and calibration. Programming model changes can enable a wide variety of sensor implementation as Short, Medium or Long.

In the front view of the radar the device includes:

- **Power Connector:** The Radar can be powered by an AC/DC 5V power supply (2.1 mm jack) with a current limit of 5A. As soon as the radar is powered the 5V led and the NRST led turn on, indicating that the device is powered on.

Figure 4.1: Power connector

- **Micro USB connector:** There is a micro USB connector that can be used in order to flashing on the device the the desired configuration and to extrapolate the information captured by the radar.

- **on-board antennas:**
  on-board antennas: used for the four receivers and two transmitters that allow you to track multiple objects with their distance and angle information.

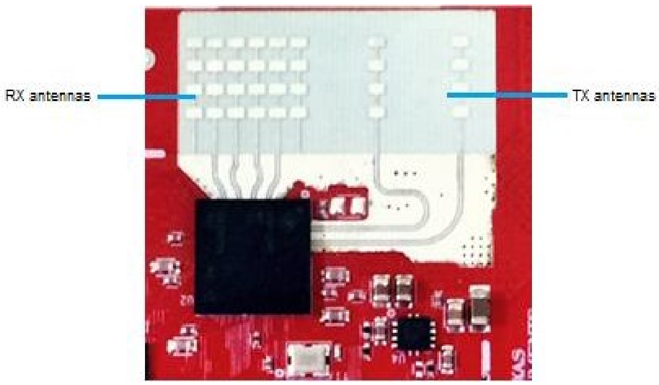


Figure 4.2: RX and TX antennas

- **60-pin connector:**
  which provides the high speed LVDS data, control signals and JTAG debug signal. This pin can be used to connect the radar with a Data capture Board (DCA1000), in order to establish an ethernet connection with the Computer. This could be very useful to store the information of an instance of the captured stream by the radar, and analyzed it, in order to understand what the radar see at a detirmined time instance and tune the radar to optimize the object detections.

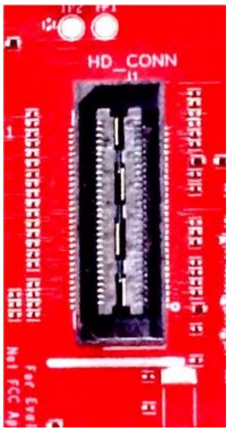


Figure 4.3: 60-pin connector

- **CAN Interface connector:**

The Connector provide a CAN-L and CAN-H signals from the onboard CAN-FD transceiver. This signal can be directly wired to the CAN bus.

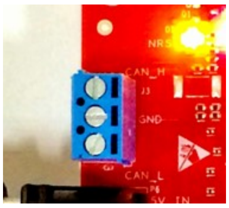


Figure 4.4: CAN Connectors

- **Sense-on.Power(SOP) jumpers:**
  The AWR 1642 device can be set to operte in three different modes based on the state of the SOP lines. A close jumpers refers to 1 instead an open jumpers refers to a 0 state of the SOP signal.

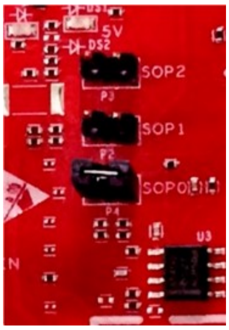


Figure 4.5: SOP jumpers

Reference	State
SOP2	1-0-1 = flash programming
SOP1	0-0-1 = functional mode
SOP0	0-1-1 = debug mode

Table 4.1: SOP state information

- **Push button, LEDs and switch:**
  It is possible to note several push buttons and LEDs on the front view of the device, the following 4.2 explain their functionality;

Reference	Comments
SW2	It is used to RESET the AWR1642
SW1	When it is pulled the GPIO1 is pulled to V_cc
DS2	This LED indicates the presence of the 5V supply
DS4	It is used to indicate the state of the nRESET
DS1	This LED glows if there are any Hardware error on the device
DS3	Glows when the GPIO is logic 1

Table 4.2: Switch and LED information

Another important switch is the SPI-CAN switch. On the device the SPI and CAN communication are muxed together, so through this pin it is possible to select between the 20-pin connector present in the rer view oh the radar or if enable the CAN-FD output.

In the rear view of the AWR device instead there are:

- **20 pin Boosterpack connector:**
  This connector has the standard TI launchpad connectors which allows a direct interface with all the TI's Launchpad. In particular there are two 20-pin connectors, which functionality is expressed in Table 4.3 and Table 4.4.

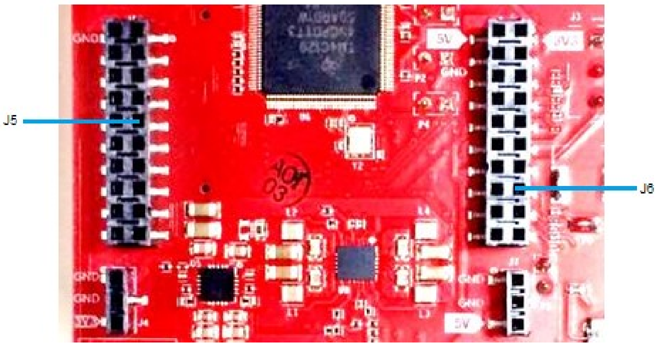


Figure 4.6: 20-pin BoosterPack connectors

Pin Number	Description	Pin Number	Description
1	NERROUT	2	GND
3	NERRIN	4	DSS LOGGER
5	MCUCLK OUT	6	SPI-CS
7	NC	8	GPIO01
9	MSS LOGGER	10	nRESET
11	WARMST	12	SPI-MOSI
13	BSS LOGGER	14	SPI-MISO
15	SOP2	16	HOSTINT
17	SOP1	18	GPIO02
19	SOP0	20	NC

Table 4.3: J5 Connector Pin

Pin Number	Description	Pin Number	Description
1	3V3	2	5 V
3	NC	4	GND
5	RS232RX	6	ANA1
7	RS232TX	8	ANA2
9	SYNC IN	10	ANA3
11	NC	12	ANA4
13	SPI CLK	14	PGOD
15	GPIO0	16	PMIC ENABLE
17	SCL	18	SYNC OUT
19	SDA	20	PMIC CLK OUT

Table 4.4: J6 Connector Pin

4.2 PC connection and communication

As has been anticipated in the hardware explanation of the device there is a micro USB connector on the radar which can be used to setup the radar in the desired configuration, (short, medium or long), and to receive the information captured by the sensor on the environment surrounding it. There are several methods to receive the information sent by the radar and several tools provided by TI that will be explained in the following chapters. In order to have a correct communication between the radar and the computer, the device must be powered on by an external power supply at 5 V 3 or 4 A, then the radar must be connected through the USB-micro USB cable on the PC.

4.2.1 Firmware installation

The device is provided by TI without any firmware inside, however there are several configurations that the manufacturer provided.
One of the standard configuration which are provided by TI is present in the SDK tool that is an application software which contains several C or ASM code which enable application, evaluation and development on TI mmWave sensors. The tools furthermore provides all the drivers that are needed for a correct identification of the radar on the computer. After the correct synchronization between the PC and the radar, in order to flash the code available on the tool has been used another program, UNIFLASH. It is important to underline that the radar must be set in the flashing programming mode, so with both SOP0 and SOP2 jumpers connected, as explained in table 4.1. Subsequently in the Program tab, must be browse and locate the images (.bin file) available by the SDK in this case.
The bin file is a build result of the .c and .h file that are needed to program the two processors (DSS and MSS) available on the AWR1642 device.

4.2.2 Matlab Tool

After the installation of the firmware inside the device, the radar is now able to send and receive the chirp configuration desired, and so to detect the objects located in its FoV, and to send the information about them.
In order to have a good interpretation of the data sent by the radar, one of the tool provided by TI is able to elaborate in Real Time the data sent by the device through the micro USB connector.
The tool is implemented on a MATLAB script and is able to graphically represent various useful information about the objects located in the radar FOV.

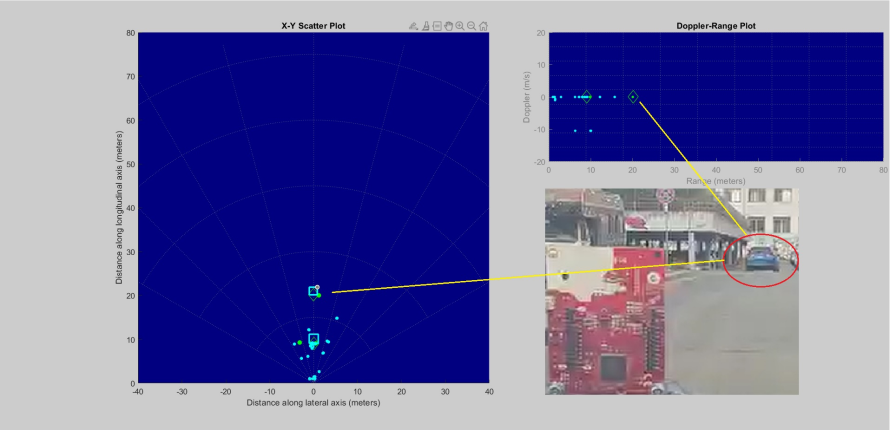


Figure 4.7: Test results from Matlab tool

On the tool there are several important information about the detected objects:

- A X-Y Scatter plot, where are represented:
    - The USRR objects.
    - The SRR objets
    - The cluster objects
    - The tracked objects
- A Doppler range plot
    - The plot indicates the velocity of the tracked objects.

This tool is very useful to understand what the radar sends but it is not able to save the information in variable that must be manage in order to tune the ACC. For this reason, an alternative method for the data acquisition has been implemented on a Simulink Environment.

4.3 Simulink implementation

One of the strengths of the ACC is to elaborate the Real Time data sent by the radar, and on the base of the information about the objects located in front of the sensor, moderate the velocity of the vehicle.
For the study on what the radar sends, and for the acquisition of the information, the radar has been connected to the computer through the micro USB communication. In order to elaborate the data, has been used a SIMULINK project in which has been extrapolate the useful information like position and velocity of the objects located in front of the device. In this study the configuration used to detect the objects located in the FoV of the radar is the one discussed in Chapter 2.

4.3.1 Data Format

The most important process to configure the ACC is to understand what the radar sends, and in which way the x-y-z axis is located in front of the radar. The center of the axes is located approximately between the last Rx antenna and the first TX antenna. The x-axis is positive along the right with a transverse orientation to the radar, and the y-axis has a longitudinal direction with respect to the radar.

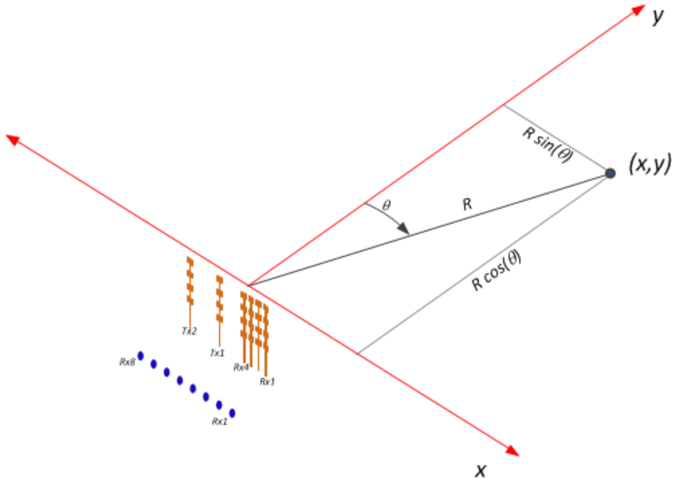


Figure 4.8: Coordinate location in front of the radar

The data format of the radar is set on the mss and dss processor and so on the firmware that is installed on the board. For the Short Range Lab the data format is a Type-Lenght-Value (TLV) used with a Little Endian byte order. This kind of data information is composed by a Frame Header and then a variable number of TLVs which depends on the number of objects detected in the scene. Each TLV item consists of a type, length and payload information. The structure of the packet is illustrated in the following figure.

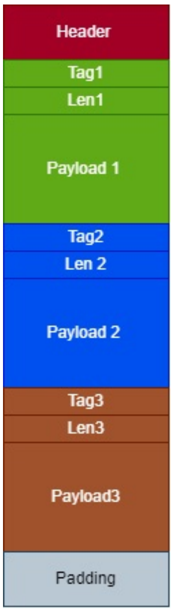


Figure 4.9: TLV data structure

Since as has been mentioned the TLVs depends on the number of objects, the length of the payload information vary frame to frame.

