ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

MASTER EN INGENIERÍA INDUSTRIAL

Título del proyecto:

“Development of a dynamic object classification module and a speed braking function while free driving”

Mirian Rodríguez Romero

Jesús Corres Sanz

Pamplona, Septiembre 2016

Abstract

Avoiding collisions is one of the most important issues in autonomous driving systems. The main goal of this system is to detect objects with potential collision risk and determine if the brake has to be triggered in order to avoid or mitigate a collision.

The two major purposes of the thesis are: first to investigate and determine the current status of the object detection system in order to further develop and implement a dynamic object classification module that can improve the emergency braking decisions when the system has to deal with dynamic objects and second, to propose a new function called “Speed braking” with which the brake system can be triggered on dynamic objects driving with lower speed by calculating a suitable deceleration in each situation.

In the first part of the thesis, a description of the actual driving assistance system is presented with a more detailed explanation of the braking system. This provides a global overview of the whole application showing its complexity and limitations.

Second, the actual weaken cases of the brake system are presented and consequently the module proposed to solve them. In addition, the development of the new function is explained and for both targets the results are shown. These results reveal the dramatic improvement of the brake system. There were 91 real traces supplied by three different clients where the expectation was not to brake what means that the scenario was dynamic and 62 of them were wrong. With the new algorithm 60 of those 62 traces were solved, which means an improvement of 66% in dynamic scenarios. From a general point of view, including static and dynamic situations, the solution developed represent an improvement of 39% which represents an overall success of 99% of the traces analyzed. Moreover a speed braking function depending on the speed of the vehicle driving in front of the host car is implemented, working without problems in a range of speeds 2-12 km/h while driving behind a car or a bicycle.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DAS	Driving Assistance System
ULS	Ultrasonic
OBJD	Object Detection
UFEX	Ultrasonic Feature Extraction
BSD	Blind Spot Detection
ECU	Engine Control Unit
P4U	Park For You
UPA	Ultrasonic Parking Assistance
ULSD	Ultrasonic Data Layer
ULFM	Ultrasonic feature Extraction Master Module
FILA	Feature Interpolation
MOPL	Movement Planner
COLM	Collision Avoidance
VCTL	Vehicle Control
VDIL	Vehicle Driver interface
UFSP	Ultrasonic Freespace
SECT	Sector
PMRG	Point Merge
PCLU	Point Cluster
SDET	Slot Detection
ASIC	Application Specific Integrated Circuit
ODOM	Odometry
SGWF	Signal Way Filter
UFEX SSP	Single Sensor Feature Extraction
UFEX MSP	Multi Sensor Feature Extraction
IEBM	Emergency Braking Module
THAS	Thread Assessment
COBR	Comfort Braking
MEXP	Exploration Mode
TTC:	Time To Collision
DTC:	Distance To Collision

1. Introduction

Today's automobile represents the most sophisticated technology owned by most consumers, and automakers continuously offer new high-tech content in their products. From the early stages of planning, automakers modernize new vehicles, recognizing that technology provides many solutions to meet consumer expectations of a vehicle such as quality, reliability, safety, and utility maintaining or even reducing the vehicle´s price.

Nowadays, auto technology on sale allows cars to "see" all around, gathering data on possible roadway concerns and giving drivers eyes in the back of their heads. Since more than 90 percent of crashes involve driver error, automakers created a range of safety systems that aid drivers for brief periods to help avoid accidents. Driver assist systems include lane departure and blind spot warnings, adaptive cruise control, automatic braking, telematics control systems and more.

New driver assist systems rely on a range of technologies working together:

- Ultrasonic sensors use high frequency sound to measure distances between objects
- Radar determines which objects are ahead of a vehicle by sending out and then retrieving radio waves, and comparing the difference between the two
- Lidar detects objects by enveloping an area with invisible laser light and analyzing the reflected results. When combined with cameras and other sensors, lidar can create a 3-D view of the environment around a vehicle

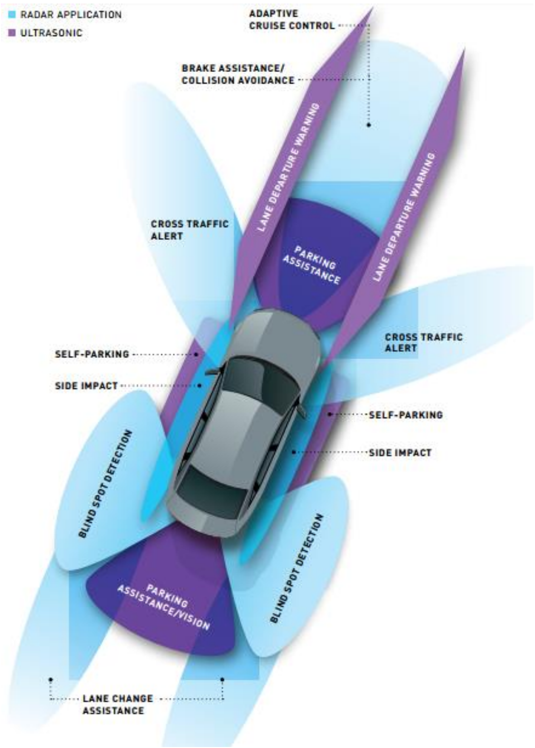


Figure 1.- New driver assist systems

What once only existed in the imaginations of science fiction writers is now being developed and tested by carmakers in laboratories and on roadways across the globe. As partially-autonomous functions in vehicles become more common, the leap to achieving fully driverless cars becomes ever smaller.

1.1. Goal of the thesis

The objective of this thesis is to study the actual system behavior, including how the system creates and detects objects using the information provided by the ultrasonic sensors located in a car in order to develop an algorithm for dynamic object detection. Moreover, it would be necessary to analyze how the brake functionality operates in order to improve it.

Therefore in this thesis, on the one hand it will be studied how to combine the actual system with a new dynamic object classification and on the other hand, the braking performance will be adapted considering the new object detection classification.

2. Valeo

2.1. Group Profile

Valeo is an automotive supplier based in France, partner to all automakers worldwide. As a technology company, Valeo proposes innovative products and systems that contribute to the reduction of CO2 emissions and to the development of intuitive driving.

At December 31, 2015, the Group had 134 plants, 17 research centers, 35 development centers and 15 distribution platforms and employed 82,800 people in 30 countries worldwide.

2.2. History

- **1923**
  Incorporation of Société Anonyme Francaise du Ferodo (the “Company”). The company started by distributing, then manufacturing, brake linings and clutch facings under the Ferodo license out of a workshop in Saint-Ouen, near Paris.
- **1962**
  First diversification into heating and air conditioning systems following the merger with SOFICA (Société de Fabrication Industrielle de Chauffage et d´Aération), which was reinforced by the acquisition of Chausson´s heat exchanger business in 1987.
- **1960s and 1970s**
  Time of development into new business sectors (thermal systems in 1962, lighting, wiper and electrical system in the 1970s) and into international markets (Spain in 1963, Italy in 1964 and Brazil in 1974).
- **1980**
  The name Valeo, a Latin word meaning “I am well”, is adopted with a view to uniting the various brands and teams under a single name.

- **1980s**

  Expansion into de United States.

  Acquisition of new systems: security systems, wiper systems, climate control systems, brake linings systems etc.

- **1990s**

  The Group's external growth continued throughout the decade.

  - Implementation of the "5 Axes" method in 1991
  - Set up of electronics research centers in France
  - First plants in Mexico and Wales
  - First joint ventures in China (1994)
  - Acquisitions in Germany, Argentina, Italy, Czech Republic, United States etc.

- **2000s**

  The Group implemented a program of industrial streamlining with production reorganized across fewer sites, a greater portion of sites in cost-competitive regions, and the sale of selective non-strategic activities.

  - Acquisition of Labinal's automotive business (Argentina, Eastern Europe, Italy, Spain, France, India, North Africa and Portugal)
  - Development in Japan
  - From 2004, the Group focused on technology through targeted acquisitions, while accelerating its expansion in Asia, particularly China.
  - Start up operations in Russia.

- **2010s**

  - Acquisition of a Japanese automotive supplier, which enabled the Group to become world leader in the interior controls market by reinforcing its presence in Asia.
  - Formation of joint ventures in USA

2.3. Business activity structure

Valeo is made up of four Business Groups and an aftermarket activity, Valeo Service.

The Comfort and Driving Assistance Systems	*Powertrain Systems*	*The Thermal Systems*	*The Visibility Systems*
•18% of the Goup´s sales •26 production sites •8 Development centers •9 Research centers •15.300 Empoyees	•26% of the Goup´s sales •36 production sites •15 Development centers •6 Research centers •19.400 Empoyees	•28% of the Goup´s sales •48 production sites •10 Development centers •3 Research centers •20.100 Empoyees	•28% of the Goup´s sales •35 production sites •15 Development centers •5 Research centers •26.300 Empoyees

Figure 2.- Business Groups of Valeo

2.3.1. Comfort and driving assistance systems business group

The mission of the comfort and driving assistance systems business group is to develop interface systems between the driver, the vehicle and the environment, which help to improve comfort and safety.

It focuses on intuitive driving, with four complementary priorities: easy, ergonomic interaction with the vehicle for the driver, driving agility with better visibility of the surrounding environment, connectivity, and safe, personalized access.

Organization of the comfort and driving assistance systems

Comfort and Driving Assistance Systems has three Product Groups:

- Interior Controls
  - *The smart faceplate*

    The intelligent faceplate is the solution developed to help carmakers achieve the complex task of reconciling the ever-increasing number of functions to operate with the need for man-machine interfaces.

Figure 3.- Smart faceplate

This product is included in cars such as: Aston Martin Vanquish and Peugeot 508.

- *Top Column Modules (TCM)*

The functional diversity of the new Valeo TCM modules means they can be adapted to innovative automotive architectures with the flexibility of modular units and a highly developed integrated design.

Figure 4.- Top Column Module

In addition to its key indicator and wiper functions, the TCM module acts as an interface with the steering wheel. New functions such as steering wheel heating, bend lighting and active steering improve driver comfort and ease of use.

- *The smart Screen*

The smart screen provides infotainment functions in all vehicles and is an optimum solution in terms of cost and size.

Figure 5.- Smart Screen

The smart screen gives constant in-car access to the latest functions while taking advantage of interfaces that are fully optimized for the driving environment.

- Driving Assistance

- *Rain-Light-Humidity Sensor*

Valeo, one of the leading suppliers of rain sensors, has developed this technology further to create a truly multifunctional sensor.

Figure 6.- Rain-Light-Humidity Sensor

In addition to making things easier for the driver, the aim is to make driving safer and reduce fuel consumption. The sensor not only measures the volume of rain: it also automatically activates and deactivates low-beam lamps depending on ambient light levels. Tunnel entrances are recognized in advance, so that lights are turned on in time.

- *360VUE Surround View System*

The new 360Vue® 3D system gives not only full visibility right around the vehicle in a view from above, but also a 3-dimensional view of the vehicle's environment. The driver can perform maneuvers or leave a parking space safely, with excellent visibility of all obstacles in the vicinity of the vehicle and the capacity to see into all blind spots.

Figure 7.- 360 View System

- *Lane change assistance system*

Valeo's Lane Change Assist system alerts the driver if there are vehicles in this area or vehicles approaching rapidly from the rear. This information is extremely valuable, especially if the driver suddenly decides to change lane.

- *Laneguide*

Unintentional lane departures are often the cause of dangerous situations and accidents. Valeo's LaneGuide® helps avoid these situations by alerting the driver before the vehicle crosses the line, and even by gently bringing it back into the correct lane.

Figure 8.- Laneguide

The system uses a compact camera located behind the interior rearview mirror, on the windshield. This camera can also perform a number of other impressive functions. In addition to the LaneGuide™ feature, it can activate Valeo's BeamAtic system, interpret road signs, and even warn the driver when the distance between the vehicle and the one in front becomes critical.

- *Park4U:*

Park4U® parks a vehicle in just a few seconds. Ultrasonic sensors scan the sides of the road to detect a suitable space. The parking maneuver takes place in the usual way, but in

Figure 9.- Park4U

hands-free mode. As soon as the car has stopped and the reverse gear has been engaged, the system takes over the steering, while the driver continues to control the speed of the vehicle with the accelerator and brake. During the parking maneuver, ultrasonic sensors in front and to the rear help the driver to use the available space as efficiently as possible, providing additional security. If the driver so wishes, the maneuver can be ended at any time: as soon as the steering wheel is touched, the system automatically deactivates.

- Interior Electronics

- *Passive entry passive start system*

Valeo’s hands-free access and start system allows the user to automatically lock and unlock the car doors without taking the key out of a bag or pocket.

The engine is started simply by pushing the ignition button.

To lock the vehicle, the user just moves away from it with the key, or touches a button or specific area on one of the door handles, depending on the ergonomic solution chosen by the manufacturer.

Figure 10.- Passive entry start system

This product is included in cars such as: Citroen DS5 and Peugeot 508.

- *Telematics module*

Valeo's telematics solutions incorporate geolocation (GPS, Glonass) and mobile telecommunications (2G/3G/4G) technologies, in a single standalone module, to provide the maximum number of connected in-vehicle services and meet new safety and security regulations such as eCall, Era-Glonass and Stolen Vehicle Recovery.

Figure 11.- Telematics module

In the event of an accident, the integrated telematics module automatically sends an emergency signal to a help center, which attempts to contact the driver to provide the best possible assistance as quickly as possible. In the event of a breakdown, a call can also be sent to an assistance center.

2.3.2. Powertrain Systems business

The mission of the powertrain systems business group is to develop innovative powertrain solutions aimed at reducing fuel consumption and co2 emissions, without compromising on the pleasure and dynamics of driving.

These innovations cover a complete product range, from the optimization of internal combustion engines to the varying degrees of electrification of vehicles, from stop-start systems to the electric car.

Organization of the powertrain systems business group

Powertrain Systems has four Product Groups:

- Electrical Systems
- Transmissions Systems
- Combustion Engine Systems
- Electronics.

2.3.3. Thermal systems business group

The mission of the thermal systems business group is to develop and manufacture systems, modules and components to ensure thermal energy management of the powertrain and comfort for each passenger, during all phases of vehicle use.

These systems help to significantly reduce fuel consumption, co2 emissions and other pollutants and harmful particles from vehicles equipped with internal combustion engines. They also increase travel range and battery life for hybrid and electric vehicles.

Organization of the Thermal systems business group

Thermal Systems has four Product Groups:

- Climate Control
- Powertrain Thermal Systems
- Climate Control Compressors
- Front-End Modules

2.3.4. Visibility systems business group

The role of the visibility systems business group is to design and produce efficient and innovative systems which support the driver at all times, day and night, offering perfect visibility and thereby improving the safety of both driver and passengers.

Organization of the Visibility systems business group

Visibility Systems has two Product Groups:

- Lighting Systems
- Wiper Systems

3. System architecture

The aim of this chapter is to give a global overview of how the actual system works.

3.1. Driving Assistance System (DAS)

3.1.1. Definition

In order to explain how the application is made, it is important to remark the definition of a DAS system which is the following:

Driving Assistance Systems (DAS) are systems that support the driver in his driving. This assistance is received in terms of information, warnings, support (partly autonomous intervention), fully autonomous intervention, or a combination of these, where the distinction between them sometimes is subtle.