The Data format is so in this way expressed:

- **Frame Header:**

Parameter	**Size**
Header	8 bytes
Version	4 bytes
Packet len	4 bytes
Platform	4 bytes
Frame Number	4 bytes
Time CPU cycles	4 bytes
Number detected Objects	4 bytes
Number TLV	4 bytes
Subframes number	4 bytes

Table 4.5: Frame Header structure

- **Detected object output format:**
  The information about the detected object have a a first initialization part where are expressed the type of TLV and the length of the following TLV. Subsequently there are the information about the object detected.

Parameter	**Size**
TLV Type	4 bytes
Length	4 bytes
Number Objects	2 bytes
XYZQformat	2 bytes
Doppler	2 bytes
Peakk Value	2 bytes
x	2 bytes
y	2 bytes

Table 4.6: Detected Object output format

- **Cluster output format:**
  As in the detected output format also in this case there is an initialization part where is expressed the type o TLV and the length of the information, and a second part where the information are directly expressed.

Parameter	Size
TLV Type	4 bytes
Length	4 bytes
Number Objects	2 bytes
XYZQformat	2 bytes
Cluster center on x direction	2 bytes
Cluster Center on y direction	2 bytes
Cluster size on x direction	2 bytes
Cluster size on y drection	2 bytes

Table 4.7: Cluster output data format

- **Tracking Output Format:**
  The data format is structured in the same way of the previous cases:

Parameter	Size
TLV Type	4 bytes
Length	4 bytes
Number Objects	2 bytes
XYZQformat	2 bytes
Tracking X co-ordinate	2 bytes
Tracking Y co-ordinate	2 bytes
Tracking X velocity	2 bytes
Tracking Y velocity	2 bytes
Tracking size on X direction	2 bytes
Tracking size on Y direction	2 bytes

Table 4.8: Tracking output data format

As it possible to see there are several methods in which the radar identify the objects in front of it. The simplest is the identification as single points for both the USRR and SRR detection. The radar is also able to make a clustering of the detected object. This is performed directly by the device thanks the firmware installed on it. It is performed using a dBscan algorithm which is applied on both USRR and SRR subframes object detection. The output of the clustering as it possible to see from Table 4.7 is composed by the mean location of the cluster and its dimension. For the USRR subframes, the cluster is a grouping of dense point clouds to rectangle. This cluster can be used for the vehicles or motorcycle that crossing the FoV of the radar.
Instead for the SRR algorithm, the clustering output is the basis for the input

to the tracking algorithm. The strongest object in the cluster is provided as the representative object to the tracking algorithm.
The tracker is based on a basic Extended Klman filter with four states, in particular, transversal position (x), longitudinal position (y), transversal velocity (v_x), and longitudinal velocity (v_y).
From the tables above, 4.6, 4.7 and 4.8 it is possible to see that the information sends by the radar differentiate on the basis on the type of identification of the objects. The structure of the payload information is based on the TLV, in particular there are three type of TLV in this firmware installed on the board.

TLV Name	Type
Detected points	1
Clusters	2
Tracking	3

Table 4.9: Type of TLV

Radar's data information is sent by default with a baud rate of 921600 bit / s, 8 data bits, without parity and with 1 stop bit. The information is also sent in hexadecimal form, so the X and Y position data must be handled to switch the information from a hex format to a float / double format. For this reason it was decided to read the data in "uint8" format. This format allows for an easier way to manage and transform information from hex format to float format.

In the tables 4.6, 4.7 and 4.8 is possible to notice the presence of the xyzQformat. The Q notation is a format establish by TI and it is normally composed by a Q letter followed by two numbers, (*m,n*), the first number *m* express the number of bit used to the integer part of the number, instead *n* represent the number of fraction bits.
The xyzQformat is very important to the commutation between the hexadecimal format sent by the radar and the double format which we want as expression of the distance. In the Firmware installed inside the AWR1642 for this application the value of the xyzQformat is always equal to 128 which is equal to 2 raised to 7. As represented in the Table 4.6 4.7 and 4.8 the data about the X and Y information are expressed on 2 bytes, so, in order to have a unique data to manage, the first byte must be summed to the second one which is multiplied by 256:

$$byte_{sum} = byte_1 + byte_2 \cdot 256 \tag{4.1}$$

Subsequently the $byte_{sum}$ must be compared with a constant, if the sum of the byte is greater than this constant the $byte_{sum}$ must be subtracted from another constant otherwise it remains equal to itself. Finally it is multiplied by the inverse of the xyzQformat. It is possible to express the concept by a simple flowchart.

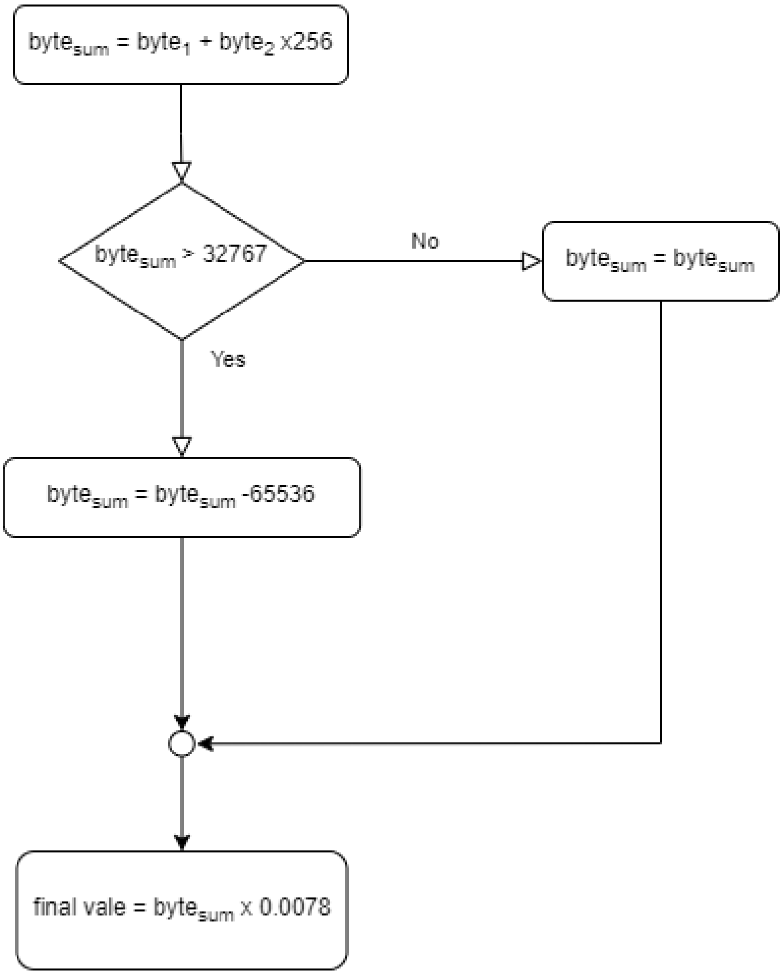

$byte_{sum}$ = $byte_1$ + $byte_2$ x256
$byte_{sum}$ > 32767
No
$byte_{sum}$ = $byte_{sum}$
Yes
$byte_{sum}$ = $byte_{sum}$ -65536
final vale = $byte_{sum}$ x 0.0078

Figure 4.10: Float calculation from byte structure

4.3.2 Simulink scheme

As been mentioned that the information sent by the radar, in the first part of the study, was read directly from the micro-USB available on the device in a Simulink environment. This section explained the Simulink scheme used to decode the Data format expressed in the previous section.

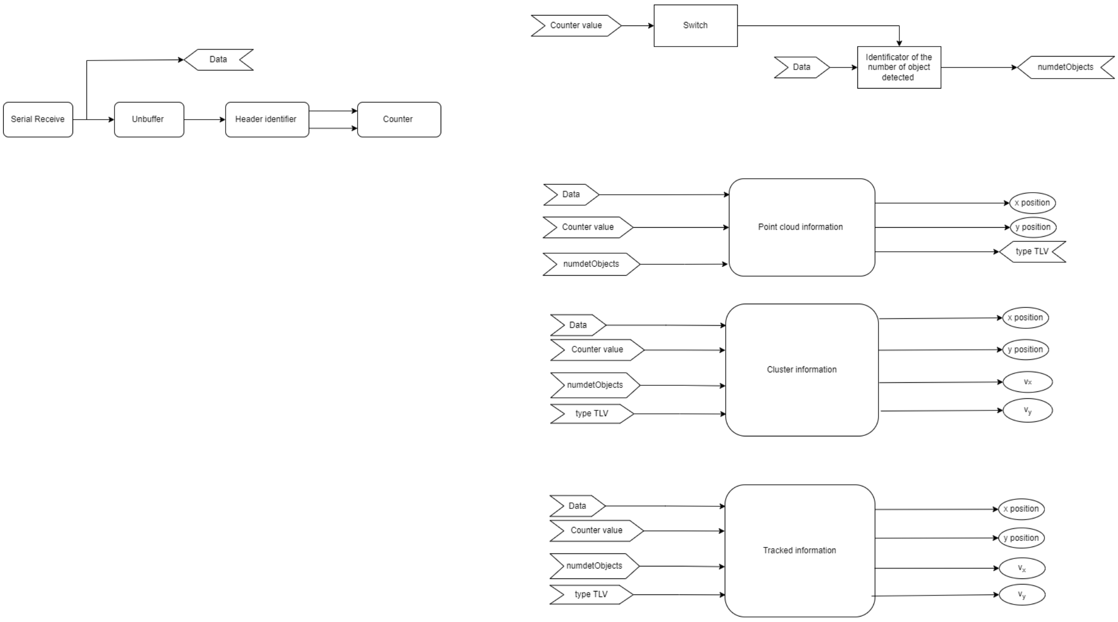


Figure 4.11: Decoding scheme implemented on Simulink Environment

The main passages are represented in the Figure 4.11 which are:

- **Serial Receive block:** In this block are configured the main parameters on how to read the data from the serial communication established with the sensor. The following setting has been chosen to read the data in Real time.

Parameter	Value
Communication port	Number of the Port
Header	<none>
Terminator	<none>
Data Size	[1 1200]
Data typer	uint8
Block sample time	0.06 s

Table 4.10: Serial Receive block parameters on Simulink

Parameter	Value
Communication port	Number of the Port
Baud rate	921600 bit/s
Data bits	8
Parity	none
Stop bits	1
Byte order	LittleEndian
Flow Control	none
Timeout	10 s

Table 4.11: Serial Configuration block parameters on Simulink

It is important to underline that Simulink is not able to read single byte at high frequency in real time, but it is able to read more low-frequency data. For this reason has been decide to buffer the bytes sends by the radar and not to read it in a polling mode (one byte at a time).

- **Unbuffer & Header Identifier:** The identifier Frame Header is always fixed, but its position inside the message is variable. For this reason has been chosen to identify the Header message with a Matlab function in which the byte sent by the AWR1642 has been read one at a time. To do this, due to the buffering process performed by the "Serial Receive" an unbuffer block has been interpose between the Serial Receive Block and the Matlab function.
- **Counter:** Due to the TLV data format, the position of the bytes determine which information the radar sends. For this reason the counter is very important to identify all the information which is sent after the Header. The counter stars counting the bytes immediately after Header identification and it is reset when a new header is found.
- **Switch:** The switch is used to determine the information about the number of point clouds objects. As indicated in the table 4.5, the numbers of detected points cloud are sent by the radar on 4 bytes. The information must be so transformed into a readable parameter, to do this must be defined a 1X4 vector:

  $$word = [1\ 256\ 65536\ 16777216]$$

  And the following expression mus be computed:

  $$numdetObjects = \sum_{k=1}^{4}(word(k) \circ byte(k)) \tag{4.2}$$

Where "word(k)" and "byte(k)" expressed respectively the word and the byte at the position k, and "$\circ$" expressed the Hadamard product.