3.1.2. Functions

The Driving Assistance System of Valeo has been improved during the years and in the same way have the features of its application.

In the following tables it can be seen all the different features, both direct and implicit, of Valeo´s system from the first generation until the currently DAS (DAS 3.3).

- Direct Features:

Table 1.- Direct Features of the application

Feature	Gen 1	Gen 2	DAS 2.50	DAS 3.1	DAS 3.2
Parallel Parking	Yes	Yes	Yes + longitudinal alignment	Yes	Yes
Perpendicular Parking forward/ backward with passing the slot	No	Yes only backwards	Yes only backwards	Yes	Yes
Perpendicular Parking forward without passing the slot	No	No	No	Yes	Yes
Park strategy selectable	No	Yes	Yes + virtual curb	Yes	Yes
Park me Out Parallel	No	Yes	Yes	Yes	Yes
Park me Out Perpendicular	No	No	No	No	Yes
Remote Parking (Longitudinal control)	No	No	No	No	Yes
Exploration mode	No	No	No	No	Yes
Parking Roadmarks (Line Parking)	No	No	No	No	Yes
Undo perpendicular parking	No	No	No	No	Yes
Garage Parking	No	No	No	No	No
Valet Parking	No	No	No	No	No
Warning Optical/Acoustical	Yes	Yes	Yes	Yes	Yes
Warning Side (Flang Guard)	No	No	Yes	Yes	Yes
Warning haptical feedback	No	No	No	Yes	Yes
Predictive Warning	No	No	No	No	Yes
Virtual rut	No	No	No	Partly	Yes
Collision Mitigation	No	Partly while parking	Yes	Yes	Yes
Collision Mitigation pedestrian (PD camera)	No	No	No	No	Yes
Comfort braking	No	No	Partly while parking	Partly + Improved by DTC + Distance to Hint	Yes
Blind Sport Detection based on ULS	No	No	No	Yes	Yes
ULS Ice/Mud Detection	No	No	Yes	Yes	Yes

- Implicit features:

Table 2.- Implicit features of the application

Feature	Gen 1	Gen 2	DAS 2.50 (2013)	DAS 3.1 (2014)	DAS 3.2 (2015)
Odometry	Yes	Yes	Yes Improvement 4 wheel odometry	Yes	Yes
Environmental Map	No	No	Partly	Yes fully	Yes fully
Fusion	No	No	No	No	Yes +camera
Sensor Control	Partly UPI,16 bit discrete	Partly UPI,16 bit discrete	Partly UPI,UPIT,16 bit discrete, Ak sensor	Yes fully UPI,UPIT, full AK and HP sensor, Temp dependant thresholds	Yes fully
ULS Multi Core Architecture	No	No	No	Yes fully	Yes
Trajectory panning	Yes	Yes only ULS side sensors considered	Yes only ULS side sensors considered	Yes +map+ Improved by safety param	Yes +map+ Improved by safety slot
Steerable rear axis	No	No	No	Yes	Yes
Park slot detection	Yes using only side sensors	Yes using only side sensors	Yes using only side sensors	Yes Map base	Yes Map base
Submerged Features	No	No	Partly	Yes Map base	Yes Map base

Focusing in the current DAS generation (generation 3), the new features achieved comparing with previous generations are described below and can be divided in four categories:

1. **New Object Detection (Map based OBJD_3)**

Up to generation 3, the whole application was based on the raw data provided by the ultrasonic sensors.

With the introduction of the Map, not only the object detection changes dramatically, but also it allows the application to combine sensor data of different origin (sensor fusion).

A briefly description of how the Map works would be: the raw data is interpreted by the ultrasonic feature module (UFEX) which provides the Map points and lines features. With them, the map is in charge of deleting, merging in case of excessive similar information, apply freespace and create clusters.

Furthermore, once the clusters are created, they are supplied to the slot detection module.

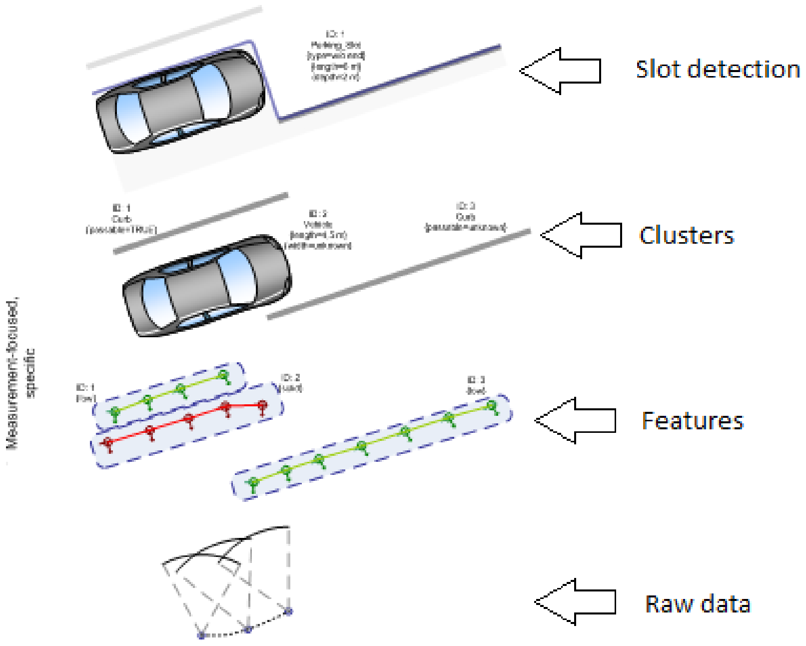


Figure 12.-New Object detection architecture

2. Parking functions:

2.1. *Perpendicular Forward without passing*

In previous generations the parking function was only available once the host car has passed the slot. This means that the system could analyze the environment, detect the slot and be in a “controlled” environment.

With Generation 3 the perpendicular forward parking was improved including in the system the use case where the driver enters into the perpendicular forward parking slot without previous passing.

In the following figures it is shown both the start conditions and the parking maneuver:

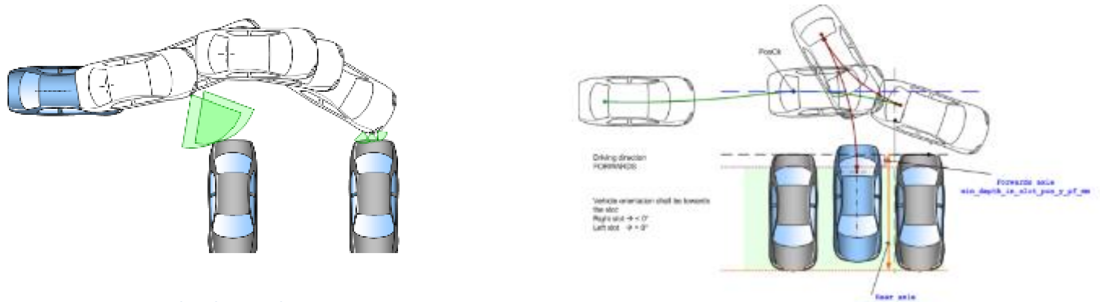


Figure 13.- Perpendicular parking maneuver

2.2. *P4U with fusion of ULS and Vision*

In order to detect slots, it has to be clarified that ultrasonic sensors can mostly detect nothing from the internal slot while passing it. Therefore it can be detected the beginning and the end of the slot assuming that it is a parking slot but it can not be assured that within it there is any object.

Including the Fusion of ultrasonic sensors and Cameras the camera can look inside the slot which improves slot detection dramatically.

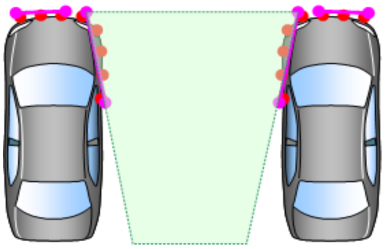


Figure 14.- Slot detection with only ULS

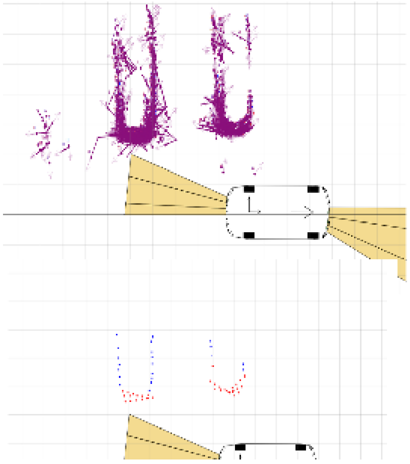


Figure 15.- Slot detection with ULS + Camera

2.3. *Park me Out perpendicular*

The improvement achieved in the park me out function is that the driver can target the orientation of the park out maneuver.

This new option is included for both perpendicular backward and forward parking slots.

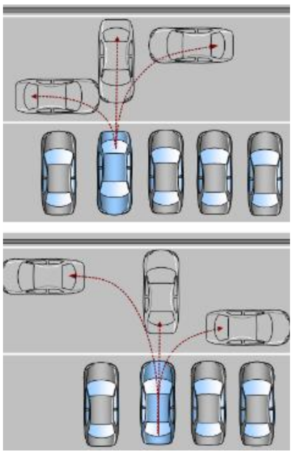


Figure 16.- Perpendicular Park me Out

2.4. Undo Parking

While parking, aborts can occur, that is why this new feature was included in the latest generations.

The Undo function plans the trajectory to move the car to the park start position after an abort. This feature enables the driver to start again the parking maneuver from the same start position.

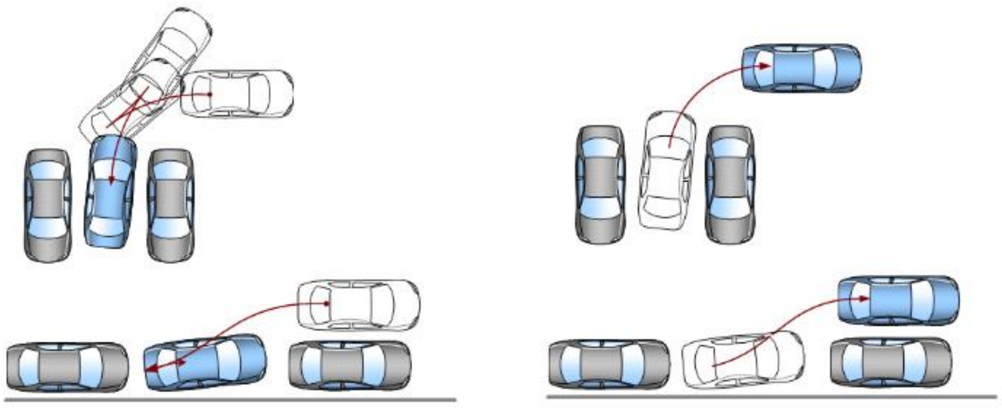


Figure 17.- Undo function

2.5. Remote Parking

This new feature enables the customer to park automatically from outside of the car controlling it with a rich interface on a smart phone. The communication between car and Smartphone is achieved by Bluetooth Smart. It is based on lateral control of the car and longitudinal parameter calculation.

The parking maneuver is produced at the same time as the driver touches the screen of his Smartphone doing a specific figure. As soon as the driver stops touching the screen, the car also brakes. It only takes 300-400 ms to the system to brake what makes the function very reliable.

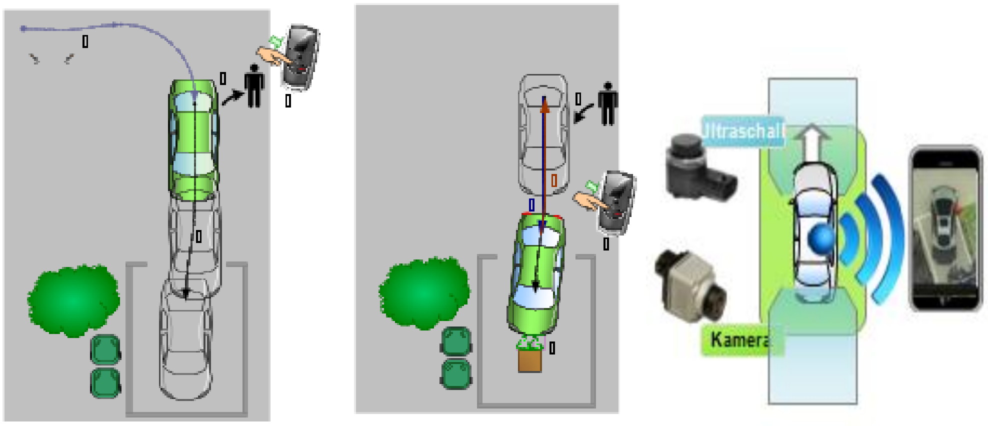


Figure 18.- Remote parking

2.6. *Parking based on road-mark detection*

In order to improve the parking function, road-mark detection has been included in the latest versions of the application.

This new feature provides the system additional information for the scene analysis and allows covering a big variety of scenarios. It improves either in park slot recognition or the parking results or both at the same time.

The following figures show different user cases in which the road-mark detection has an important impact.

Figure 19.- Road mark detection

As it can be seen in the third example, there is no object delimiting the parking slot that is why only the information provided by object detection is not enough for parking and the road mark detection solves this situation.

Moreover, it can be appreciated that the parking results are also improved.

2.7. Exploration Mode with fusion of ULD and Vision

Exploration mode is the function that checks if there are obstacles in the path of the vehicles. If there are, it computes the required wheel angle to try to avoid them.

This function is available in backwards and forwards direction and including the fusion between ULS and Vision, not only additional information for the scene analysis is gained, but also increase the detection area, plan a better steering behavior and react better on pedestrians.

Figure 20.- Exploration mode

2.8. Garage parking

The goal of “Garage Parking“ (GP) is to park in and out of a garage or park slot/area automatically with the driver supervision from outside of the vehicle.

In order to do it, the application combines forward and backward exploration mode. This function needs sensor fusion with ultrasonic sensors and cameras with either corner radar and/or Scada/IR LED because in a garage there are many visual and spatial characteristics that have to be detected.