- **Point Cloud, Cluster and Tracked information detection blocks:** The implementation used to determine the information about the point cloud, the cluster and the tracked objects it is almost the same.
  In each case has been used a Chart (Stateflow) with three main input:
  - Data sends by the radar
  - Number of detected objects
  - Number of the Counter
  - TypeTLV (Only for the Cluster and Tracked objects subsystem)

  And as output:
  - The x and y information of the point cloud detection
  - The number of cluster objects, x position and y position information of the cluster objects.
  - The number of tracked objects, x position, y position, v_x and v_y of the tracked objects

In which are computed the Equations expressed in 4.1 and in the Figure 4.10. As output of the cluster and tracked block used to individuate the information about the objects has been chosen to do not extrapolate the outputs regard the X-size and Y-size expressed in Tables 4.7 and 4.8. This decision is due to fact that these information are reputed irrelevant for the ACC's working principle developed subsequently.

4.3.3 Lead Car Identification

To identify the information on which the ACC is based, a limitation was made on the x-axis of the radar. The limitation on the x axis is due to the fact that the ACC must follow the car which is present on the same lane of the host vehicle. From the road regulations it is know that the width of a lane is about 2.80 m [10] So has been chosen to limit the transverse axis between -1 m and 1 m. This restriction allows to avoid the possible identification of objects which are present in different lane with respect to the ego vehicle. Once the objects in this range have been identified, the object with the shortest distance to the radar is determine as the lead vehicle that the ACC must follow. In case there are no vehicle which are identified as lead vehicle in the FoV of the radar the information that the launchpad sends to the ECU are the one releated to a virtual vehicle located at a distance of 80 m with a relative velocity of 0 km/h. This ensures the ego vehicle to maintain the cruising velocity until a new vehicle is detected by the radar as the new vehicle to follow.

To have the correct detection of the lead vehicle, was made a comparison between the different objects that are simultaneously detected by the radar. Has been decide to have vectors as outputs of the Stateflow used to calculate the information about the objects detected at the same time instant.

Each vector will have a number of filled cells equal to the number of tracked object identified. The main Chart on which is based the determination of the Lead car is the Tracked information block. All the outputs will be the inputs of one Matlab function which determine the position and the velocity of the target vehicle to be follow as expresse above.

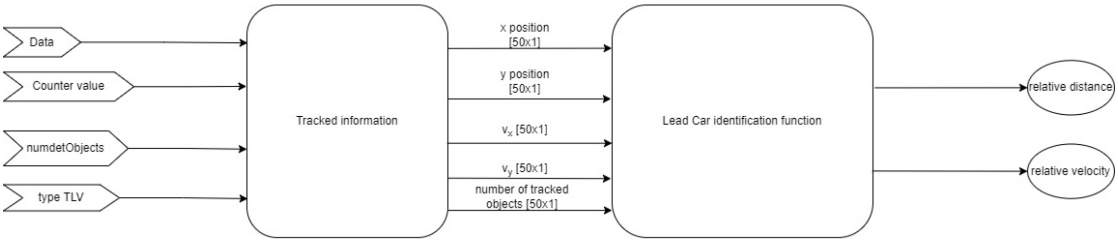


Figure 4.12: Identification of the leading car Scheme

The CAN messages sent by the microcontroller are of the uint8 data type. This data format makes a rough approximation of the data processed in the previous functions. To overcome this problem, a mapping block has been placed between the identification function of the vehicle target and the message to be sent. In this way a greater accuracy of the data sent via CAN serial is guaranteed.

The following PseudoAlgorithm is used to determine the position and longitudinal velocity of the Lead car.

Algorithm 1 Identification of the lead car

```
function $[relativedistance, relativevelocity, i] = fcn(xpositiontracking, vx, ypositiontracking, vy, numtrackedObj)($
    $)$
    persistent RELATIVEDISTANCE, RELATIVEVELOCITY;
    if $isempty(RELATIVEDISTANCE)$ then
        RELATIVEDISTANCE = 0;
        RELATIVEVELOCITY = 0;
        relativedistance = 0;
        relativevelocity = 0;
    end if

    min distance = 80;
    velocity = 20;
    i=1;
    k=1;

    if $(numtrackedObj\ \ = 0)$ then
        for $i = 1 : numtrackedObj$ do
            if $(xpositiontracking(i,1) >= -1\ AND\ xpositiontracking(i,1) <= 1)$ then
                if $(ypositiontracking(i,1) < mindistance\ AND\ ypositiontracking(i,1)\ \ = 0)$ then
                    min distance = ypositiontracking(i,1);
                    velocity = vy(i,1);
                    RELATIVE DISTANCE = mindistance;
                    RELATIVE VELOCITY = velocity;
                else
                    RELATIVE DISTANCE = mindistance;
                    RELATIVE VELOCITY = velocity;
                end if
            else
                RELATIVE DISTANCE = mindistance;
                RELATIVE VELOCITY = velocity;
            end if
        end for
    end if

    mindistance = 80;
    velocity = 0;
    RELATIVEDISTANCE = min distance;
    RELATIVERELATIVEVELOCITY = velocity;
end function
```

4.3.4 Launchpad Integration

In the Automotive sector a huge amount of information are sends via CAN communication. This is due to the great properties of this serial communication, like simplicity, reliability and its noise immunity. In the firmware used for this thesis project the CAN serial communication is not activated on the AWR1642 board. For this reason it was chosen to interconnect a TI's Launchpad F28069M between the sensor and the ECU capable of switching the UART serial communication received by the radar into CAN messages.

Figure 4.13: Launchpad F28069M

The designed Simulink model of our decoder plus the lead car function has been so deployed on the TI Launchpad with several adjustment.
The first important adjustment implemented, was to lowered the Baud rate of the serial UART communication. This change has been done directly in the firmware implemented on the AWR1642. This decision is due to the fact that after several tests, in different conditions has been noticed that the the Ti Launchpad lose a huge amount of data if the communication between the microcontroller and sensor have a Baudrate higher than 230400 bit/s.
A consecutive change was the acquisition of the data. Differently from the scheme 4.11 the data on the Launchpad has been acquired in a polling mode, so the unbuffer block has been removed from the original scheme. The launchpad is able to receive all the information sends by the radar with a sample time of 13e-5s which correspond to a frequency of 7.7 kHZ. However it is not possible to maintain this frequency on the entire scheme. The identification function of the machine to be followed requires a longer operating time. This is due to the fact that quite large vectors are handled in several if and for loops that are concatenated with each other. For this reason, a rate transmission block was interposed between the decoding of the information and the function used to identify the car to be followed. The rate transition is necessary because the function has been slowed down to a sample time of 2.3e-3, equal to 200 times the acquisition speed, which implies a

frequency of identification and sending of messages via CAN of 384 Hz.

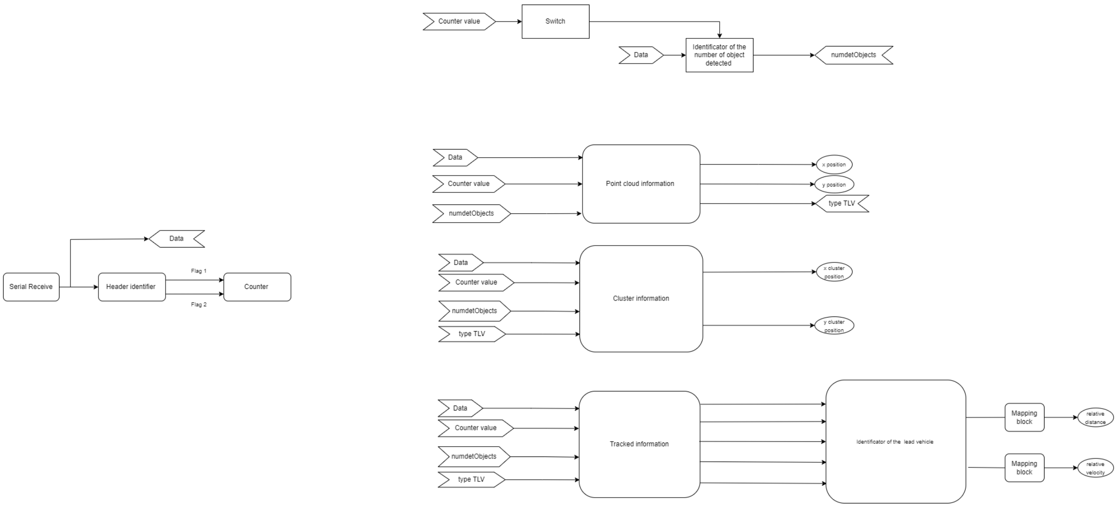


Figure 4.14: Decoding scheme deployed on Launchpad F28069M

The microcontroller therefore decodes the data received from the radar and transmits only two CAN messages to the ECU the relative distance and relative speed of the lead car.

4.4 dSPACE Implementation

The ACC is implemented and integrated on a prototype car which race in the Formula SAE. The vehicle is an electrical and driverless car with four AMK motors in-wheel. The control implemented on the vehicle is made in order to satisfy the rules imposed by the formula student competition. In order to do this the entire control of the car is implemented on a real time Electronic Control unit (ECU), the dSPACE MicroAutobox III used for performing fast in-vehicle function prototyping.

The control unit receive and manage all the feedback from the different sensors present on the car. The control implemented on the ECU consists of an upper level controller, a lower level controller. The upper level controller analyse the information received by the sensors and decide the states of the host vehicle like the acceleration command. The desired acceleration commands from the upper level controller manage the lower level controller. The lower level controller receives the desired acceleration, and commuted it into a couple distribution thanks the implementation of a torque vectoring. Instead when the deceleration request is bigger than the one achievable by the motors the lower level control operates on the brake, to satisfy the desired deceleration.
Since the vehicle is a driverless car prototype, an autonomous braking system

(ASB) has been implemented on the vehicle. This system has an important role in the deceleration phase.

4.4.1 ACC Implementation

In the ACC design and implementation a case of straight road condition has been considered. The ACC is deployed on the same ECU which control the vehicle behaviour. It can be divided in a two main states: a state machine and a Constant time Gap (CTG) control. When the ADAS system is activated, on the basis of the information received by the radar, it determines the desired acceleration of the host vehicle.

4.4.2 State Machine

The state machine implemented has the main role to set the cruising velocity when the driver activate the ACC. In similar way to the one implemented on the commercial cars, has been inserted a button on the cockpit of the prototype car able to activate and deactivate the ACC. When the driver press the button the car tend to maintain the velocity set at that time instant. The implementation on the Simulink environment is made through a Stateflow which receive different fundamental inputs to determine the state of the ACC:

- **Speed of the center of gravity (V_CG):** it is the main input which is determine through a Fuzzy logic based on the data received from the encoders available on the wheel of the car.

- **State of the ACC (ACC_State):** It is set as Boolean variable. It is set to 1 when the ACC is activated (the driver press the button on the cockpit for the first time) and 0 when the ACC is deactivated (driver press the button on the cockpit for the second time)

- **State of the CC (CC_State):** It is set to 1 when the driver want to increase the velocity and set to 2 when the driver wants to decrease the velocity of the car. Unlike the standard application of these buttons in standard cars, it was decided to impose a fixed speed increase or decrease of 5 km/h each time the appropriate buttons were pressed.

- **Brake pressure:** The brake pressure it is another important input which determine the deactivation of the ACC. When the driver press the brake pedal the ACC must be deactivated.

As output of the state machine there is the speed that the host vehicle must maintain on the basis of the inputs received.

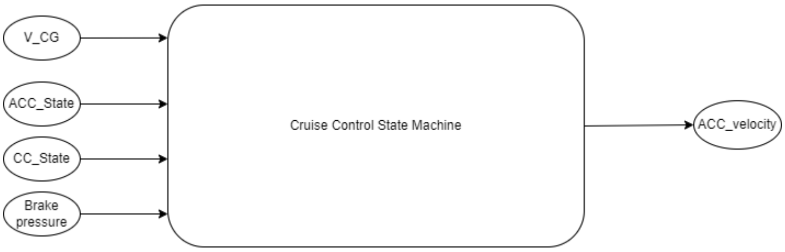


Figure 4.15: I/O Cruise control state machine

To determine the longitudinal behaviour of the car when the ACC is activated it is possible to divide the State Machine in four different states:

- **State 0:** In the State 0 the ACC is deactivated and the velocity of the car is determined from the pedal throttle imposed by the driver.
- **State 1:** This is the Central state of the state machine in which the ACC is activated from the driver. The vehicle maintain a constant velocity at the instant on which the button is pressed by the driver, or the speed undergoes variations due to state 2 or 3. This state is disactivated when the driver press again the ACC activation button or press the brake pedal.
- **State 2 and State 3:** Both State 2 and state 3 are an instantaneous states connected with state 1, where the velocity is increased (State 2) or decreased (State 3) by 5 km/h.

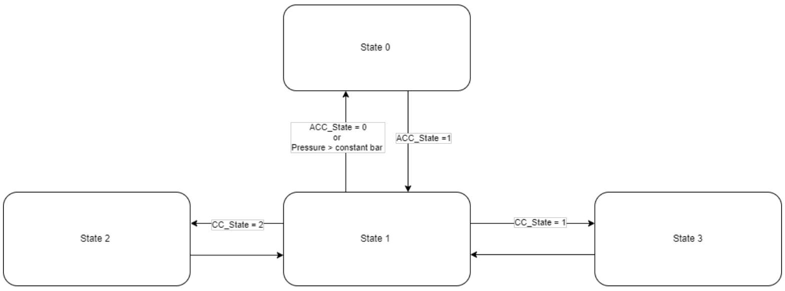


Figure 4.16: State Machine

The ASB in the SCD22 has been implemented as a mechanical system consisting of a servomotor that rotates, and a steel wire which acts directly on the brake pedal. The angle given to the pedal determines the brake oil pressure. This system has an important role in the deceleration phase imposed by the ACC and on its deactivation.

To guarantee the correct behaviour of the system, the deactivation of the ACC in

braking situation, must be guaranteed only when the brake pedal is pressed by the driver and not when it is pressed by the ASB. To do this, a Matlab function has been implemented before the State Machine. Inside has been made a control on the state of the ASB and on the brake oil pressure. Has been noted that the maximum pressure action exercised by the ASB system is about 15 bar. For this reason the ACC is deactivated when the ASB does not act on the braking pedal or if the brake oil pressure is bigger than 15 bar when the ASB is activated.
Due to the fact that the button on the cockpit of the prototype car sends an impulsive signal, the Matlab function also has the role to maintain the signal "ACC_State" to 1 when the button is pressed for the first time, and to 0 when it is pressed for the second time by the driver.

4.4.3 Constant time gap

The State Machine implemented is set only to maintain the desired velocity by the driver. The speed of the car subsequently must be tune accordingly to the information receive by the radar, the relative distance and the relative velocity of the lead car. The control system used to regulate the longitudinal behaviour of the car is a Constant time Gap (CTG) control. As mentioned the entire control of the vehicle is composed by an upper level controller and a lower level controller. The upper level controller compute the desired acceleration (a_{des}) for the host vehicle to achieve the desired spacing or velocity. The upper level controller operates in two different control modes:

- **Velocity control:** the velocity control mode is entered when the radar does not detect any vehicle in the forward path oh the host vehicle. The ACC regulate the host velocity at the velocity imposed as target by the driver.
- **Space Control:** the space control mode instead is entered when the radar sensor detects a leading vehicle in front of the host vehicle. The host vehicle is controlled to maintain the desires distance from the leading vehicle. defined as.

$$d_{tar} = d_0 + t_{h,tar} \cdot v_h \tag{4.3}$$

  where:

  - d_0 is the safety distance to be maintained in case of complete stop of the lead vehicle.
  - $t_{h,tar}$ is the time headway, it is the time it takes for the host vehicle to collide with the target vehicle when the latter stops and the host vehicle maintains its speed.
  - v_h is the host vehicle velocity.

The CTG block implemented on Simulink received as inputs the speed of the center of gravity of the car (v_h) and the two main information sends by the radar, the relative distance (d_{rel}) and the relative velocity (v_{rel}) of the lead vehicle. The cruising velocity of the car is so regulate on the basis of the information received by the radar. The theoretical structure of the CTG control can be represented as:

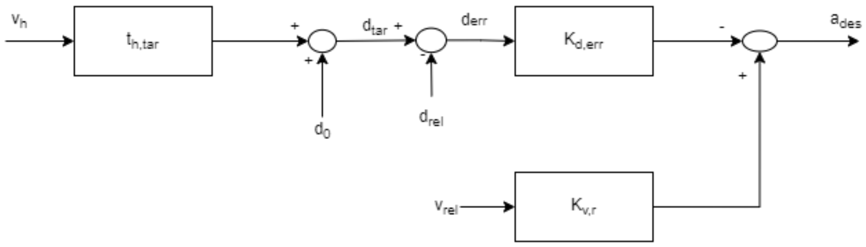


Figure 4.17: CTG scheme

Which can be expressed by the following equation:

$$a_{des} = v_r \cdot K_v - k_{d,err}(d_{tar} - d_r) \tag{4.4}$$

The control used includes a Proportional action on both the relative distance error d_{err} and the relative speed v_{rel}. The are two proportionals gains denoted with $K_{d,err}$ and $K_{v,r}$. The two gains are used to give more or less importance to the relative distance or to the relative velocity. An higher value of $K_{d,err}$ with respect to $K_{v,r}$ means that the relative distance has more effective on the desired acceleration with respect to the relative velocity.

Based on the sign of the desired acceleration imposed on the CTG the signal must receive a subsequent modification. For positive values of acceleration, the command is sent to the motor, instead for negative value of acceleration the signal must be sent to the ASB system. The request acceleration calculated by the 4.4 equation has been limited. The upper and lower limit was chosen on the basis of the regulations in force on ADAS systems expressed by European regulations [ref] and to guarantee a good comfort to the driver.

When the acceleration is positive, due to the necessary to have a target speed as input of the torque vectoring system implemented on the SCD22 the desired acceleration when it is positive, must be integrated. For this reason has been chose to integrate and saturate the a_{des} with an anti-windup system. The velocity imposed after the integrator is saturated between 0 and the velocity imposed by the State Machine. The saturation block is used to guarantee the reachability and not exceeding of the cruising velocity set on the State Machine, when no target vehicle are individuated by the radar. This implementation guarantee a smooth and fast response by the system.

Consequently for negative values of the desired acceleration the output of the CTG must be multiply by a constant used to scale the desired acceleration to have a plausible value for the ASB system.

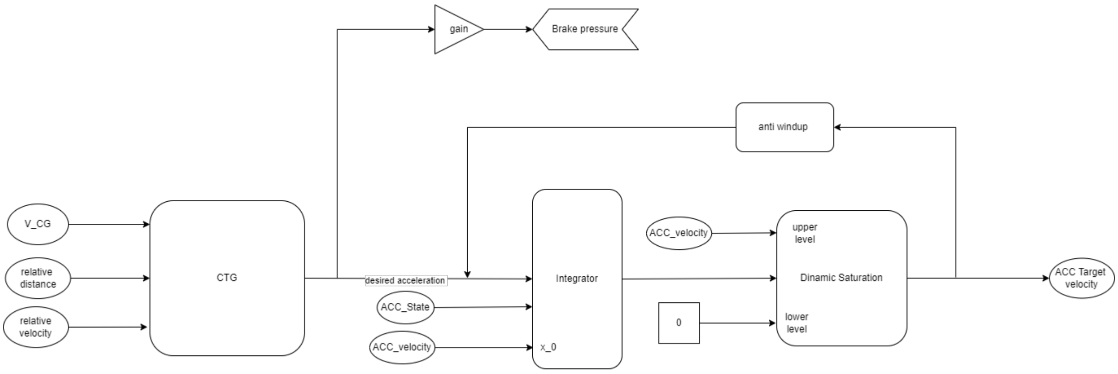


Figure 4.18: I/O CTG

4.5 Control of the motors

The torque that must be imposed to the motors can derive either from the throttle pedal or from the ACC. To differentiate these inputs and manage the longitudinal behaviour of the car, a dynamic saturation has been interpose between the input velocity and the Torque vectoring. The main input of the dynamic saturation block is the signal coming from the throttle pedal, while the lower limit changes according to the activation or deactivation of the ACC. When the ACC is engaged the lower limit is enforced to be equal to the velocity imposed by the ACC. In this way when the driver release the throttle pedal the speed is imposed by the lower limit and consecutively by the speed set by the ACC.

The dynamic saturation block allow also the driver to increase the velocity of the car temporally. In fact if the driver through the throttle pedal give an input velocity grater than the velocity imposed by the ACC the car starts to accelerate. When the driver release the throttle pedal the SCD22's speed is again controlled by the ACC. When the ACC is deactivated the ACC_State is equal to zero and so the lower limit is imposed to zero and the torque depends only on the velocity imposed by the throttle pedal.

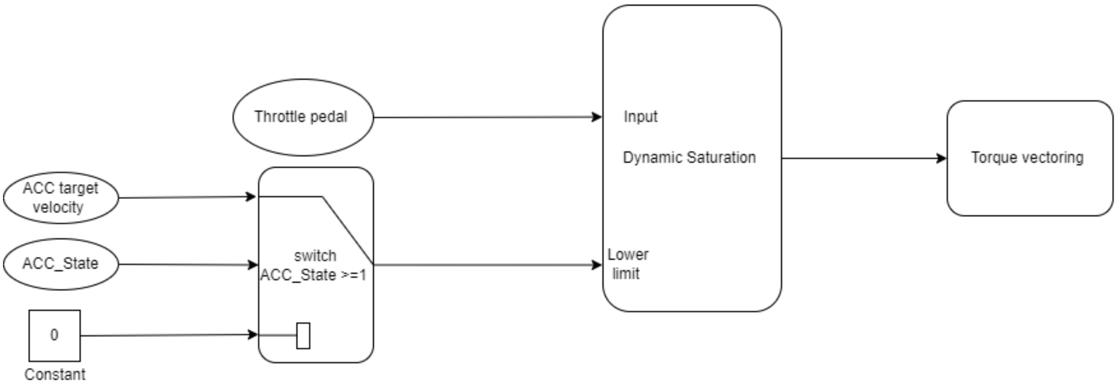


Figure 4.19: Control of the motors Scheme

Chapter 5

Hardware Integration

All the implementation expressed in the previous Chapter as been mentioned has been integrated on a prototype car which race in the Formula SAE competition. The Radar sends the information about the objects present in its FoV to the Launchpad via serial UART communication. Inside the microcontroller is deployed the designed Simulink model which receive the data from the sensor and transmit the information about the relative distance and velocity of the lead vehicle via serial CAN communication to the ECU dSPACE Microautobox III. In the ECU is implemented the ACC and the entire control of the car which govern the behaviour of the vehicle.

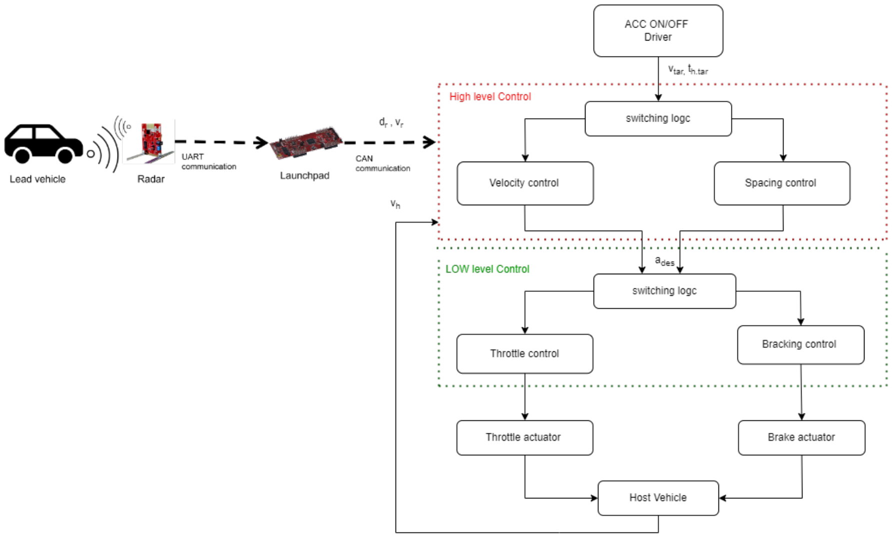


Figure 5.1: Structure of the communication

5.1 Communication

The main communication used in the Automotive sector is the CAN communication. This allow to create a robust communication between the ECU, with a single cable with a "high immunity", to electrical interface and the ability to self-diagnose and repair data error.
The type of information sent by the radar used in this project depends on the firmware installed on it. For this thesis project has been used a firmware provide by the producer TI. The original firmware installed on the AWR1642 boars enable only the Serial UART communication with a Baud rate of 921600 bit/s. With this standard implementation the senor is able to send data only by the micro-USB connector available on the board.
Looking to the physical hardware it is possible to see that a CAN interface connector is available on the board. Due to the presence of this connector, the first change which is made, on the original firmware, was to try to activate the CAN communication. However after several firmware changes only CAN-FD was activated.
The CAN-FD with respect to the classical CAN communication it is able to increase the data transfer up to 5 times faster and with larger frame messages size for the ECUs. The primary difference between the CAN and the CAN-FD is that the ECUs can dynamically switch to different data-rate with smaller or larger message size.
However with this kind of communication several problem has been encountered. The first main problem was the decoding of this information. Has been noted that the information sent by serial CAN communication differentiate from the serial UART. In particular has been observed several padding byte, present in casual order, at the end of some messages. The presence of these bytes caused a difficult decoding process of the information sent by the radar, and consequently a difficult creation of the CAN bus database (CAN DBC). Another consecutive problem was the impossibility to the dSPACE Microautobox III to read this type of CAN communication.