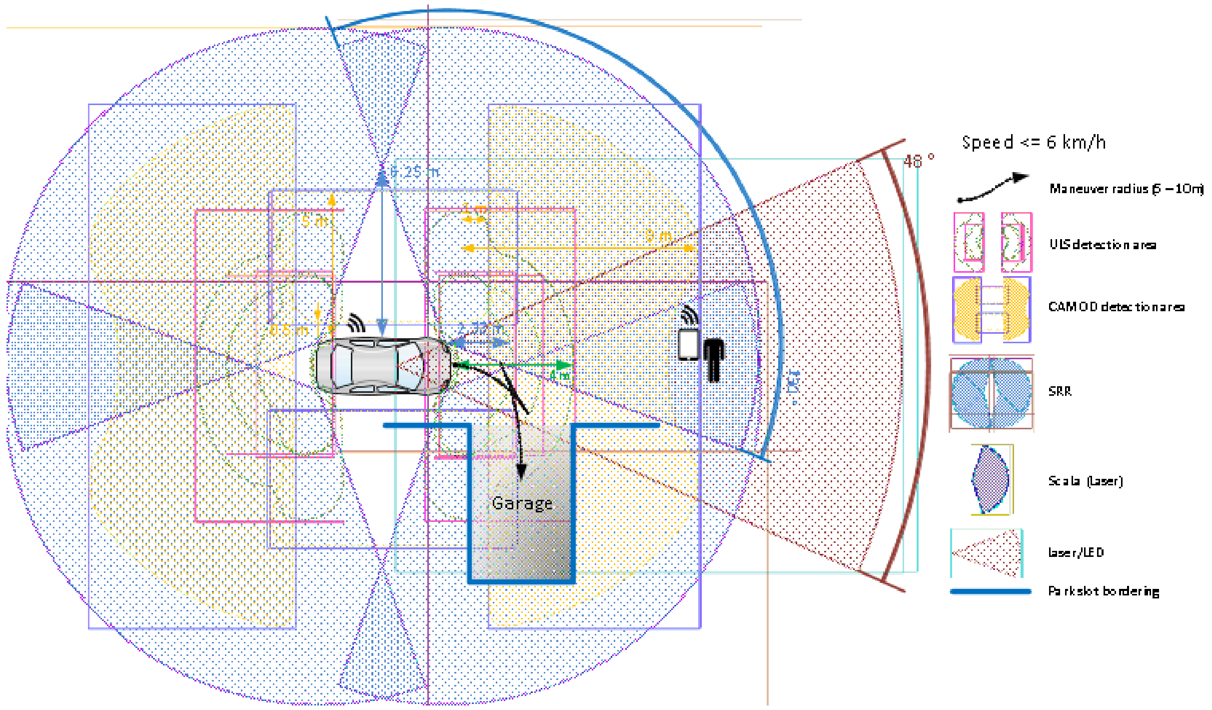


Figure 21.- Garage parking

2.9. *Valet parking (in development)*

The idea of Valet parking consist on driverless driving from the transfer zone of a parking house to the parking space (and back) with the driver outside of the car.

This system should be able to achieve the followings functions:

- Capability to Park
- Respect driving rules
- Driveless driving (Movement)
- Capability to find the park slot
- Ensure finding the way back
- Brake function
- Object detection

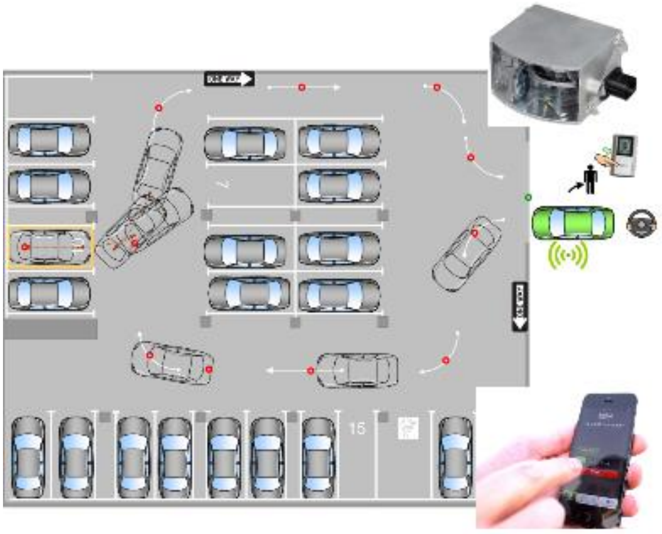


Figure 22.- Valet parking

It has to be mention that this function can not be achievable only with ultrasonic sensors. Laser scanner and camera are needed.

3. Warning, Braking

3.1. Haptical Steering Feedback

The driver can be warned by the system by different ways: acoustical, optical and haptical. The module in charge of warning the driver is called SECTOR and it should only warn the driver if there is any object potentially dangerous in the driving tube.

Haptical warnings imply that the system has detected an obstacle within the driving tube and outside of the “hard to avoid area” and it will try to find a safe driving radius. If there is a risk to collide, a slight steering torque is applied (only if driver has the hands on the steering wheel) that pushes away from the obstacle.

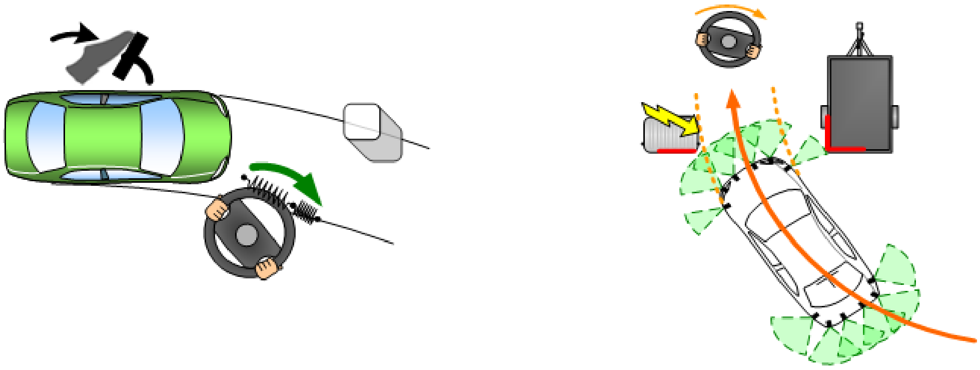


Figure 23.- Haptical steering warning

3.2. Virtual rut with Fusion of ULS and Vision

The system detects a parking slot in front of it and tries to find a safe driving trajectory to the goal position and the orientation. A steering torque is applied to guide the driver into the parking slot. Further in this scene the system shall avoid collisions with the environment by emergency braking.

The trajectory is applied for the forward move. For the backward movement Haptic functionality is applied. In all following forwards steps steering torque will be applied immediately.

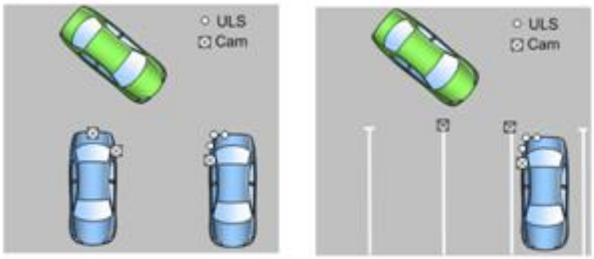


Figure 24.- Vision and Ultrasonic Fusion

3.3. Blind Spot detection (BSD)

The blind spot detection is a function made to warn the driver of approaching vehicles by turning on a signal in the side mirrors of the car. This feature can be achieved with only the side ultrasonic sensors of the rear and front.

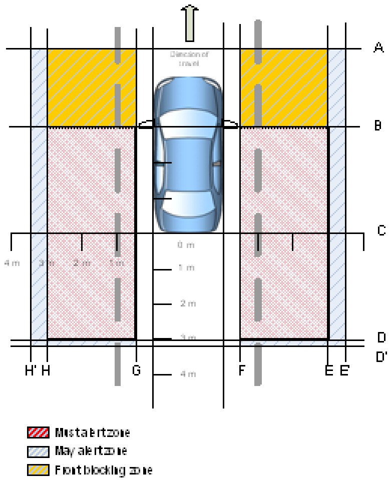


Figure 25.- Blind spot detection

3.4. Collision Mitigation with Fusion of ULS and Vision

This function makes the system brake on the side, rear or front of the vehicle if a camera or the ultrasonic sensors detect a static obstacle. It can be configured to brake in high, active or tracked obstacles.

The advantage of including fusion in this function is not only that the detection area is increased, but also the robustness of the system because there are two different sensor systems with its own algorithms.

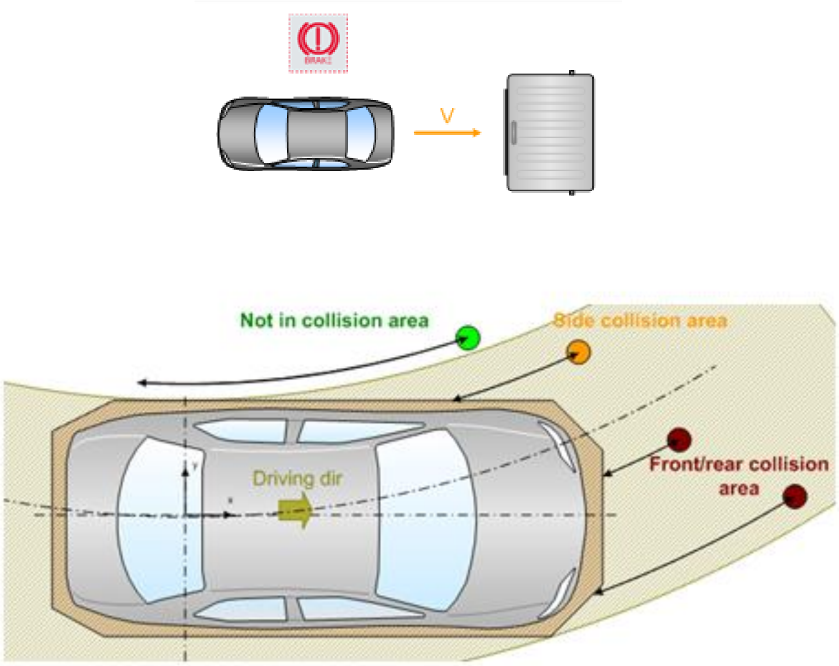


Figure 26.- Collision mitigation with fusion

Further information will be provided in the brake system chapter.

3.5. Brake on Pedestrians with Fusion of ULS and Vision

Similar to the previous feature, braking on pedestrian is a very difficult target to achieve only with ultrasonic sensors, that is why including fusion it could be reached.

The goal of this function is to brake automatically to standstill when the host car gets too close to a pedestrian that is located within the driving tube.

During development, the first priority of the system is to brake on pedestrians that are on the rear of the car and the second would be for pedestrians on the front.

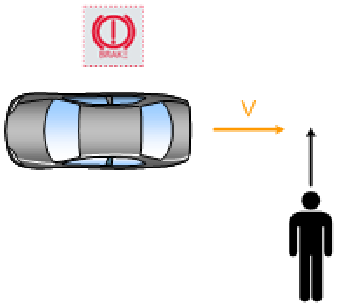


Figure 27.- Brake on pedestrians

3.6. Comfort Braking

By providing the rest way until stop point, it is possible to slow down the vehicle smoothly.

Driving tube filtering: The system is able to filter out objects which are not inside the driving tube, and also objects which are outside the area where a collision is hard to avoid (this area depends on host speed)

Figure 28.- Comfort braking

4. Longitudinal Control and Trajectory Planning

4.1. Trajectory planning safe slot/comfort brake

When the system detects a parking slot it also creates uncertainty border that create what is called safe slot. In order to execute the parking maneuver, the trajectory planning programmes it according to the detected slot plus the safety values. Moreover, comfort braking uses also the safe slot borders and considers the planned trajectory.

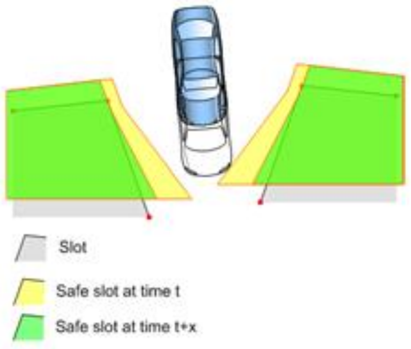


Figure 29.- Trajectory planning

3.1.3. DAS Architecture

3.1.3.1. Hardware

The system is composed by:

- ECU

  Up to generation 2.50 the ECU used in the application was composed by a 16bits micro controller.

  From generation 2.50 until actual DAS, the ECU uses a 32bits micro controller.

  The characteristics of the ECU are:

  - All DAS functions run on one ECU
  - Valeo housing and connector available,
  - customer specific possible
  - Different channel system: High speed CAN interface or Flexray
  - LVDS or Ethernet Interface for Camera
  - Functions: trigger sensors, process sensor signal and trigger speakers.

Figure 31.- 32bits ECU

Figure 30.- 16bits ECU

- Park4U sensors and UPA sensors

  In addition to the ECU, the system is composed by 12 ultrasonic sensors, 10 UPA sensors and 2 park for you sensors.

  The main difference between this two kind of sensors is the opening angle. P4U sensors, which are located in the front bumper, have a smaller opening angle because its function is to detect as accurate as possible the parking slot and by reducing the opening angle the ultrasonic energy is concentrated in less space.

Figure 32.- P4U sensors

  The ultrasonic sensors have the following characteristics:

  - Based on Park Assist sensor 5th generation
  - Longer membrane and thicker decoupling ring than previous generations
  - Unique sensor characteristic
  - Shielded housing
  - Different housing colour and connector coding

The next figure represents how the different components are located in the car:

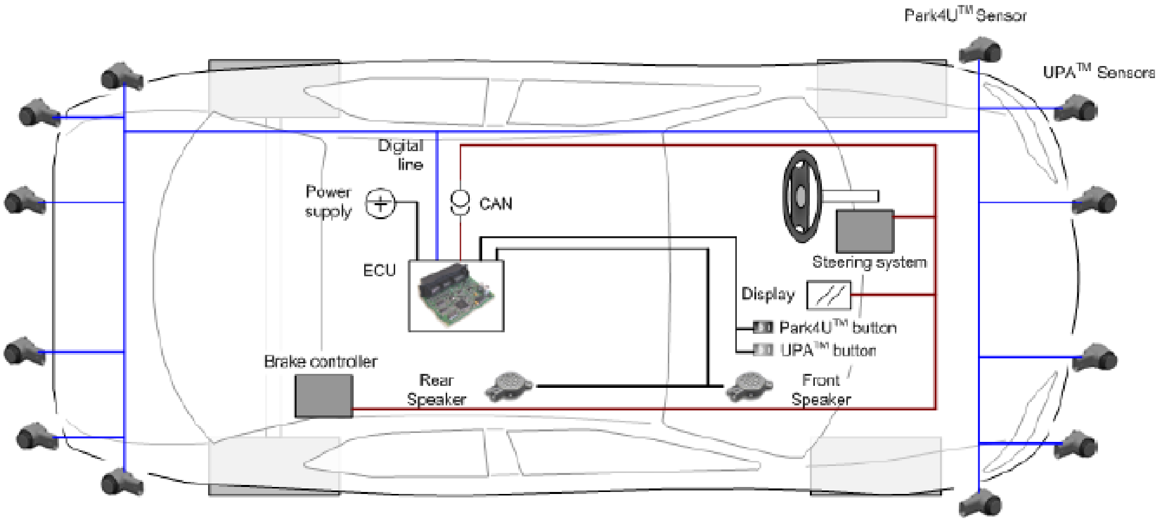


Figure 33.- Scheme of hardware components

In a more exhaustive analysis of the application´s hardware, the next figure shows all the hardware components:

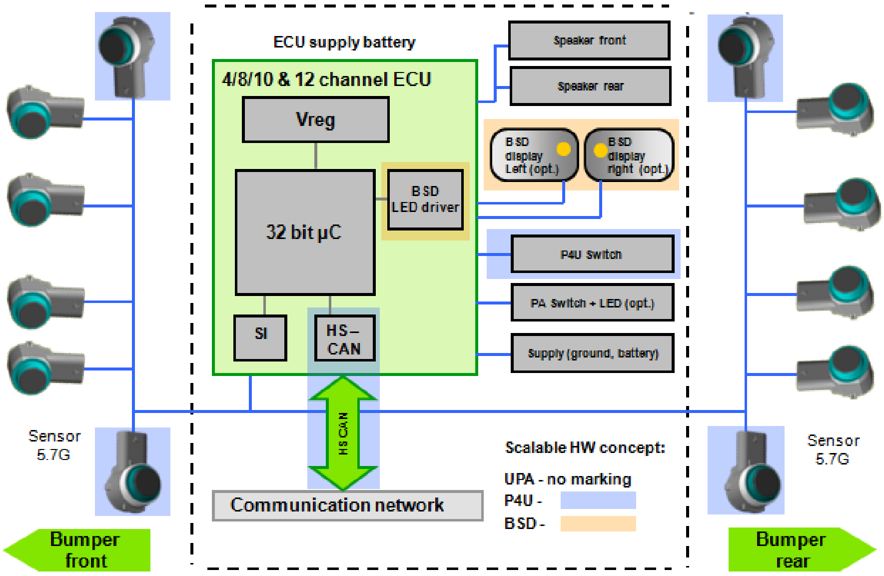


Figure 34.- Hardware components

3.1.3.2. *Software*

To describe how the application´s software is, it has to be mentioned that it is divided in master modules which at the same time are composed of sub modules.