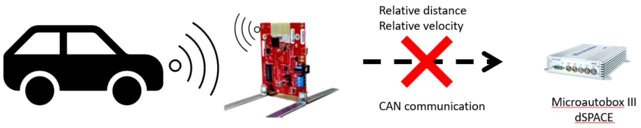


Figure 5.2: AoA accuracy with respect to angle

Due to the impossibility to have a direct communication between the radar and the ECU via CAN communication has been decide to interconnect between the radar and the dSPACE a microcontroller. It has the role to send the main information about the lead vehicle to the ECU via CAN. The radar so communicate via serial UART communication to the Launchpad which elaborate the data and sends the main information to the dSPACE Microautobox III.

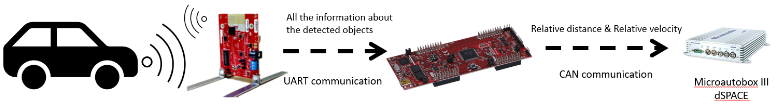

All the information about the detected objects
UART communication
Relative distance & Relative velocity
CAN communication
Microautobox III
dSPACE

Figure 5.3: AoA accuracy with respect to angle

To communicate between the Radar and the Launchpad several changes has been made on the AWR1642 board. Looking to the schematic of the sensor it is possible to note that the serial UART communication present on the micro-USB is also available in one of the 20 pin Boosterpack connector on the back of the board. The pin is located at the position 9 of the J5 connector (MSS_LOGGER), expressed in the Table 4.3. However this pin is not activated by default. In order to activate it and have a UART communication available from this pin a resistor of 0 Ω must be add with a solder bold. This guarantee rhe continuity of the signal which allow to communicate also from this pin.

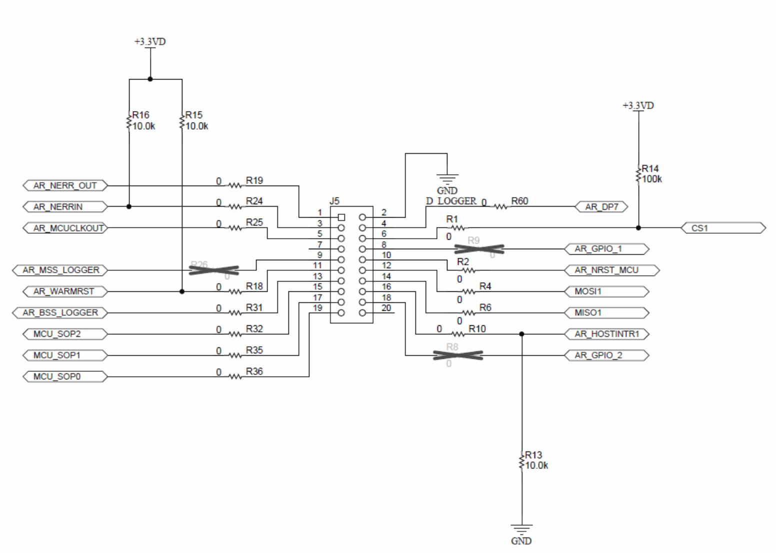


Figure 5.4: J5 schematic

The hardware communication between the Radar and the Launchpad is made through cable which are properly welded on the two boars.

The Launchpad receive all the information from the radar in a hexadecimal form. These information are read in a uint8 format by the microcontroller, elaborate and managed in a float format through the decoding Simulink scheme deployed on it and sent in a uint8 format to the ECU via CAN communication. The information is sent in uint8 format from the microcontroller because this is the mandatory data type required by the Launchpad for CAN communication.

The presence of the Launchpad between the radar and the ECU causes several problems from the point of view of data communication. With the standard firmware implemented on the radar, the frequency of the information sent by the AWR1642 is too high to be readable and elaborate by the microcontroller. To solve this problem, it was decided to decrease the Baud rate of the data sent by the sensor to a speed that would allow the launchpad to read the data correctly in real time. However the Baud rate should not be too small, this because in the case of a complicated scenario, with more objects to be identified, the radar stops. The freezing situation is due to the fact that the number of information to be sent is too high for the amount of time available. After several acquisitions in different scenarios a baud rate of 230400 bit / s was imposed.

To establish a correct detection of the messages sends by the launchpad a CAN DBC file must be created. This is a text file that contains information such as message ID, message name, DLC, and other information for decoding the RAW CAN bus data to human readable values. The DBC file has a well-defined structure where each value in a position has its own meaning. Each message has a CAN identifier which make them unique.

To guarantee a correct and unique communication between the hardware available in the DBC file must be defined the receiving node and the transmitting node. In our application the receiving Node is always the Microautobox III and the transmitting Node is always the Launchpad and so the Radar.

The Radar does not need any messages from the ECU to be activated. The instant at which the radar start to send the information is defined directly on the firmware installed on it which is made by some modification on the code. For simplicity, has been imposed that the radar starts sending information about the detected objects as soon as it is turned on. Consequently at the node's definition, to understand which information has been sent by the microcontroller must be define the messages. The messages received by the Launchpad are the relative distance and the relative velocity. In order to guarantee the correct identification of the messages sent by the microcontroller, must be defined an ID to each message. This must be unique, and allows the ECU to understand which data is sent by the Launchpad. Has been decide to attribute an ID equal to 1 to the message contained the information about the relative position of the leads vehicle, and an ID equal to 2 for the message which contain the information about the relative velocity of the leads vehicle.

Name	TX messages	RX messages
dSPACE	none	relative distance, relative velocity
Radar	relative distance, relative velocity	none

Table 5.1: Node definition

Name	ID decimal	TX Node	RX Node	Length
relative distance	1	Radar	dSPACE	8
relative velocity	2	Radar	dSPACE	8

Table 5.2: CAN messages

Also in this case a properly cable management has been made in order to guarantee the correct communication between the ECU and the microcontroller.

The entire control which govern the entire behaviour of the SCD22 is deployed inside a unique ECU, the dSPACE Microautobox III. It is possible to divide the entire control of the vehicle in two main parts. The high level control of the car, and the low level control of the car. The high level control receive the information by the several sensor and devices implemented on the car and determine the acceleration of the vehicle. The lower level control on the basis of the acceleration request by the high level control determined the torque on the motors present on the wheels. The entire control implemented on the SCD22 with the implementation of the ACC has undergoes some changes. In the upper level control has been implemented the State Machine and the CTG expressed in the previous chapter. So, when the ACC is activated by the driver, the acceleration of the SCD22 is governed only by the ACC and consecutively by the CAN messages received by the micrcontroller. The acceleration is transformed initially in target velocity which the vehicle must achieve. The target speed is given to the motors as torque control thanks the use of a torque vectoring.

5.2 Integration for the SCD22

To have a configuration similar to the one that is present on the commercial cars, it was decided to integrate the radar and consecutively the micocontroller inside a box. The box has been specially modified in such a way it can contain the two boards inside. To place the microcontroller inside the box, a PLC support has been created which has been fixed inside the box.

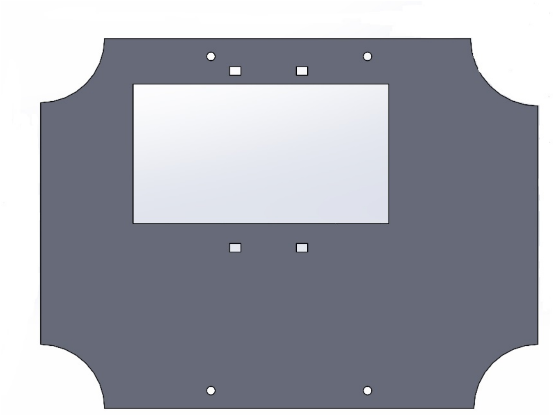


Figure 5.5: Front view of the support used for the Launchpad

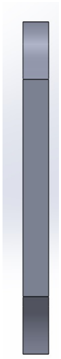


Figure 5.6: Lateral view of the support used for the Launchpad

The main hole has been used to place the microcontroller inside the support, however the other holes are used to fix the support to the BOX.

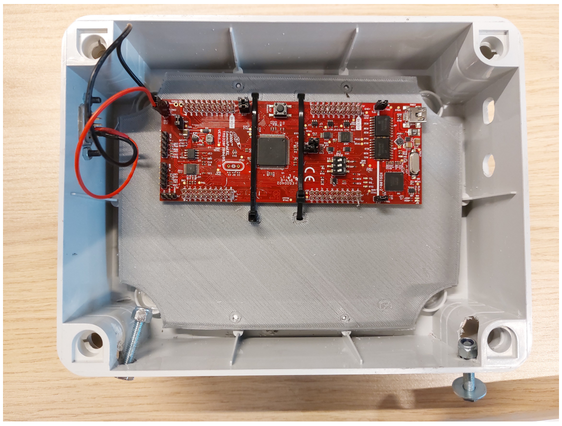


Figure 5.7: FMWC block diagram

A support has been created which guarantees the fixing of the radar to the lid of the box. The vertical angle of the radar with respect to the ground is very important for the acquisition of the objects present in the FoV of the sensor. Has been noted that an angle of 3-4 degree with respect to the ground can effect drastically the objects detection. For this reason the radar has been fixed in multiple point to guarantee a position as perpendicular as possible to the road. For this reason the sensor has been fixed in multiple point to the box. It has been fixed on the bottom part to the support with two M3 screw, and on the top part to the lid of the box with two M3 screw. This guarantee a stable vertical position of the radar also in case of different vibration due to the road surface.
To avoid the presence of possible interference that could disturb the radar chirp transmission and reception, it was decided to cut the front part of the box. This allow a free surface in front of the radar.

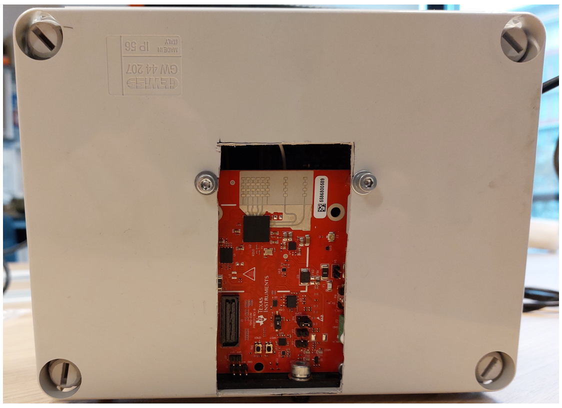


Figure 5.8: FMWC block diagram

The serial UART communication between the radar and the Launchpad has been insured with the welded cable inside the box.

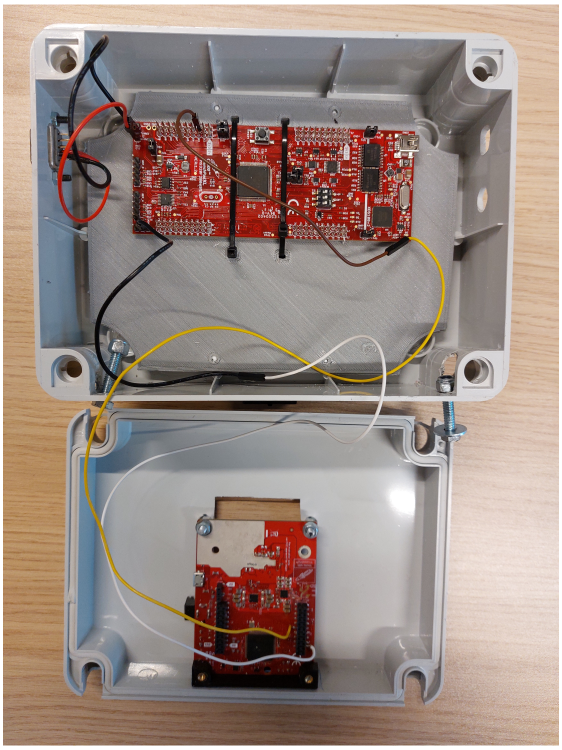


Figure 5.9: Communication inside the BOX

Where, in the Figure5.9 the cable are so defined:

- **Yellow/Black cable:** define the serial UART communication
- **White/Brown cable:** define the ground (GND) between the two board.
- **Black cable:** define the CAN-L lines starting from the launchpad
- **Red cable:** define the CAN-H lines starting from the launchpad

In one of the lateral part of the box instead the box has been properly modified in order to fix on it a DB9 connector. This connector is used for the CAN communication between the microcontroller and the ECU. Its presence guarantee a solid communication which will be very important in the test phase.