The software architecture of DAS 3.2 is represented by the following scheme:

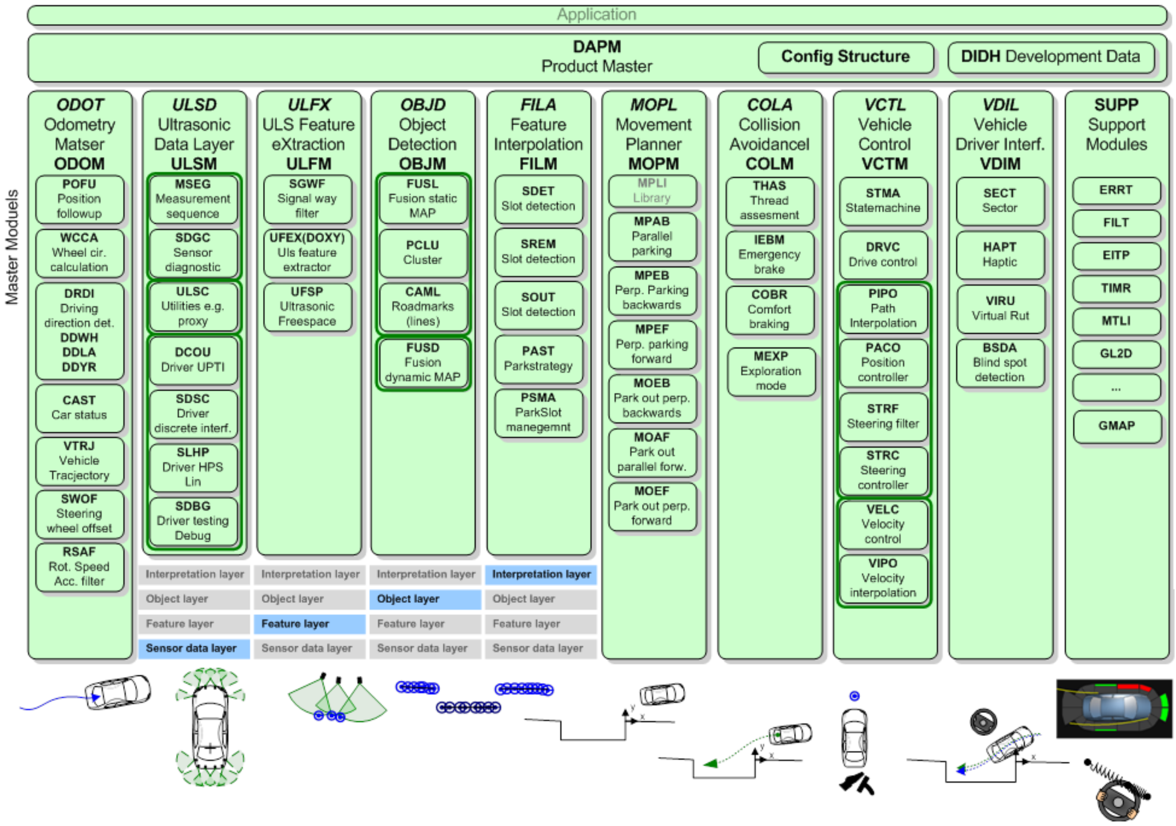


Figure 35.- Software architecture

As it can be seen in the figure above, it can also be distinguished four different layers starting by the sensor layer (ULSD), followed by the feature and object layer, which are composed by the ultrasonic feature extraction module (ULFM) and the object detection module (OBJD) respectively and ending by the higher layer level, which is the interpretation layer where we can find multiple master modules such as: Feature Interpolation (FILA), Movement Planner (MOPL), Collision Avoidance (COLM), Vehicle Control (VCTL), Vehicle Driver Interface (VDIL).

3.2. Object detection

3.2.1. Global Overview

According to the software architecture, in order to detect objects, the four different layers have to be taken into account.

The next figure shows a basic overview of how objects are detected by the system with ultrasonic sensors:

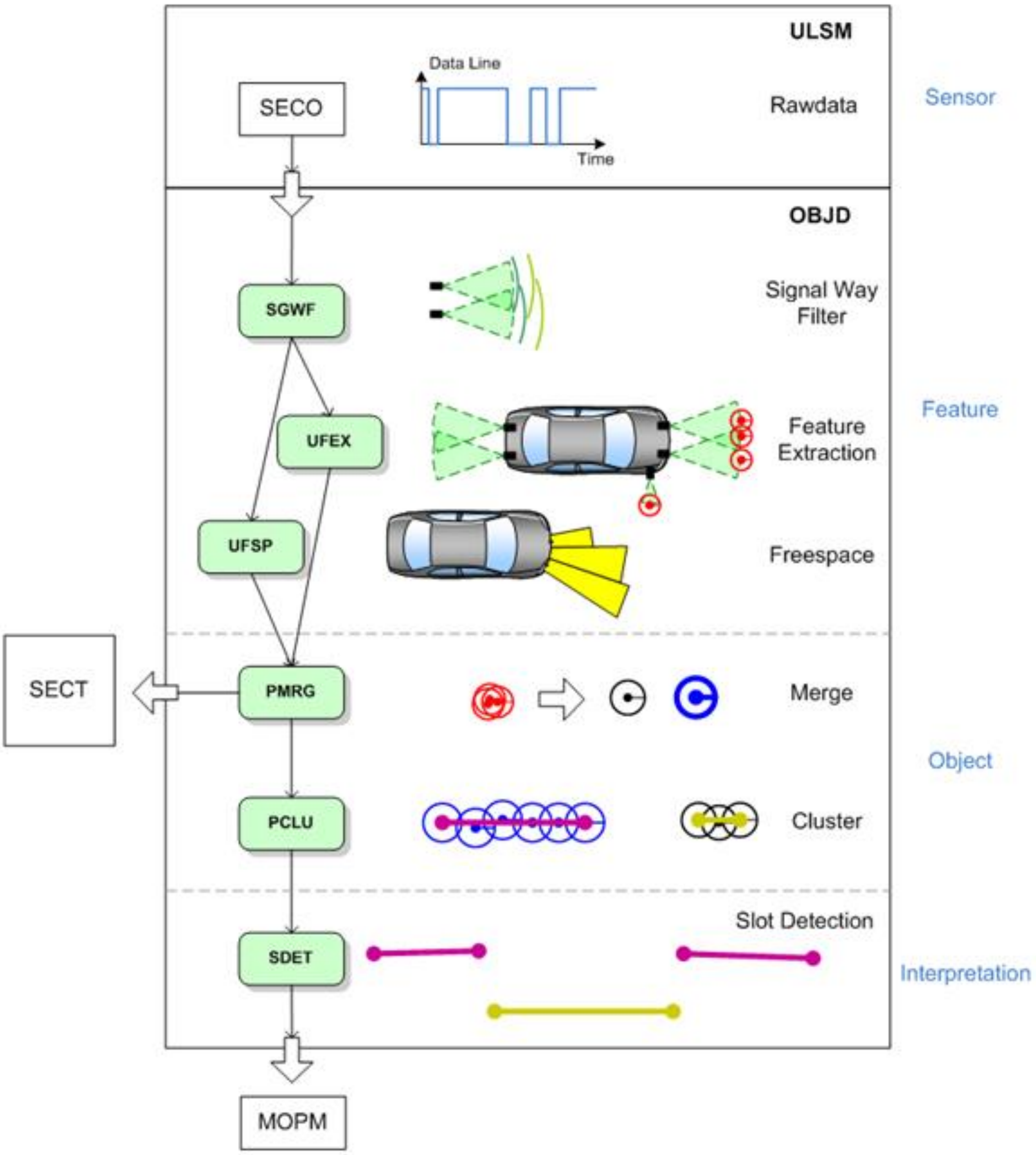


Figure 36.- Object detection´s overview

3.2.2. Ultrasonic Sensors

In first instance, there is the sensor level where the ultrasonic sensors provide the raw data in terms of echoes to the system. The module in charge of this function is the ultrasonic data layer.

The principle of distance measurement with ultrasonic sensors is based on four steps:

- First, the membrane of the sensor vibrates in order to send an ultrasonic wave and the counter starts.
- Second, the sensor listen for incoming ultrasonic signal while the counter still running.
- Third, once the signal is received it has to be processed.
- Finally, the driver has to receive a feedback.

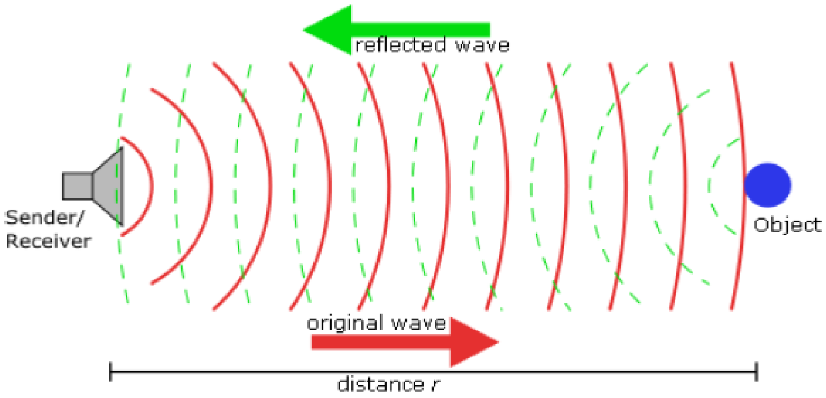


Figure 37.- ULS principle

In the following figure, it can be seen an example of an ultrasonic signal of a sensor.

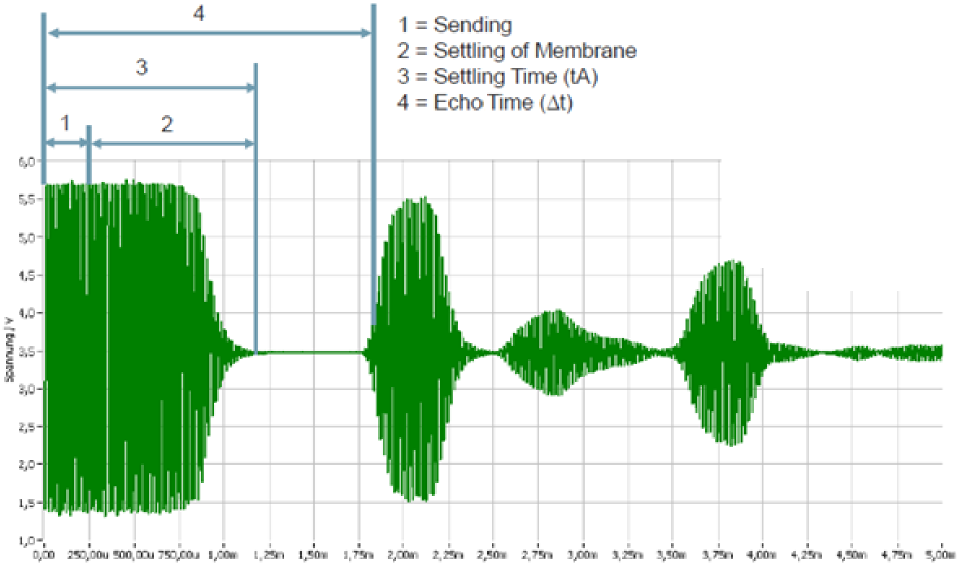


Figure 38.- Ultrasonic signal

At the beginning of the signal, it can be appreciated the transmitting or settling time (3). This tune includes the sending time where the system makes the membrane of the sensor vibrate (1) and then the settling time of the membrane that is the time that the membrane needs to stop vibrating.

As it was explained before, at the moment when the sensor starts transmitting, the counter starts until the sensor receives an echo. This time is called Echo time (4).

In the figure above, it can be seen that the signal has received three different echoes, what means that three different distances can be measured.

In order to calculate the distance, the next formula is applied:

$$Distance = \frac{Echo\ time\ \cdot Sound\ speed}{2}$$

It is important to take into account that ultrasonic sensors have a minimum measurable distance due to the settling time. During this time there can not be any signal received what means that below a certain distance (around 30 cm) the sensors are blind. This area is called blind area.

However, objects are usually seen from far away, what means that with previous information and the odometry of the car the object position can be estimated even if it is located within the blind area.

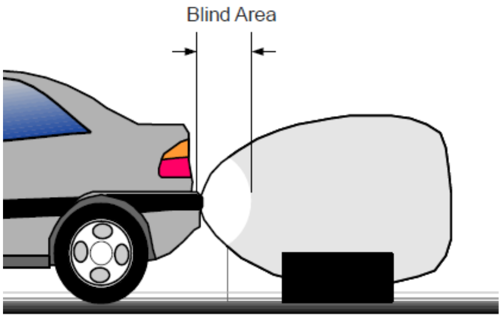


Figure 39.- Blind Area

Moreover, it has to be taken into consideration that not every echo received by the sensor comes from a real object. As an example, there can be reflections originated by the ground or noise echoes. In order to filter these signals, there are thresholds applied. The thresholds are generated in the sensor ASIC, stored in the ECU and downloaded into the sensor after power-up. As the reflections received are distance dependant, the threshold also varies with the obstacle distance.

The next figure shows how both, the threshold and the signal change depending on how far is the obstacle from the ultrasonic sensor.

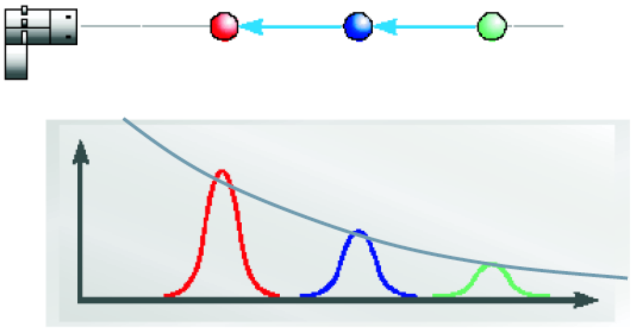


Figure 40.- Signal Threshold

Furthermore, the signal waves are not only distance dependant, but also depend on the shape of the obstacle. In the following figure, different examples are represented using some obstacle shapes.