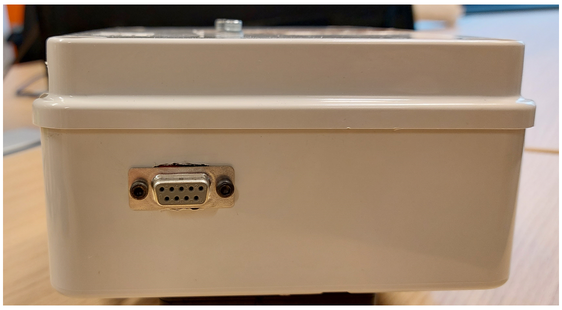


Figure 5.10: FMWC block diagram

On the other side of the box, instead, two holes were made which were used for the passage of the power cables of the radar and the launchpad.

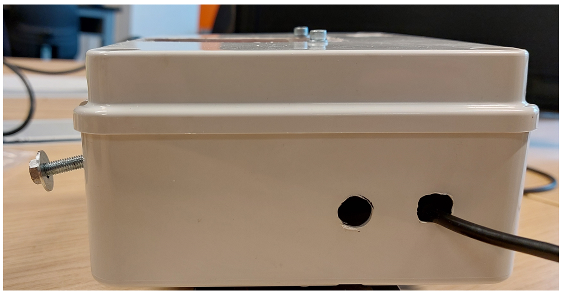


Figure 5.11: FMWC block diagram

To avoid the presence of long cable on the external part of the prototype car, and to avoid possible power problems has been decide to power the radar and the launchpad by a power bank which was positioned in the rear part of the box through the use of two supports made in PLC through a 3D printer.

Figure 5.12: Power bank

This guarantees a power supply of 5V to 2.A which allows to power the microcontroller and the radar for very long test sessions.

5.3 Integration on the SCD22

To better capture the objects placed in the radar FoV, it was decided to place the box containing the radar and the microcontroller in the front of the machine.
The decision of the positioning on the car of the sensor is a crucial point for the correct acquisition of the objects located in the FoV of the radar. After several tests it was decide to locate the radar at a height of 60 cm from the ground. With the Tx and RX antennas at 65 cm from the ground.

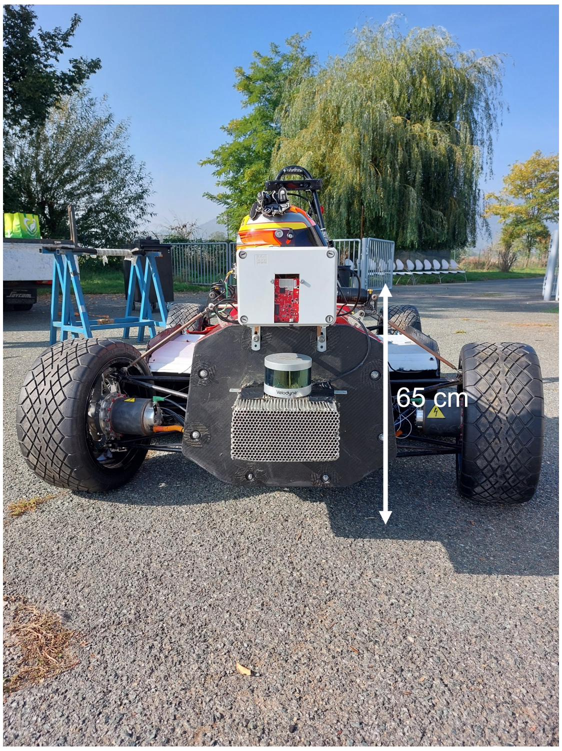


Figure 5.13: L-shape support

This positioning ensure a correctly identification of the objects up to a distance of 70 m. The decision of the height from the ground was also influenced by the structure of the car. The radar has been located in the front part of the prototype because this position can guarantee a good data acquisition and a good stability. The box containing the radar has been fixed to the car through two L-shapped support which are represented in the following photo.

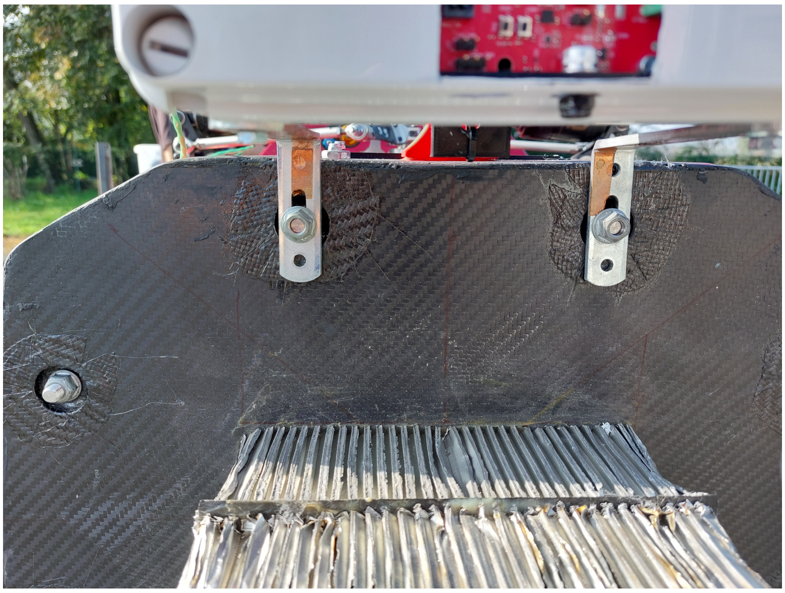


Figure 5.14: L-shape support

As mentioned in the previous chapter, a DB9 connector has been specially implemented in the side of the box. A long CAN cable has been suitably soldered to this connector to obtain the information sent by the launchpad. The cable goes directly to the ECU to communicate directly with it from the microcontroller.

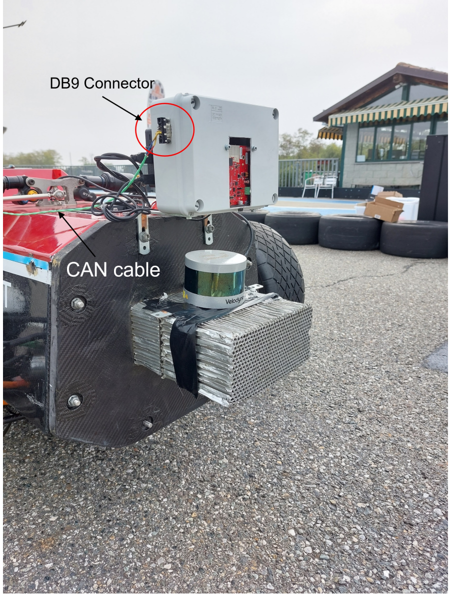


Figure 5.15: CAN communication from box

Chapter 6

Test and Results

Several tests were performed to examined the implementation and integration of the ACC. The control implementation was initially checked with a simulation of the information extracted from the radar to test the correct functioning of the ACC in different situations. Subsequently the ACC is tested in a static situation with input information coming directly from the radar to test the accuracy of the information sends by the sensor. Ultimately, the entire implementation has been tested in various simulations with different real driving conditions, on the basis of which the goodness of the controller in terms of passenger comfort was evaluated. Before analyzing the data, it must be considered that the tests were carried out on a prototype car with very different characteristics from those normally found on common commercially sold cars. The first characteristic is the weight of the machine. The car has a weight of 230 kg which is much less than that of any commercial car, the traction is integral, thanks to the 4 AMK motors on the wheels, while the braking procedure is almost similar to that present on the common cars we drive on a daily basis. The parameters configured for the correct functioning of the ACC are based on the characteristics mentioned above.

6.1 Static simulation with simulated inputs

The first test phase of the ACC was carried out with the car positioned on the stands and simulating the data sent by the radar with relative distance values and relative speed which could simulate real driving situations. This phase was very useful to test the correct behaviour of the ACC and for having the first indications on the parameters to set on the CTG, in order to have a good and fast response from the car. The parameters that influence the ACC response on acceleration and braking situations are the gains $K_{d,err}$ and K_v expressed in the formula 4.4 and the saturation limit imposed on acceleration desired ACC-imposed acceleration.

The first simulation was performed by simulating the approach and consecutive departure of a vehicle with the following settings:

- d_0 = 10 m
- $t_{h,tar}$ = 1.5 s
- $k_{d,err}$ = 0.02
- k_v = 0.02
- a_{min} = -3 m/s^2
- a_{MAX} = 2 m/s^2

In this phase, in addition to the reaction of the CTG based on the fictitious inputs sent by the radar, the correct activation and deactivation of the ACC was evaluated. The working principle is the same on the one implemented on the commercial auto. The ACC is activated when the appropriate button is pressed on the cockpit and deactivated when the button is pressed for the second time. Another test has been done on the braking pedal. As been mentioned the ACC must be deactivated when the oil pressure is grater than 15 bar. The last test that was carried out is a condition of possible overtaking, where the driver wants to increase the speed imposed by the ACC through the throttle pedal. The only difference compared to the classic ACC implemented on road cars is the increasing and decreasing of the speed thanks to the appropriate button when the ACC is activated. Due to the inability to add two new buttons on the cockpit of the prototype car, which allows the vehicle to increase and decrease the speed, the inputs has been done and tested only via software.
The CTG was also tested in a stop-and-go situation with a virtual car positioned in front of the ego vehicle. To carry out this simulation, an approach to a vehicle with, relative speed equal to zero, and with relative distance decreasing up to the safety distance d_0 imposed on the control was simulated. The test shows that the ego vehicle decreases its speed until it stops completely when the safety distance has been reached. The restart was tested in a similar way, simulating a move away of the virtual car placed in front of the sensor. In this situation the ego vehicle begins to increase its speed until it reaches the cruising speed. All the tests carried out reported positive results, demonstrating the correct functioning of the ACC implemented on the SCD22.

6.2 Data acquisition from the radar

After testing the correct functionality of the ACC, based on the simulated inputs, the reliability and accuracy of the data sent by the radar was verified. To authenticate the accuracy of the information sent by the sensor, the SCD22 was positioned on stands at a height of about 30 cm from the ground.

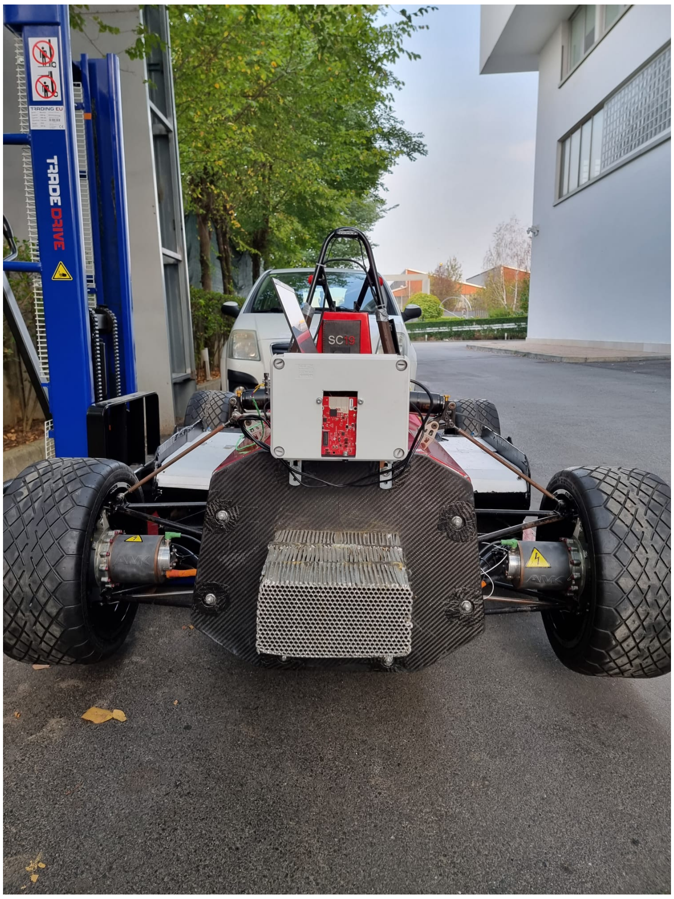


Figure 6.1: Vehicle positioning for information testing

The height was chosen in such a way as to allow the wheels to move freely in order to have a feedback on the fictitious speed of the center of mass, and to allow the radar to see the vehicles placed in its FoV. To test the error of the information has been located a car in front of the radar a fix distance of 6 m and has been verified the data acquisition which has been transmitted through the CAN messages between the launchpad and the ECU.