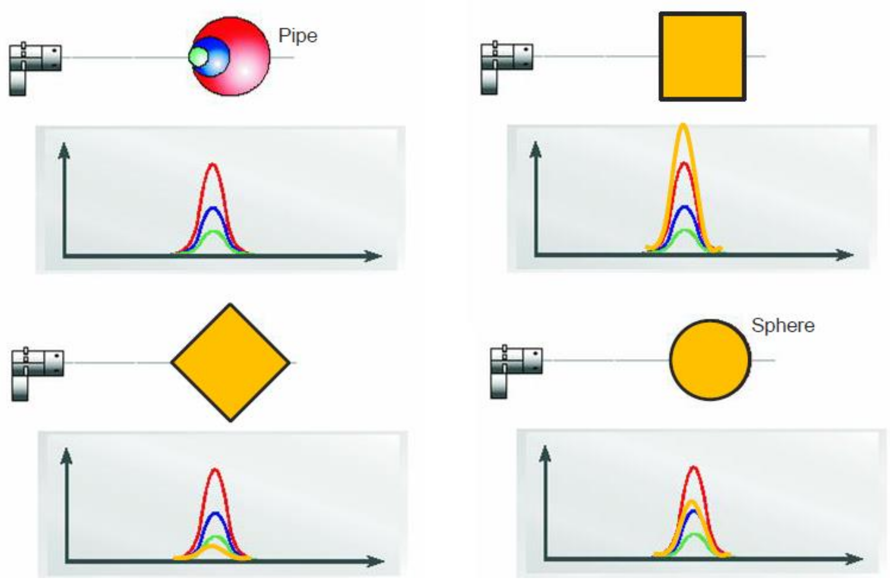


Figure 41.- Obstacle shape

It can be deduced from the examples showed above that the bigger the obstacle is, the more intensity the signal way will have. Moreover, it can be seen that the power of the signal also depends on the angle, having the maximum power when the object is located in front of the sensor, what means with angle cero.

To sum up, it is important to remark the physical limitations of ultrasonic sensors:

1. The range until an object can be detected depends on the shape and the size of the object. Smaller objects will be detected only in short range, bigger objects until bigger ranges. Objects with an angular shape may not be detected at all.
2. The range until an object can be detected depends on the sensor orientation relative to this object. Typically, if the object is a wall, the sensor will more easily detect it if its axis is orientated 90º to the wall, but it will most probably not detect the wall at all if the sensor axis is parallel to it.
3. It follows from 1 and 2 that if the sensor does not detect any object, it does not imply that there is no object (the detection rate of an ultrasonic sensor is smaller than 1).
4. Ultrasonic sensors cannot see through an object (no “X-Ray” capability). Therefore, if an object is detected, it can be safely assumed that there is no high object between the sensor and the object. There could only be a low object as the sensor could look over it.
5. Ultrasonic sensors have no angular resolution. Therefore, it is not possible to estimate the relative speed of objects which are moving in a direction perpendicular to the sensor axis. This is typically the case for objects moving on the side of the vehicle, but this can also happen for objects crossing the vehicle path at the front or the rear.

3.2.3. Feature extraction

3.2.3.1. Global Overview

The ultrasonic feature extraction subsystem is the one in charge of providing freespace information and features to the map having as input the sensor measurement data and the odometry information. Moreover, the outputs of the module are the processed signal ways, features and freespaces and BSD (Blind Spot Detection) warnings.

The next figure represents the subsystem diagram where it can be found the different sub-modules of the system and its inputs and outputs.

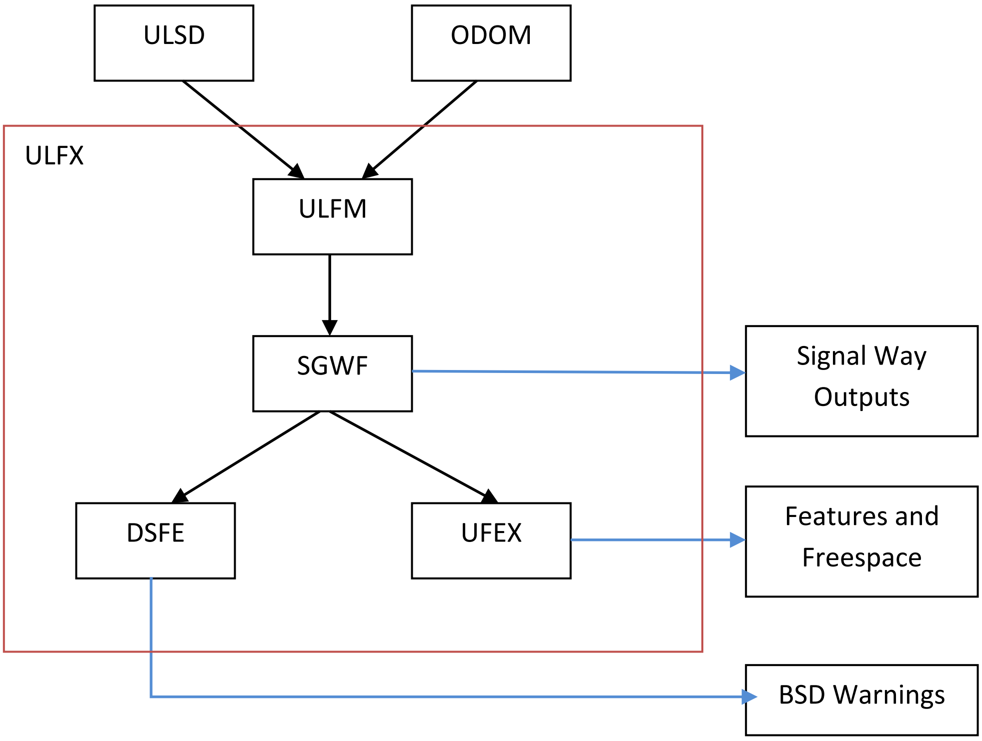


Figure 42.- Diagram of ULFX module

As it can be seen, the subsystem is composed by a master module (ULFM) and three separated sub-modules: SGWF, UFEX and DSFE.

The ultrasonic feature extraction module has three main functions:

- *Calculate distances:*

  The ULSD module provides raw data from the sensors to the ULFX module and it has the function of converting the flight time of the ultrasonic echo into distances.
- *Extract position information:*

  This module is in charge of determining the position of an obstacle taking into account multiple measurements. In order to do it, it uses the odometry information to compensate the vehicle movement and makes some hypothesis to define if the environment can be assumed as static or pseudo static.
- *Detect dynamic objects:*

  Dynamic obstacles are objects of interest in the blind spot detection functionality. Using a detection logic, it can be determined if a dynamic object is approaching to either side of the host car.

3.2.3.2. SGWF

In order to describe how the signal way filter sub-module works, it is important to define that a sensor measurement is called signal way and it can be classified as:

- Direct measurement, when the measuring sensor is in transceiver mode.
- Indirect Measurement, when the measuring sensor is in the receiver mode and there has an adjacent sensor, either left or right, in transceiver mode at the same instant.
- Noise Measurement or noise way, when the measuring sensor is in the receiver mode but with no adjacent sensor in transceiver mode.

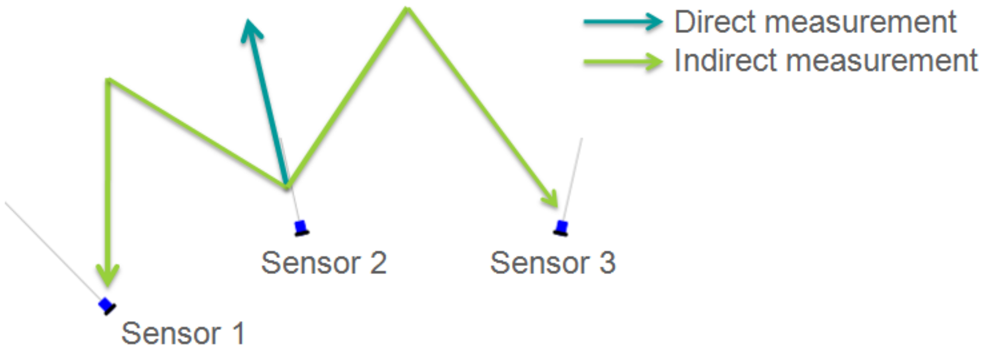


Figure 43.- Signal ways

The system uses multiple signal ways in order to realize its different functions. The signal way configuration for a 12 sensor system (6 front and 6 rear) is represented in the following figure.

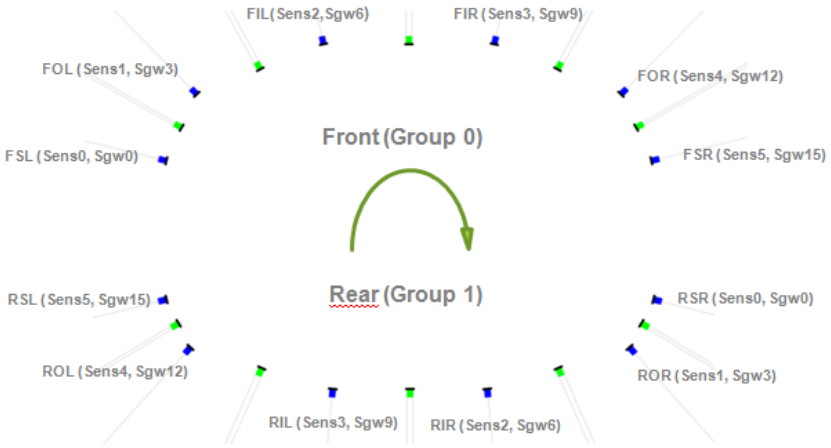


Figure 44.- Signal way configuration

As it can be seen, the signal way system is formed from unique transmitter-receiver pairs what means that each transmitting sensor (blue) has two sensors in receiving mode at each side (green). Each transmitting sensor produces one direct signal way while each sensor pair produces two indirect signal ways. This leads to a total of 16 signal ways for each bumper what means 32 sensor measurements for a 12 sensor system.

Furthermore, the signal way filter sub-module is the one in charge of the following functionalities:

- Calculate the signal way distance

  SGWF is supplied by the information of the ultrasonic sensors and converts the ultrasonic flight times (2us resolution) into distances (1mm resolution).

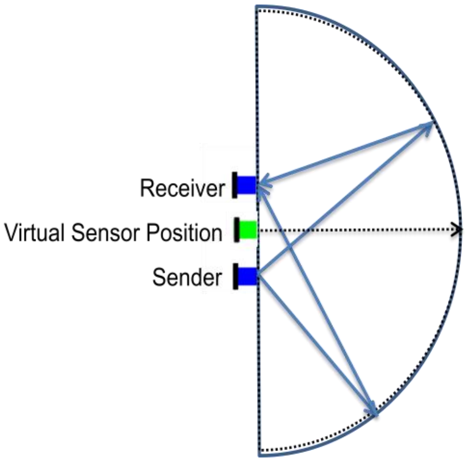


Figure 45.- Convertion of flight times to distances

  From the sensors it can be calculated a distance, however the object can be anywhere on the circumference shown above.

  The circumference is approximated with the virtual sensor as centre (with parameters that are derived from the average of the tx and rx sensors) and radius calculated using the distance between the sensors and the calculated flight time distance.

- Calculate the blind zone distance

  Ultrasonic sensors have their limitation being one of them the inability of measure in very short distances. That is why this sub-module is in charge of calculate distance within the blind zone distance taking into account previous information and the motion of the vehicle.

- Calculate the signal way confidence by association

  Another function of the signal way filter module is to associate echoes between each measurement if they are detected to come from the same object. The purpose of this association algorithm is to calculate a confidence for each signal way.

The confidence level of a signal way determines the certainty of the attributes of this signal way and it is used in the upper layer (UFEX) to generate features, or to avoid generating features with a low confidence.

In order to do it, the association algorithm is based on predictions on the distance of next echo. The prediction is based on two hypotheses, first one is that the object is static and the second one, the object is relatively static.

The confidence of a signal way will increase as it is being associated with previous measurement cycles.

- Determine the height of a signal way

  With the information provided by the sensors, this sub-module is able to determine if the object detected if high or low. The height determination of an object is based on the presence or absence of a double/multi-path reflection after the first echo in the same signal way.

  When another echo appears shortly after the first one it is assumed that the object detected is a high object. This is called, multi-path reflection and can be represented in the next figure:

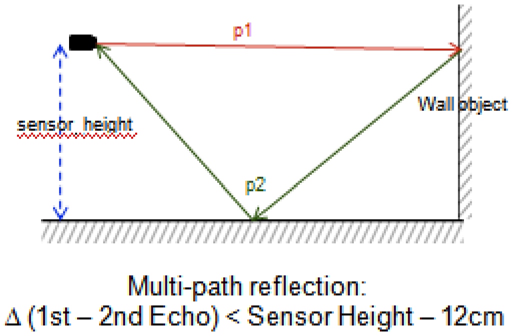


Figure 46.- Multi-path reflection

Moreover, high objects also produce double reflections. This means that there is a second echo in a signal way, which distance is double of the first one.

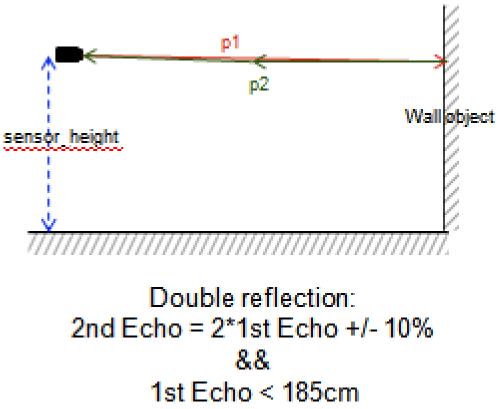


Figure 47.- Double reflection

- Calculate the trend and relative velocity of a signal way

  Furthermore, the signal way filter module can determine the trend of a signal way. This means that it can deduce if the detected object is approaching or departing.

- Detect noise and ice-mud

  SGWF is also in charge of checking for external disturbances and classify them as noise. Moreover, it can also detect if the sensors are covered by ice or mud.

3.2.3.3. UFEX

UFEX sub-module uses the information from SGWF and the vehicle motion and is in charge of extracting feature information and calculating freespace polygons.

In order to create features, UFEX uses two different methods:

- UFEX_SSP: Single Sensor Feature Extraction

  UFEX_SSP builds lines and points features by processing only one signal way (the side sensors). In order to do it, it has to store consecutive signal ways inside a buffer and build hypothesis between the new signal way and the previous signal ways stored in the buffer. The following figures show how point and line features are created:

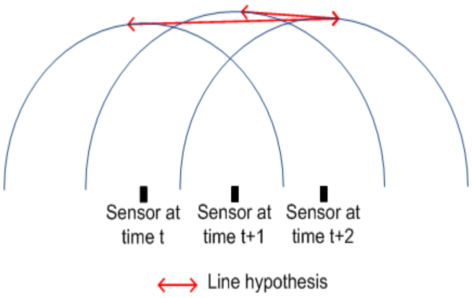


Figure 48.- SSP Line hypothesis

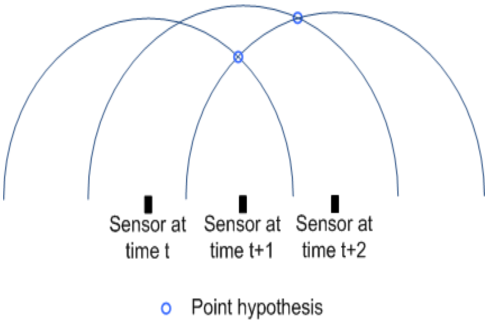


Figure 49.- SSP Point hypothesis

- UFEX_MSP: Multi Sensor Feature Extraction

  UFEX_MSP is in charge of generate features by building hypothesis with the three signal ways from a firing sensor what means two indirect signal ways and one direct signal way. The fact that this method uses three different measurements makes it able to store two hypotheses between the direct and the indirect signal ways per firing sensor.
  In the next two figures it is represented examples of the line and point feature hypothesis.

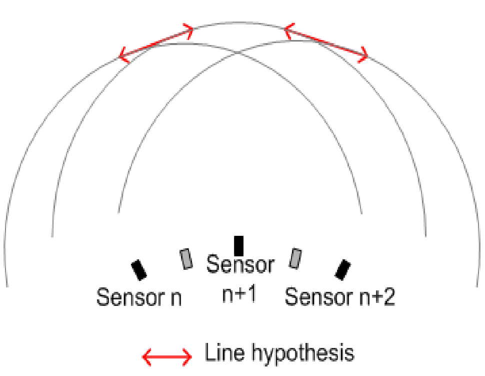


Figure 50.- MSP Line hypotheses

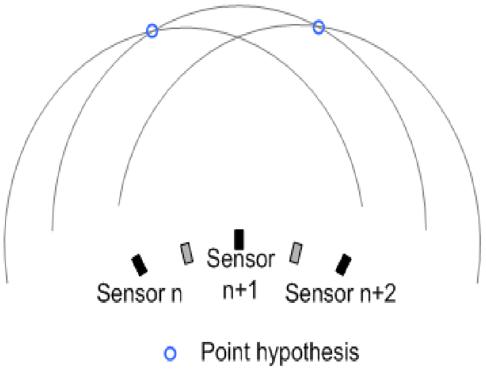
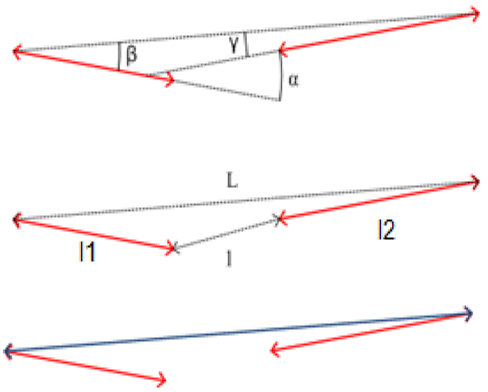


Figure 51.- MSP Point hypotheses

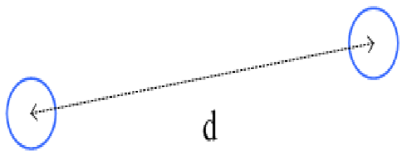
In order to generate line features (blue line) having as input two line hypotheses (red lines), the following conditions have to be satisfied:



- α,β,γ < 10°
- l < 50cm
- L > 30cm (or) L < 30 cm & L > (l1+ l2 – 5cm)

Figure 52.- Line feature conditions

In addition, UFEX_MSP generates a point feature if the following conditions are satisfied and will output the point in the middle of both hypotheses.



- d<10
- The sensors which generated the hypotheses are at least 10cm far apart (meaning, if same physical sensor, it must have moved in between)
- Both hypotheses should have come from different transmitting sensors.

Figure 53.- Point feature conditions

Moreover, UFEX is also responsible of extracting freespaces. In order to do it the first distance of signal way is provided as freespace information and additionally the blind distance of the sensor.

3.2.3.4. DSFD

The main functionality of this sub module is to perform warnings.

3.2.4. MAP

As it was already mention in previous sections, the inputs of the MAP are provided by UFEX module and are freespace information and features.

With the information supplied, the MAP is in charge of the following functionalities:

- Merging:

  In order to reduce the number of features and increase accuracy, the map uses merge areas to check if a new feature can be merged with an old one. If a new a new feature is within the merge area of an old one, it is merged with the old feature.

  The merge are of a new feature is directly constructed from the spatial uncertainties delivered by UFEX_SSP and UFEX_MSP locating the new feature in the middle.

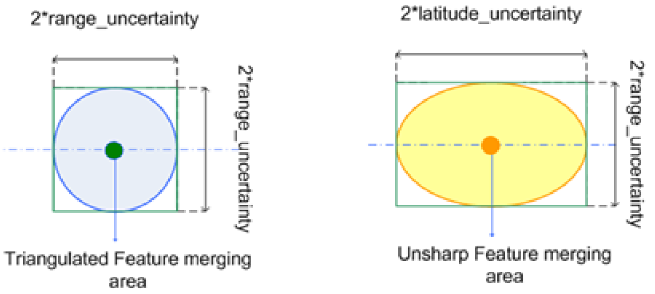
  

  Figure 54.- Merge area

  For runtime reasons, it cannot be checked if a new feature is within the search area of all map features. Therefore, a search for merge candidates is first performed.

- Freespace:

  A freespace model is used by the application in order to delete features taking into consideration the limitation of the ultrasonic sensors that were previously explained.

  To do it, a freespace polygon is defined for each firing sensor. Depending on the polygon, a classification of freespaces is made:

  - Feature freespace

    Taking into account the properties of the sensors, the range and orientation under which a feature was detected are stored. Therefore, if a feature is not seen under the same orientation (+/- 13°) and at a closer distance (with 20 cm offset) it was seen before or under 60 cm, the feature is located in feature freespace and can be eventually deleted after some debouncing.

- <u>Active freespace</u>

  Active freespace is defined as the area between the sensor and an active measurement. In order to not generate wrong active freespaces, the active measurement shall have enough confidence, using features instead of signal ways, and a good spatial accuracy to know in which direction the active freespace shall be generated.

- <u>Sensor freespace</u>

- <u>Creation of clusters:</u>

  When a new feature is created, the sub module PCLU searches for potentially matching clusters located around the new feature using different search areas for features located inside and outside the vehicle strip.

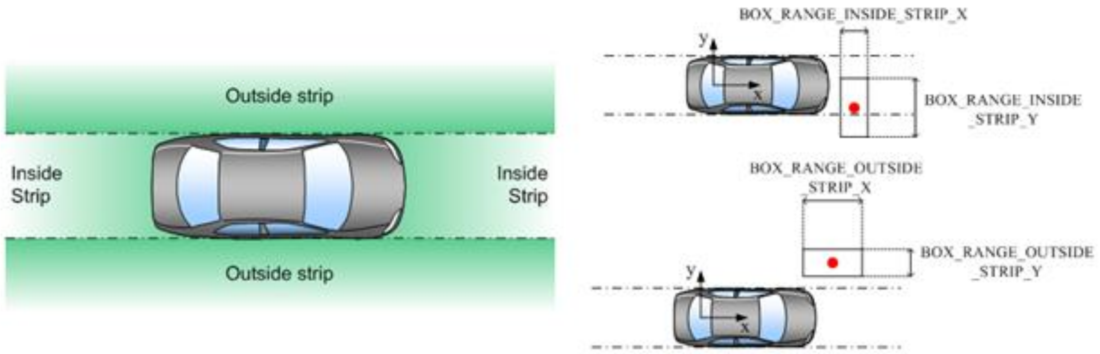


Figure 55.- Cluster creation

The decision to create a new cluster, assign a new feature to an existing cluster or merge clusters is made according to a T-shape, barycenter and adjacent filter algorithms.

3.2.5. Interpretation

Within this layer there is the slot detection sub module (SDET) which main functionality is:

- Slot detection and construction

  SDET is in charge of determining parking slots by finding enough slot depth and length between two high objects. In order to do it, it is supplied by the clusters already created by PCLU.

  In addition, slot detection has also to define which kind of slot is detected, differentiating between parallel or perpendicular slots.

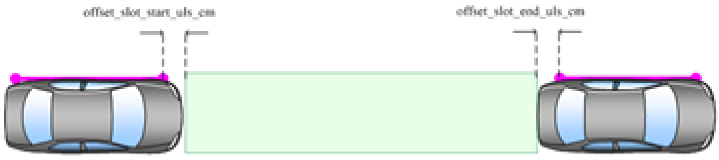


Figure 56.- Slot detection

4. Antecedents: State of the art

Passenger car safety is an issue that has received increasing attention over the past few decades. Over the last decades, automobile crashworthiness has improved drastically. Research studies show that the cause of most of car accidents is either the driver distraction or his failure to react in time. Therefore electronic brake control system leads to significantly greater driving safety. The risk for a fatal injury in a new car has decreased by 90% compared to a car from the early 80’s.

4.1. Brake system

The purpose of the automated car braking system is to shorten the stopping distance so that an accident can be completely avoided, or in case of an accident occurring, to reduce the severity of the accident by reducing vehicle speed without any driver input. The system will successfully detect an obstacle (vehicle, person or object) at a specific range and create a way for the system to avoid collision by braking the car.

Valeo´s emergency braking system (IEBM) uses its ultrasonic sensors in order to observe the environment directly and detect vehicles or obstacles in front and back of the host vehicle. Based on information from the sensors, the decision of braking is made to avoid imminent collisions.

4.1.1. Functions

IEBM has two main functionalities:

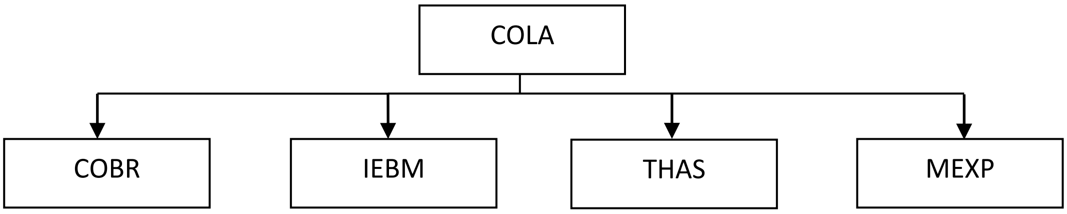
- Collision avoidance:
  As the term “avoidance” implies, this function completely avoids collisions with objects by braking early enough and achieving a standstill state before the accident occurs.

- Collision mitigation:
  In this function, braking is activated to reduce the power of the impact even if it is known that the car will not come to standstill before colliding the other object.

Moreover, it is important to add that Valeo´s IEBM subsystem includes a Prefill request which gives the system the information that a brake is likely to happen soon. By this, it can align the brake shoe to the brake disk and start building up some pressure in the brake system.

4.1.2. System Architecture

In previous DAS generation the braking system function was all in one module. Nowadays in the latest DAS version (DAS 3.2) the braking system is spitted into several modules as it can be described in the next figure.



- COLA : Collision Avoidance (Master Module)

  - THAS: Thread assessment

    The module THAS calculates the wheel angle ranges where a collision is possible and assigns a criticality level to each range. This data can be used to take decision to decrease vehicle speed and/or brake, or steer to avoid collision.

  - IEBM: Emergency brake

    The module IEBM calculates data that can be used to take decision to decrease vehicle speed and/or brake, by estimating distance and time to objects and avoidance criteria.

  - COBR: Comfort braking

    The module COBR reports the DTC (distance to collision) in driving direction after checking if the obstacle is not avoidable. It also reports the DTC in the direction opposite to driving direction.

  - MEXP: Exploration mode

    The module MEXP checks if there are obstacles in the straight path of the vehicle. If yes it computes the required wheel angle to avoid them. It also checks if there is a wall within a configurable distance and angle to the vehicle and aligns the vehicle to it.

4.1.3. Braking conditions and results

In order to trigger the braking system, the following conditions are checked:

- Speed range and driving direction

  Different parameters can be configured according to each driving direction.

- Object detected with a certain minimum confidence

  For avoiding failures on the braking system the application must have a minimum certainty level that the object exists. This level can be differently configured for front, rear and side collisions.

- Object matches activity criteria

  Depending on the project it can be selected to brake in every object or for example only in measured objects.

- Side collisions can be covered

  It can be configured to switch on or off the side collisions.

- Objects in the driving tube with a certain percentage

  Normally, the system only brakes on objects within the driving tube or at least with a high percentage. Objects detected out of it are not considered for the braking function.

- Object height

  The brake function can be set to be triggered depending on the height of the obstacle:

  - High only
  - High and unknown object height
  - All heights

- Object validated with more than a certain number of detection algorithm

  It can be configured that in order to make the system brake, the object has to be detected by not only ultrasonic sensors, but also by camera.

- Optionally: object tracked

  It can be chosen to brake only on objects that have been tracked.

- Optionally: point within “hard to avoid area”:

  If an object is within this area means that the driver could not avoid hitting it with an easy steer. Therefore, if the object is located outside this area the brake system won´t be triggered because it is assumed that the driver can avoid it easily.

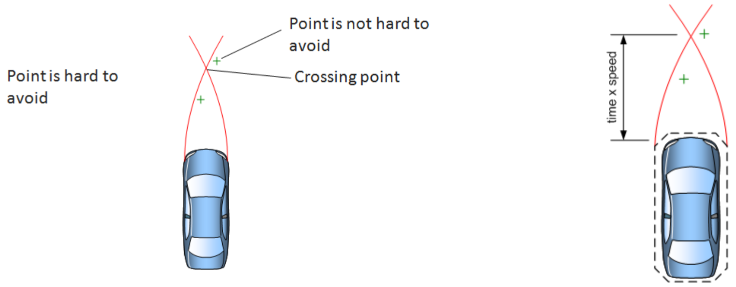


Figure 57.- Hard to avoid area

- TTC (Time To Collision is below a certain threshold)
- DTC is below speed relevant distance:

  For each car a Braking Table is made in order to determine the thresholds of the distance to collision at which the host car should brake depending on its speed.

For speeds above 10 Km/h, as it can be seen in the example below, the DTC threshold remains constant. The reason is because if the threshold keeps increasing with the speed, when the car drives with high speed at the first moment when the system detects an object it will trigger the brake and this will make increase the rate of false positives dramatically.