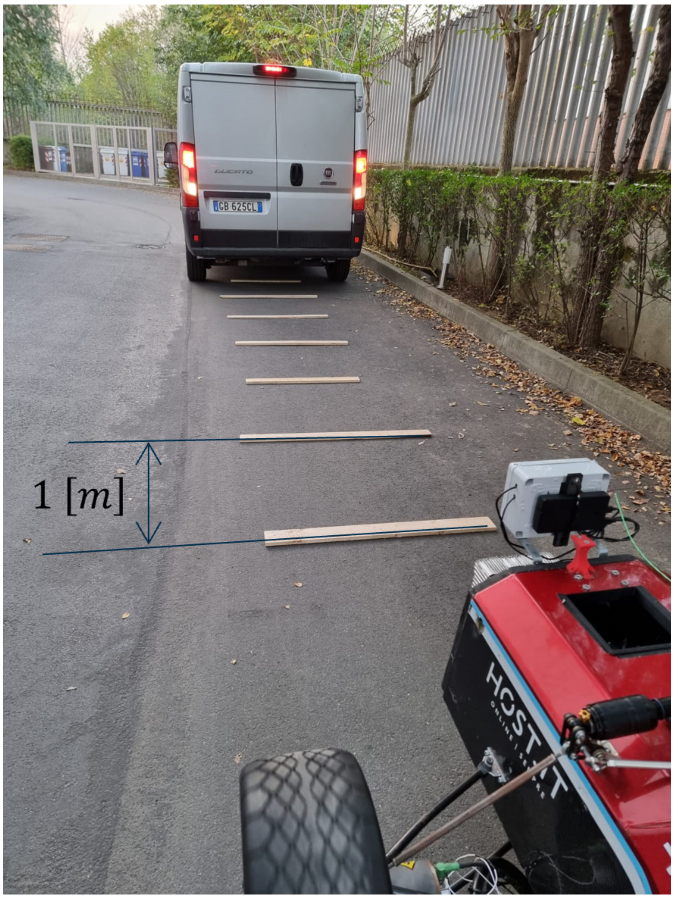
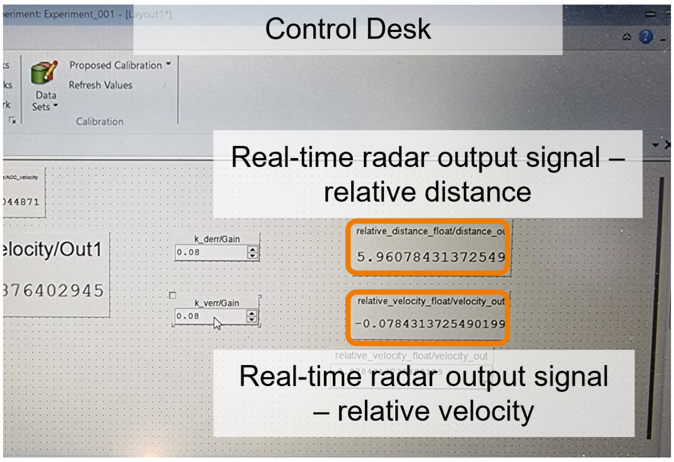


Figure 6.2: Validation of radar output information in terms of relative distance



Real-time radar output signal – relative distance

relative_distance_float/distance_ou
5.96078431372549

relative_velocity_float/velocity_out
-0.0784313725490199

Real-time radar output signal – relative velocity

k_derr/Gain
0.08

k_verr/Gain
0.08

/ACC_velocity
)44871

elocity/Out1
376402945

Figure 6.3: Relative distance and relative velocity of the lead vehicle

From the figure **??** and 6.3 it is possible to notice that the information about the lead vehicle sends by the radar are quiet accurate. The error is about 0.04 m. This accuracy guarantee a good and fast response of the ACC implemented on the SCD22 in the test phase on the different real scenarios.
After verifying the accuracy of the data sent by the radar on the relative distance and relative speed of the objects placed in its FoV, the correct functioning of the ACC was verified again. In this case it has been tested on the basis of the data acquired directly from the sensor and no longer through the simulated inputs. The test has made simulating an approaching and a departure of the lead vehicle. Due to small area available and the close proximity of the lead vehicle and the impossibility to reach huge speeds with the lead vehicle, has been chosen to give more influence to the relative distance value with respect to the relative velocity on the CTG equation. For these reasons $K_{h,tar}$ increase with respect to the previous static test done without the sensor integration. Also the safety distance decrease to simulate the stop and go function of the ACC implemented. The main parameters of the CTG has been so change in the following values:

- d_0 = 3 m
- $t_{h,tar}$ = 1.5 s
- $k_{d,err}$ = 0.08
- k_v = 0.02
- a_{min} = -3 m/s^2
- a_{MAX} = 1.5 m/s^2

Also in this case the ACC works fine with the input acquisition which come directly from the sensor

6.3 Simulation of road conditions

After the testing on the correct functionality of the ACC both with simulated inputs and with the real inputs coming directly from the radar sensor, has been tested the entire working principle in a possible real condition scenarios.
The tests has been done in a safety conditions. The tests were carried out on a free kart track with only the SCD22 and another car which simulate the lead vehicle which the ACC must follow. Differently from what the regulations impose for the testing phase of the ACC [ref] has been chosen to validate the ACC working principle at lowered speed. This is due to the impossibility to reach speeds greater than 50 km/h on the kart trace.

The first test has been done with a constant speed of 8 km/h with the approaching of the ego vehicle to the lead vehicle without the simulation of the stop-and-go situation. From the data acquisition it is possible to see that the SCD22 response in a good manner to the approaching and departure of the lead vehicle.

The following setting has been imposed on the CTG scheme:

- d_0 = 3.5 m
- $t_{h,tar}$ = 1.5 s
- $k_{d,err}$ = 0.08
- k_v = 0.08
- a_{min} = -3 m/s^2
- a_{MAX} = 2 m/s^2

In the first phase of the test on the possible realistic scenario has been chosen to give the same importance to the relative distance and to the relative velocity. Furthermore has been chosen to have a safety distance equal to 3.5 m due to the lower speed reached by the ego vehicle.

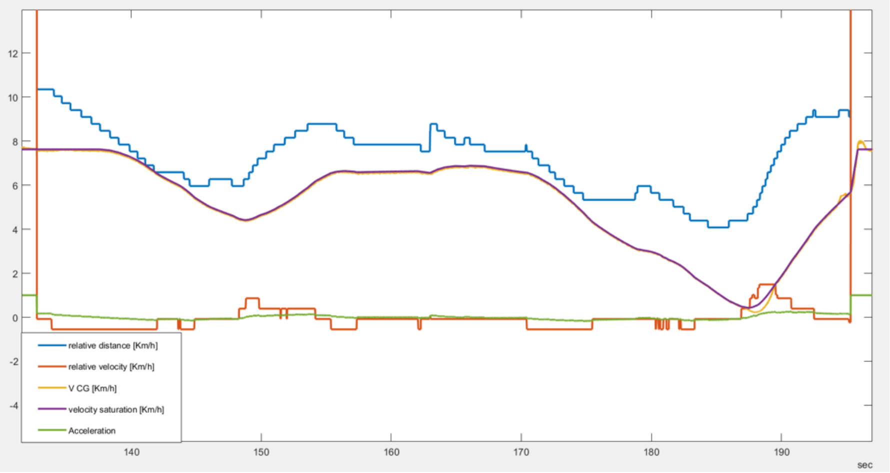


Figure 6.4: ACC test at 8 km/h

In the Figure 6.4 and in the Table 6.1 has been expressed the relative distance and relative velocity with respect to the lead vehicle, the desired acceleration imposed by CTG, the velocity of the center of mass, and the velocity saturation which derive from the integration imposed on the desired acceleration.

Time **[s]**	**Relative distance** **[m]**	**Relative velocity** **[km/h]**	**Desired acceleration** $[m/s^2]$	V_CG **[km/h]**	**Desired velocity** **[km/h]**
133.36	10.353	-0.078	0.1691	7.585	7.626
136.36	9.411	-0.549	0.0561	7.585	7.626
139.36	7.843	-0.549	0.0561	7.413	7.481
142.36	6.588	-0.090	0.0561	6.325	6.398
145.36	5.960	-0.078	-0.102	5.198	5.257
148.36	5.960	0.392	-0.038	4.441	4.463
151.36	7.843	0.392	0.093	4.954	5.018
154.36	8.874	-0.078	0.0908	6.170	6.240

Table 6.1: Data acquisition at 8 km/h

It is possible to notice that the ego vehicle have an initial velocity of 8km/h until a lead vehicle enter in the FoV of the radar at a relative distance (blue line) of about 10 m. At the same time instant on which the car is identified by the radar the acceleration starts to decrease drastically (green line), this cause a consecutive decrease of the velocity saturation (violet line) and of the velocity of the center of mass (orange line). After the first approaching the relative distance of the lead vehicle starts to increase and consecutively the desired acceleration increase, with a consecutive increasing of the V_CG.

The second function of the ACC which must be tested is the stop-and-go feature implemented on the CTG. To authenticate this functionality has been chosen to simulate a following of the leading vehicle which starts to slow down until stops. After the stop the lead vehicle restarts its motion and so the relative distance starts to increase again. The test has been done with the prototype car at a cruising velocity of about 12 km/h and after several tests the best parameters which allows a smooth and fast response with a good driver comfort are the following one:

- d_0 = 8 m
- $t_{h,tar}$ = 1.5 s
- $k_{d,err}$ = 0.15
- k_v = 0.08
- a_{min} = -3 m/s^2
- a_{MAX} = 1.5 m/s^2

These parameters allow the host vehicle to stop at a safety distance of about 8-10 m from the stationary vehicle in front of the ego vehicle. The safety distance has been increased from 3.5 to 9 and given the low speeds, also in this case more importance has been given to the relative distance than to the relative speed. The maximum acceleration has been also decreased to 1.5

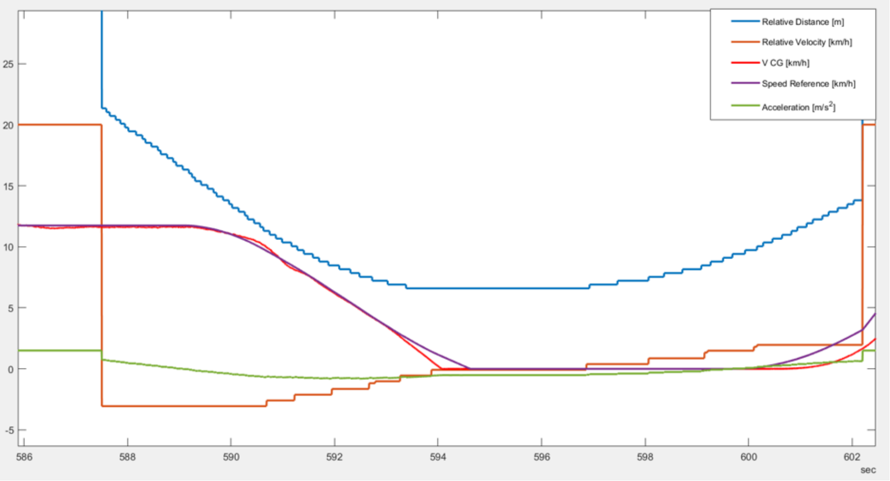


Figure 6.5: Start-and-stop test

As it possible to see from the figure 6.5 the car enter in the FoV of the sensor at a distance of 20 m and starts to decrease its velocity rapidly and consecutively the relative distance decrease drastically. These factors cause a big deceleration of the car and a decreasing of the velocity of the center of mass until 0 km/h. The ego vehicle stops at a distance of 6.58 m from the lead vehicle until it departures again. The relative distance at which the host vehicle stops is less than the safety distance imposed on the CTG. This is due to the gains imposed which influenced the relative distance and velocity. Increasing $k_{d,err}$ and k_v a quicker response from the point of view of the system would be obtained, but there would be a more drastic deceleration and acceleration which vary too fast in response to the sensor's information. This behaviour will cause less comfort for the driver. For this reason we have chosen not to modify these values.