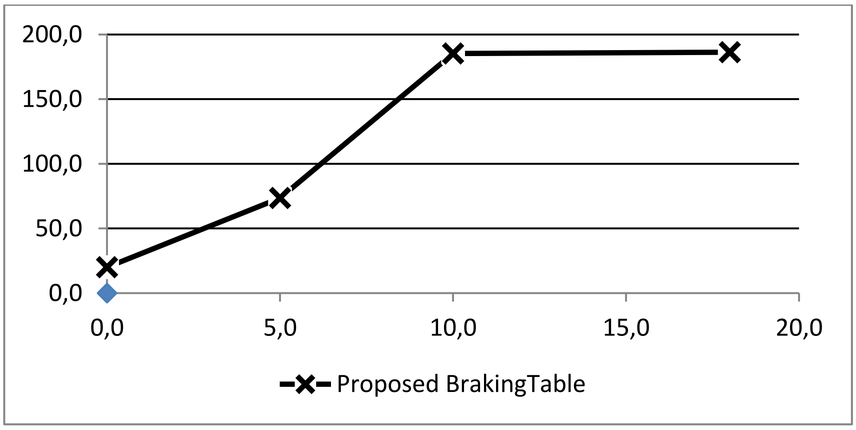


Figure 58.- Proposed braking table

In order to generate the braking table of a specific car, the following steps are followed:

1. The car needs a functional braking system with a first threshold set.
2. Enter a first (preliminary) braking table into de ECU.
3. Drive with different speeds towards a large and high object. Then, mark the position of the object on the ground.
4. Car should brake
5. Once the car has brake, measure the distance to object. If the car has hit the object, the measured distance should be negative.
6. Fill out a test protocol.
7. From the information of the traces, the following data should be available: Speed where braking starts, DTC from brake start and distance to object after standstill.
8. Make a diagram Braking Distance against Speed. The "Braking Distance" should be a curve that gets higher over speed. Get the parameters of the fitting curve.
9. Enter the parameters of the fitting curve in the braking table sheet in addition with the maximum speed which the customer expects collision avoidance and mitigation and the desired distance to object after standstill.
10. Check if the proposed braking table diagram fulfills your expectations.
11. If not, take values go again to step 2.

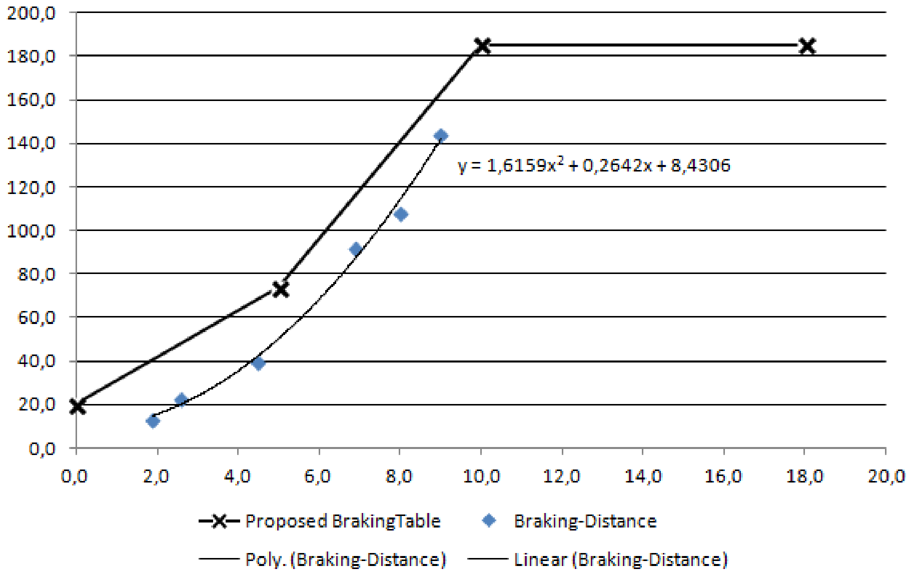


Figure 59.- Comparison of proposed table and braking curb

On the other hand, it is important to analyze if the results of the braking system are fitting the expectations.

The next table shows the different results that the brake system can provide depending on what was expected from the braking point of view and what was its real reaction.

Table 3.- Posible braking results

		Expectation	
		Brake	NO Brake
Reaction	Brake	**True positive (TP)** Exp: Brake Reac: Brake	**False positive (FP)** Exp: NO Brake Reac: Brake
	NO Brake	**False Negative (FN)** Exp: Brake Reac: NO Brake	**True Negative (TN)** Exp: NO Brake Reac: NO Brake

As it can be seen the different results that can be obtained are:

- *True positive:* the scenario requires an emergency brake intervention and the system react with an emergency brake action.
- *False positive:* the scenario does NOT require an emergency brake intervention and the system react with an emergency brake action.
- *False negative:* the scenario requires an emergency brake intervention and the system does NOT react with a emergency brake action.
- *True negative:* the scenario does NOT require an emergency brake intervention and the system does NOT react with a emergency brake action.

4.1.4. Applications

The existing applications of Valeo´s braking system are:

- Braking while automatic parking

  Nowadays, an automatic parking application provided by the company offers the possibility to include the emergency braking function.

  The brake system gets triggered if during the maneuver the host car gets too close to another car or object or if any unexpected objects appear in the driving tube. Usually the objects that appear during automatic parking are static, that is why the environment is "controlled" and the braking function has a low probability of failure.

  The fact of including this functionality in automatic parking increases the damage protection of the host car and the added value of the product.

- Pedal misapplication support after direction change

  This function consists on limiting the motor power or car speed after the driver has changed the gear if there is an object detected in the intended driving direction.

  The added value that represents this function is that the impact of accidents due to wrong pedal usage is limited. Additionally, it helps at each direction change.

  The risk of failure of this application is low, however if it fails, it causes a remarkable impact in the customer.

- Pedal misapplication support plus brake after direction change

  This function gives support to the driver not only limiting the motor power but also by braking when he changes the gear and there is an object in the driving tube. This application helps in each direction change and its risk of failure is low.

- Backwards braking while UPA is active

  This function triggers an emergency brake if the car is driving backwards and UPA is active.

  The positive result of this function is that it improves the damage protection of the host car while it is driving reverse.

  Generally, backwards driving is done in controlled environments and during short distances that is why the risk of collision at false braking is low.

- Braking while UPA is active

  The objective of this function is to have the braking system active together with UPA and activate UPA only with the reverse gear.

  This application leads on a remarkable benefit in maneuvering situations.

- Pedal misapplication support during standstill

  This function pretends to limit the motor power or vehicle speed if there is an object detected in the intended driving direction. It is suppose to provide a benefit in terms of limiting the impact of an accident due to usage of a wrong pedal but it is also important to consider that there are risky use cases as it is a Red-Light situation where there are likely false limits.

  Therefore, the risk of failure of emergency braking in this application is determined as high.

  In order to achieve this target, the company requires developments such as a dynamic object classification which is the main goal of this thesis.

- Braking during free driving

  The goal of this application is to achieve a braking function while free driving if the host car is too close to other objects detected in the driving direction.

  This function is not currently implemented in the application because of the high risk of failure of the braking system.

In order to summarize all the possible applications, the next figure is made taking into account the effort and risk of each different function.

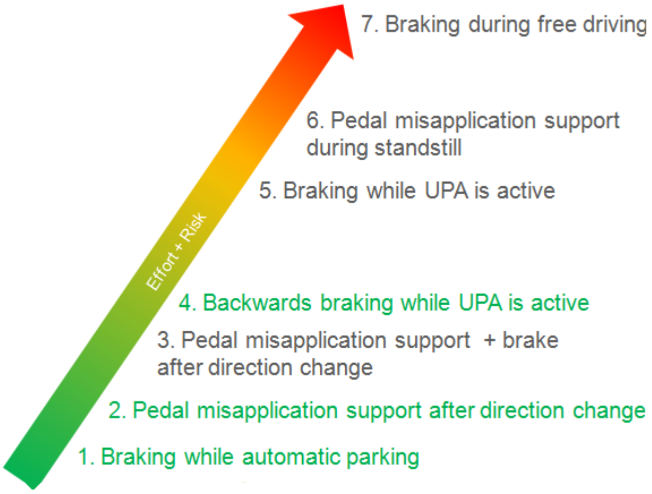


Figure 60.- Braking levels

4.1.5. System limitations

Currently, as the brake function is based on the data provided by the ultrasonic sensors, which can supply limited information, there are some system limitations such as the ones showed below.

In this section of the report the actual system limits are described.

- *Situations that the system can´t detect and might trigger a brake even if it is not desired*

  Examples:

  - Driving through "grass" :

    As it is known, ultrasonic sensors are not able to classify which kind of object they are detecting. Therefore, there are situations as driving against grass where the ultrasonic sensors detect a high object in front of them and trigger the brake when it is not desired.

Figure 61.- Brake system limitation 1

  - Driving into water

    This situation is similar to the previous state where the limitations of ultrasonic sensors are remarkable.

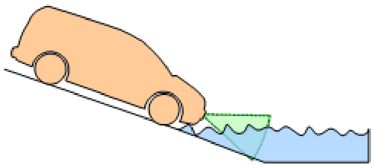


Figure 62.- Brake system limitation 2

  - Sensor de-adjustment

    The de adjustment of the sensor position in a car, due to for example an impact, limits the system because some of parameters would be wrong and maybe, as it is shown in the figure, the only echoes received by the sensors come from reflections of the ground.

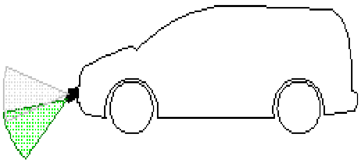


Figure 63.- Brake system limitation 3

- *Objects that are exactly on the border of the driving tube*

  As it is explained in the braking conditions section, the braking system is usually triggered when the object is completely or with a high percentage in the driving tube but when the object is located exactly on the border of the driving tube, it is hard to distinguish if they are in the driving tube and if it is needed to trigger the braking system

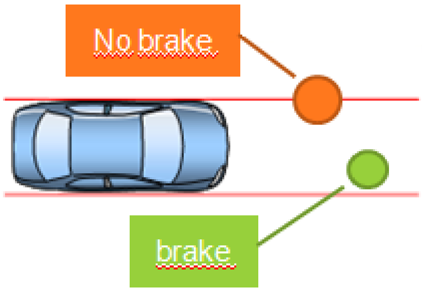


Figure 64.- Brake system limitation 4

- *Pedestrians*

  The detection of pedestrians with only ultrasonic sensors is really difficult to achieve due to:

  - Bad reflection properties (soft clothing)
  - Undefined contour / reflection points
  - Slow (as compared to noise/update time)
  - Hard to predict
  - Large (as compared to tracking area)

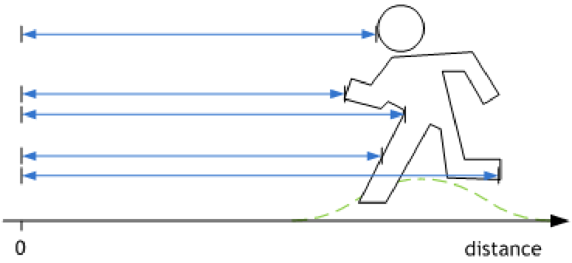


Figure 65.- Brake system limitation 5

- *Dynamic objects*

  The actual ultrasonic object detection does not fully support moving objects which makes the brake function difficult to succeed while free driving.

5. Case presentation

The major problem when designing a system with intention to avoid or mitigate collisions, is deciding when the system should intervene. This decision making problem is complicated, not only because of the measurement and process uncertainty, but also because of how the driver will perceive an intervention.

Nowadays in heavy traffic, while driving too close behind other cars, the system wrongly brakes (False positives). The actual limitation of the system is that, as it is mentioned in the previous section, it only detects static objects and the whole application in based on it. For this situation a classification of the leading object as “dynamic” is needed.

Moreover, the same problem appears when the host car is stopped at a traffic red light behind another car. At the time that both cars start moving again, the system detects an object close to the host car and triggers the braking system.

The risk of false positives in the braking system is very high because it can produce traffic accidents. If the car brakes when it should not during free driving, it can cause a crash with the car that is behind the host car.

In order to show clear examples of the situations, two real traces, one obtained from an two different clients will be presented.

These traces were analyzed with a simulation tool called DASte, which enables to reproduce a trace step by step. This tool provides information from all the principal modules of the application including the information from the braking system.

5.1. Situation 1: Red-Light

In the figure below it can be seen a real trace from a second client car in a traffic red light situation. It is represented the different echoes distances (raw data) that the front sensors receive. In order to analyze this situation it can be noticed that there are 3 different situations in the trace.

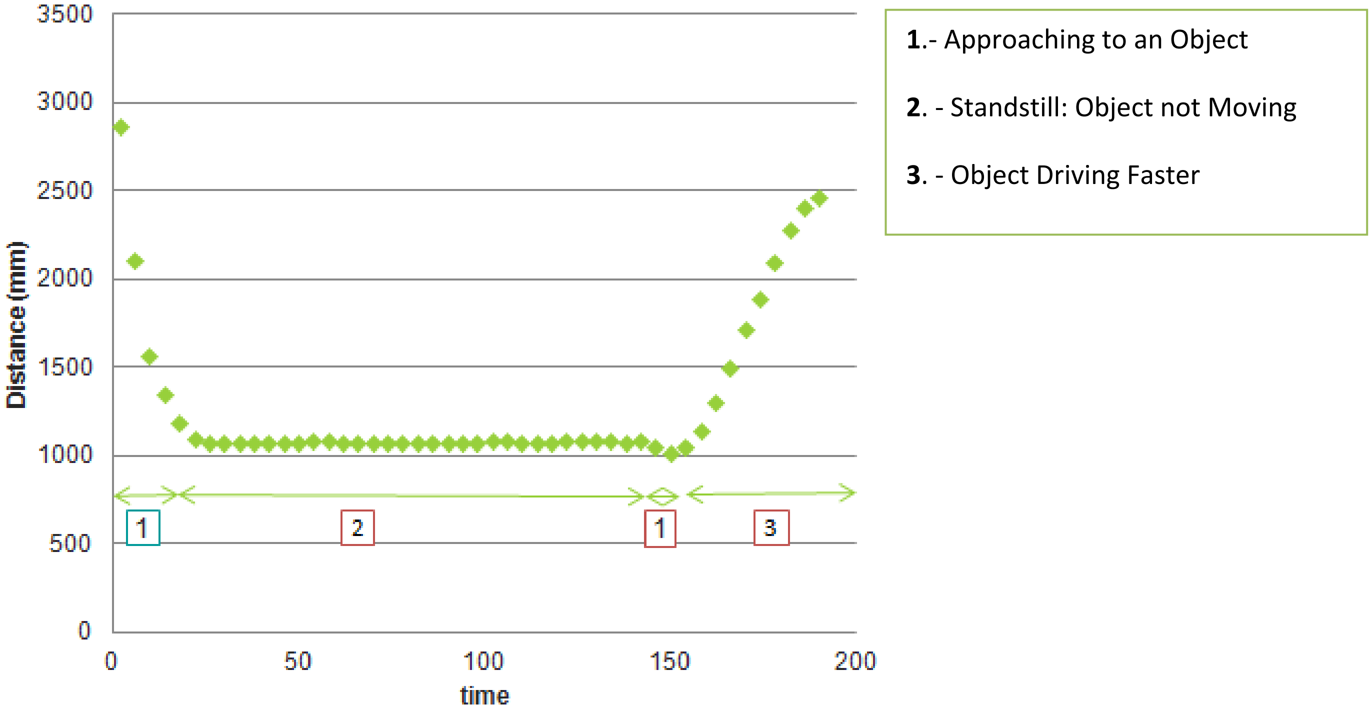


Figure 66.- Redlight trace

The three clearly differentiated states are:

1. The first situation that can be appreciated in this trace matches with the state: **Approaching to an object** [1], which in this case was a car standing in a traffic red light. This affirmation is based on how the echoes distances decrease during this period.
   In a more exhaustive analysis, it can be demonstrated if our car is approaching to a static object or if it is approaching to a dynamic object which speed is lower than our own car.