However the safety distance is maintained in case of restart of the leading vehicle. In fact the car restarts when the lead cars reach and overcome the safety distance d_0.

Time **[s]**	**Relative distance** **[m]**	**Relative velocity** **[km/h]**	**Desired acceleration** $[m/s^2]$	V_CG **[km/h]**	**Desired velocity** **[km/h]**
587.5	21.333	-3.050	0.728	11.628	11.743
590.5	11.921	-3.058	-0.609	10.436	10.080
593.5	6.58	-0.549	-0.672	1.8576	2.25
595.5	-0.0784	6.588	-0.518	0	0
598.5	7.843	0.8627	-0.2545	0	0
601.5	12.549	1.960	0.516	0.3726	1.716
151.36	15.686	-0.078	0.3346	8.194	8.345

Table 6.2: Data acquisition in a stop-and-go situation

The last test carried out was done at a speed of about 50 km/h. This test includes both the implementation of the ACC studied in this thesis project. The variation of speed and following situation based on information about the leading vehicle, and the stop-and-go function. So the approaching situation to a vehicle and the stop and go function were simulated starting from a higher speed than in previous tests. The prototype car was brought to a cruising speed of 45 km/h and after the activation of the ACC through the appropriate button on the cockpit an approaching situation was simulated accordingly to previous tests. The lead car enters in the radar's FoV at a relative distance of approximately 24 m with a much lower relative speed than the ego vehicle. This distance and the huge difference of velocity between the two cars is detected by the ACC as a possible dangerous situation. For this reason at the moment at which the lead car is detected, the host vehicle undergoes a huge deceleration and consequently a drastic decrease in its speed. After the first approach to the lead car there is a slight move away of the vehicle to be follow. In this situation the deceleration imposed by the ACC decrease to 0 in order to try to maintain the actual speed of the prototype vehicle. However the car to follow stops and consequently the ego vehicle stops to avoid a possible collision situation at a distance of about 9 m. When the lead car restart again its motion and the relative distance starts to increase again the prototype car tends to reach the cruising speed of 45 km/h imposed at the beginning.

- d_0 = 12 m
- $t_{h,tar}$ = 1.5 s
- $k_{d,err}$ = 0.15
- k_v = 0.1
- a_{min} = -3 $m/s^2/$
- a_{MAX} = 2 m/s^2

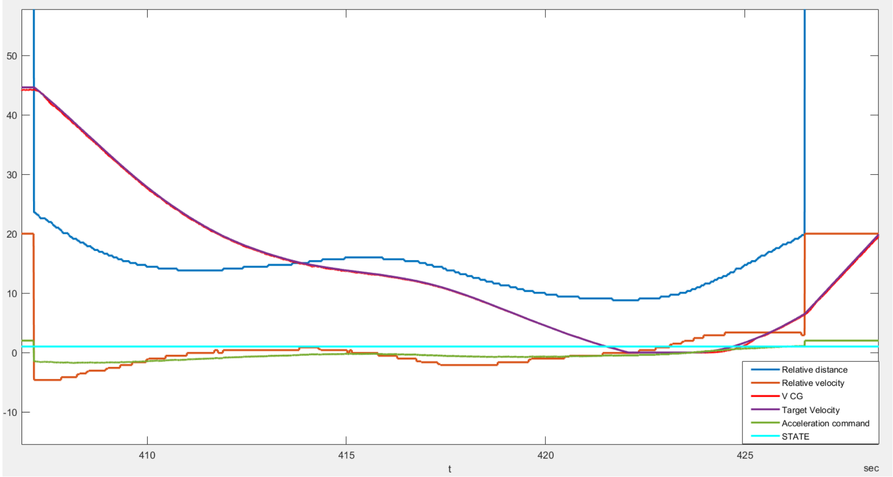


Figure 6.6: ACC and stop-and-go tests at medium speeds

It is possible to notice form the Figure 6.6 that the entire test was done with the ACC activated, this is underline by the "State" line on the graph, which is always at 1 as expressed in "ACC Implementation" on Chapter 4. The ACC impose a fast and smooth response on the prototype car. In fact the ACC impose a big deceleration as soon as the dangerous situation is identified by the radar. It is also possible to see from Figure 6.6 that the desired target velocity imposed by the ACC and the speed of the center of mass of the prototype car are almost the same (red and blue lines). As in the test at lower speeds the ego vehicle stops at distance that is less than the safety distance imposed, also in this case this is due to the gains which influence the effect of the relative distance and the relative velocity $k_{d,err}$ and k_v. However the SCD22 restarts its motion when a relative distance of about 12 m is reached by the lead car.

Time **[s]**	**Relative distance** **[m]**	**Relative velocity** **[km/h]**	**Desired acceleration** $[m/s^2]$	V_CG **[km/h]**	**Desired velocity** **[km/h]**
407.5	21.333	-3.050	0.728	11.8155	11.743
590.5	11.921	-3.058	-0.609	10.436	10.080
593.5	6.58	-0.549	-0.672	1.8576	2.25
595.5	-0.0784	6.588	-0.518	0	0
598.5	7.843	0.8627	-0.2545	0	0
601.5	12.549	1.960	0.516	0.3726	1.716
151.36	15.686	-0.078	0.3346	8.194	8.345

Table 6.3: Data acquisition in a stop-and-go situation

Chapter 7

Conclusions and Future Works

7.1 Conclusion

The implementation of the ACC on the road cars must guarantee more comfort to the drivers in case of traffic congestion, but must also help the pilot in case of possible dangerous situations.
Due to the impossibility to test the radar at very high velocity,(more than 50 km/h) has been done an estimation on the safety conditions guaranteed by the radar at different velocity. The estimation has been done considering the common driver reaction at a possible dangerous situations compared to the reaction of the ACC system implemented.
A person's reaction time in a car can depend on age, gender, health, psychological condition and a possible distraction element used by the driver, such as the telephone, the use of the in-vehicle infotainment etc. The human reaction time must be added to the time needed to operate the braking system on the machines. For this reason has been supposed to have about 1 s of delay between the individuation of the critical situation and the achievement of the signal to the braking system when this is activated directly by the driver. On the basis of this information has been calculated the distance traveled in 1 s at different velocity.

$$d = \frac{\bar{v} \cdot 1000}{3600} \tag{7.1}$$

The same calculation was carried out for the system implemented on the prototype car considering all the systems that interconnect the radar to the ASB. In particular, as mentioned, the radar is able to send information on the position of the leading car at a frequency that is about 400Hz. However this frequency is not

the frequency at which the information are sends to the ASB. This message in fact is then received by the ECU which manages the entire control system of the vehicle and sends the information to the various sensors of the car with a frequency of 200Hz.
The ASB is a mechanical system as such it has a mechanical delay which must be considered on the reaction time of the whole system. After various tests it has been observed that the time which elapses between the request of actuation by the ECU and the effective achievement of pressure on the braking system is approximately 0.74 s. This value is less than human ration time. The distance traveled in the juncture of time is therefore less.

$\bar{v}$ **[km/h]**	**dist [m]**	$dist_{radar}$ **[m]**	$dist_{ECU}$ **[m]**	$dist_{ASB}$ **[m]**
20	5.555	0.015	0.03	4.163
30	8.333	0.022	0.044	6.248
40	11.111	0.028	0.056	8.333
50	13.889	0.036	0.072	10.417
60	16.667	0.043	0.086	12.500
70	19.440	0.050	0.100	14.580
80	22.220	0.058	0.116	16.665
90	25.000	0.065	0.130	18.750
100	27.778	0.072	0.144	20.833

Table 7.1: Distances traveled according to the different reaction times

Despite the delays due to the various systems implemented for the correct functioning of the ACC, the reaction time of the implemented control is less than that of a person. This results in less distance traveled by the vehicle from the moment a potentially dangerous situation is detected until the system starts braking with the desired oil pressure. The distance traveled is indicated in the Table 7.1, where with *distradar* is the distance traveled between one information and another sent by the launchpad, $dist_{ECU}$ is the distance traveled between one information and another sent by the dSPACE and $dist_{ASB}$ is the distance traveled between the request of the pressure by the control and the actuation of it on the braking system.

The communication between the different devices present on a vehicle is one of the most important phase to have a correct manage of the longitudinal and lateral behaviour of a car. In particular when the aim is guarantee a comfort to the driver and avoid possible collision situations which can be critical for the pilot and the passengers of the car. This thesis project focused on the implementation and integration of an ACC system on a prototype vehicle. The main work was to understand decode and extrapolate information about objects located in the sensor's field of view and how to send this information to the ECU. On the basis of this information, the design of the adaptive cruise control functionality has been deliberately simplified in order to focus more on the information's flow from the sensor to the ECU, and on its integration on the prototype car. The structure of the communication implemented allows a very fast communication between the AWR1642 and the ECU. However the action on the different devices is limited due to the low frequency needed to control all the other systems. However the results obtained from the control implemented, guarantee a smooth and fast response by the system in different driving scenario and a good working principle at slow and medium velocity. This approach has no major disadvantages in the automotive domain.

7.2 Future works

In order to make the architecture more efficient for more complex driving scenarios a possible sensor fusion between the other sensors is needed like stereo camera and lidar. This allow to not limited the application of the implemented ACC only in a straight road condition but also to a road with several curves. The sensor fusion guarantee the possibility to recognize in a better way the car presents on the same lane of the host vehicle and the cars which are present in the other lane with a good weather conditions, however the implementation of the latter sensors could not be useful in case of particular weather conditions.

Another possible improvement can be made on the communication, in particular a direct communication between the sensor and the ECU could guarantee a more fast and robust communication. The direct communication can reduce the possibility to lost some data due to possible noise or disturbances to which is subject the serial UART communication established between the radar and the Launchpad.

The last but not the less important improvement can be an optimization of the control implemented for the ACC. The actual control implemented to manage the longitudinal behaviour of the car is a simple control which depends on only two gains. A possible improvement can be the use of a Fuzzy logic or a MPC to have more fast and smooth response on the inputs received by the ACC.

Bibliography

[1] Anders Cioran. «System Integration Testing of Advance Driver Assistance Systems,» in: (2015) (cit. on p. 1).

[2] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Cristoph Stiller, and Herman Winner. «The Decades of Driver Assistance Systems: Review and future Perspectives». In: (2014) (cit. on p. 1).

[3] Azim Eskandarian Aedalan Vahidi. «Research Advances Intelligent Collision Avoidance and Adaptive Cruise Control,» in: (2003) (cit. on p. 1).

[4] Feng Gao Lingyun Xiao. «A comprehensive review of the development of adaptive cruise control systems,» in: (2009) (cit. on p. 1).

[5] Fabrizio Berizzi. «I sistemi di Telerilevamento Radar». In: (). URL: `2015th%20ed.` (cit. on p. 4).

[6] Emre Kural, Tahsin Hacıbekir, and Bilin Aksun Güvenç. «State of the Art of Adaptive Cruise Control and Stop and Go Systems». In: (2020) (cit. on p. 21).

[7] Venhovens P.and Naab K.and Adiprasito B. «Stop and go cruise control. International Journal of Automotive Technology Vol. 1, No. 2». In: (2000) (cit. on p. 21).

[8] Vahidi A. and Eskandarian A. «Research advances in intelligent collision avoidance and adaptive cruise control. IEEE Transactions on Intelligent Transportation Systems Vol. 4, No. 3». In: (2003) (cit. on p. 22).

[9] Rajesh Rajamani. «Adaptive Cruise Control». In: (2014) (cit. on p. 23).

[10] «REGOLAMENTO PER L'ESECUZIONE DEL TESTO UNICO DELLE NORME SULLA DISCIPLINA DELLA CIRCOLAZIONE STRADALE.» In: () (cit. on p. 53).