2. As it can be seen in the trace, the distance to the object remains the same during all the period and as our car is in Standstill it can only means that the object in front of it is also standing without moving or is a static object. Therefore, the second situation that appears in this trace [2] corresponds to a **Standstill: Object not moving** state.

3. The last situation that can be distinguished in the figure above [3] can only match with a situation in which the object in front of the host car is **Driving faster.** This supposition is done based on the distances detected by the sensors which are increasing during this period of time. We distinguish between object driving faster and object departing depending on the speed of the host car. If the host car is in Standstill and this situation appears, the situation would have been Standstill: Object Departing.

Running the trace with DASte it could be noticed that the car brakes during the third situation while the car in front was driving faster than the host car, exactly when the distance to collision (DTC) is 1436 mm. At this moment a line feature is created and as the speed is relatively high (8.8 Km/h) all the conditions for triggering the brake are fulfilled and therefore, the car brakes.

As it can be clearly seen in the figure of the whole trace, during this situation the echo distances keep increasing and there is not risk of crashing, then the brake is an example of a false positive because the car should not brake.

5.2. Situation 2: Driving behind another car

The trace next presented is from a car of the first client. In this trace, the host car beginning from a standstill state starts following another car in front of it with multiple changes of speed that make the trace more interesting. These changes are the ones that produce the different situations that will be analyzed below.

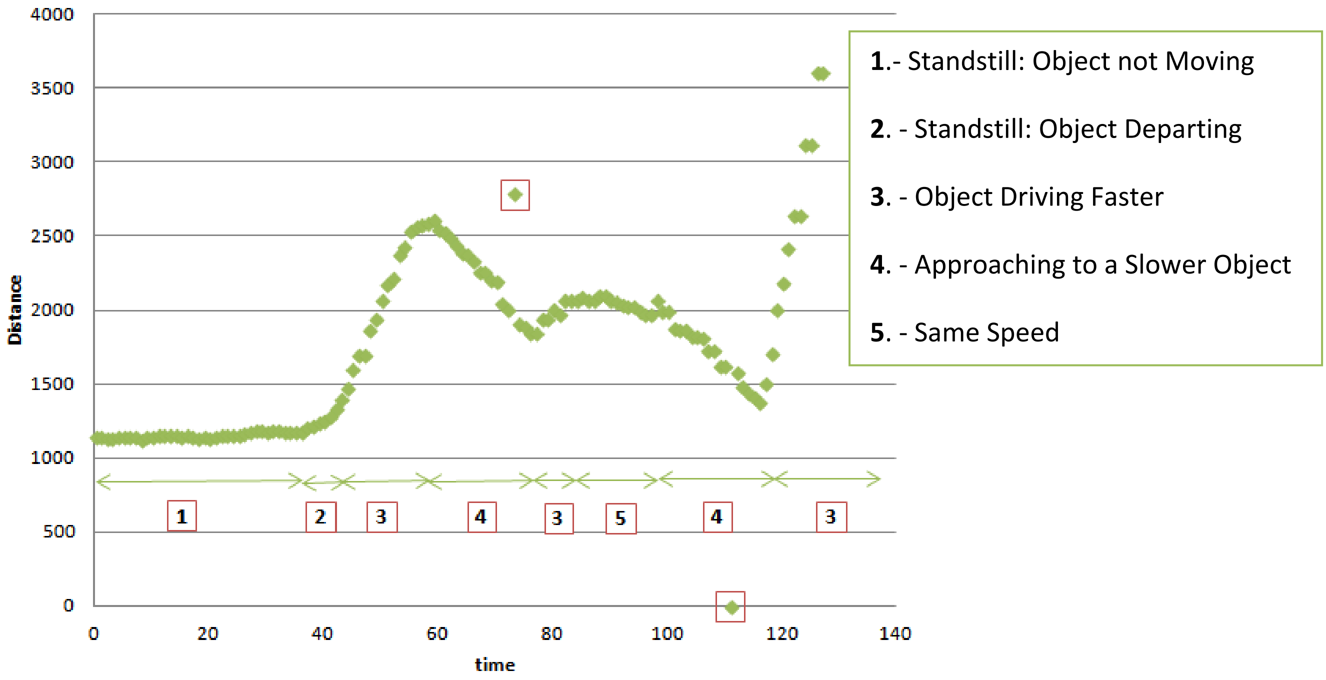


Figure 67.- Driving behind another car trace

As the number of different situation in this trace increase, it´s complexity also does it. In this trace it can be seen five different states:

1. In this first period it can be appreciated that the echo distances remain constant and as the host car is in standstill, it can be assumed that the object in front of it is a static object or is another car also in standstill. Therefore, the state of this situation is **Standstill: Object not moving**.
2. During the second period it can be seen that the echo distances start increasing, however the host car stills in a Standstill state. This situation can only means that the object in front of the car started moving and is departing while the host car is in standstill that is why the state of this period is called: **Standstill: Object Departing.**
3. As it can be seen in the figure, in the third situation the echo distances increase during this period of time while the host car is driving. Therefore, it can only means that the object detected is driving with a higher speed. Then the state is: **Object Driving Faster**.

4. During this period it can be assumed that the car is approaching to another object because the echo distances are decreasing. However, based on the information provided by the system and taking into account that the previous state is Object Driving Faster, it can be distinguish between a situation where the host car is approaching to a static object or is approaching to a dynamic object that is driving with a lower speed. This is the case of this situation and therefore, its state is **Approaching to a Slower Object.**
5. Finally, it can be appreciated another different situation where the echo distances remain more or less constant while the host car is driving. It can easily be assumed that this case corresponds to a situation where the car is driving behind another car with a similar/**Same Speed**.

Moreover, another remarkable issue that can be seen in this trace is a measure [time 77] that does not fit with the trend of the trace. It is important to notice that it is a wrong measure and that the application should be able to also notice and classify it as an invalid measurement.

In addition, it can be noticed that there is a measure [time 111] in zero mm. This case can mean that the object is too far to detect it by the sensor or there is no object or as it happens in this case the sensor miss a measurement. In this situation it is assumed that in front of the sensor there is Freespace.

Doing an analysis with DASte, it can be reproduced a brake during the fourth situation when the host car is driving behind a car that drives with a lower speed. More accurately, the brake occurs at time 59, when the distance to collision is 2616 mm and the speed of the host car is 5.11 Km/h.

In addition, at time 85, the distance to collision is 2096 mm and the speed of the car is 4.9 Km/h the system triggers wrongly again the brake while the car is driving behind another car with a similar speed.

6. Work development

6.1. First steps

To analyze and understand the real problem of the system, it was important to learn how the different tools and application worked. Some real traces were supplied and using DASTE a first analysis was made in order to have different solution proposals.

First of all, a trace where the host car was approaching to a wall was taken. The reason to start with this kind of trace was that the final solution should solve false positives of the brake system and should only trigger the brake when the host car is approaching to a static object. Therefore, it is important to analyze thoroughly this case to see the difference between the real development of the echo distances among the time and a theoretical approach.

In the theoretical approach used, the time and the speed of the host car in each step were taken into consideration and with these data, it was able to make a supposition of the next distance to the object and interpret the results.

In the following figures it is presented the different results obtained in this first analysis. The first figure represents the development of the echo distances of the raw data provided by the ultrasonic sensors, the second represents how the development of the distances should be under the theoretical estimation. To sum up, the third figure shows both results at the same time.

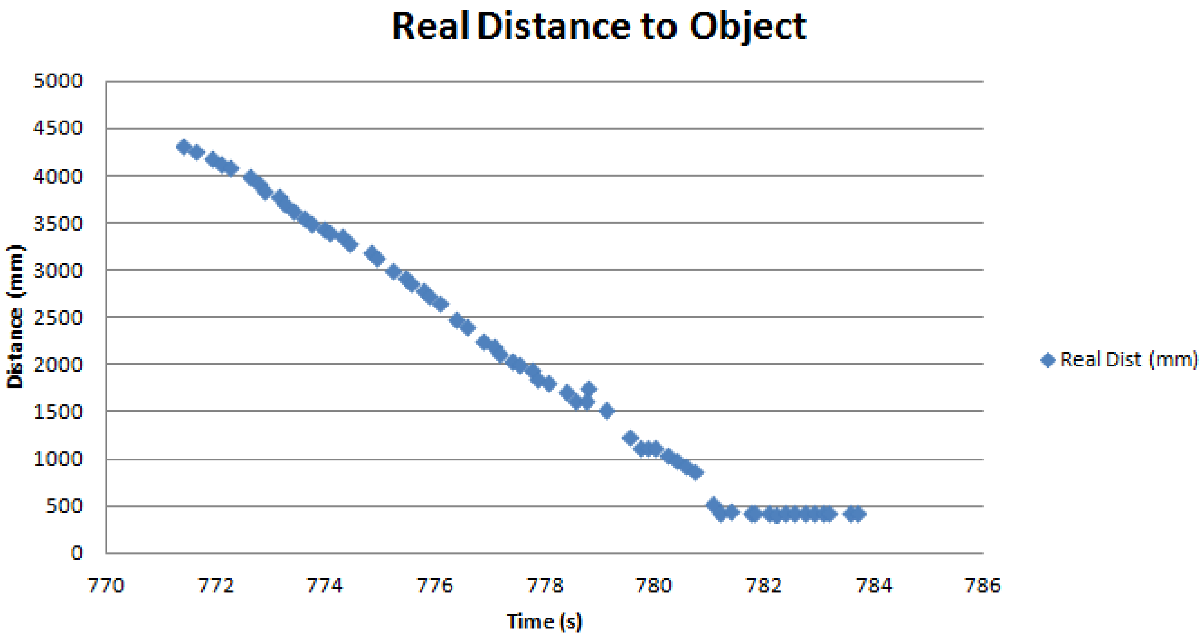


Figure 68.- Real Distance to Object

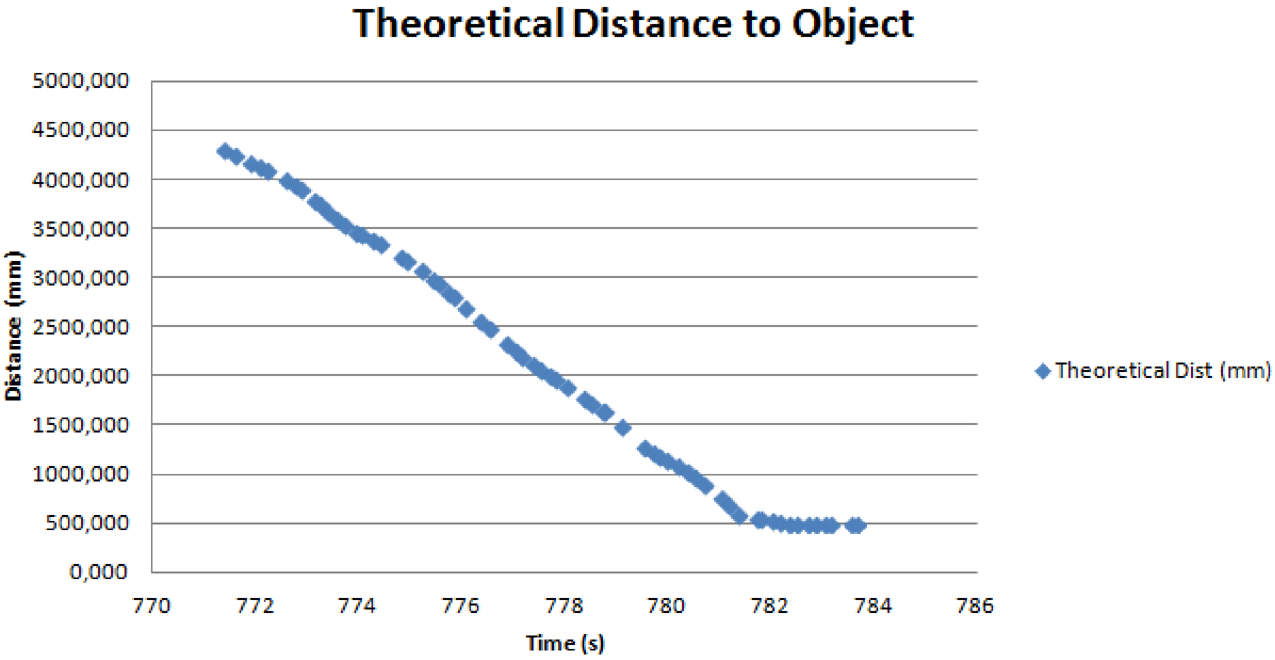


Figure 69.- Theoretical Distance to Object

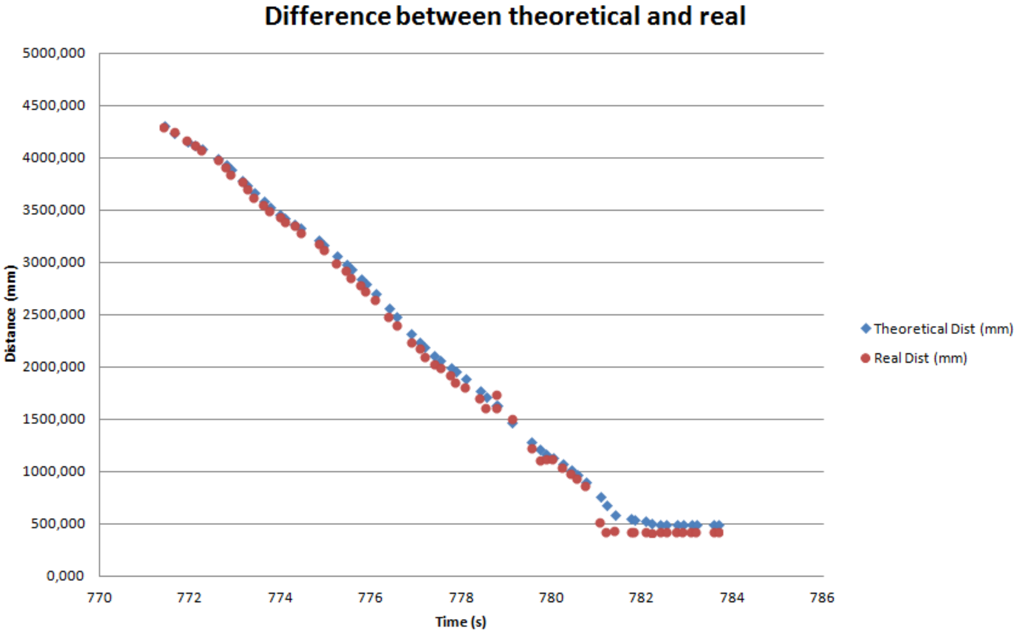


Figure 70.- Results of a first analysis

As it can be seen in the figure above, the results are very similar what means that in order to propose a solution it can be based on a theoretical estimation. The complete analysis is attached in Annex 1.

6.2. Possible solutions

6.2.1. Software layer analysis

In order to develop a code that would be able to make a dynamic object classification it is important to discuss at which layer it should be implemented. The different options are from the highest level (Map) to the lowest (Signal ways):

- Functional layer (IEBM)

  As the braking decision is taken in the IEBM module, one option could have been implementing an algorithm directly in at this layer. However, this option was barely considered as at this layer, a lot of previous static assumptions are already made.

- Map layer

  Developing the code at this level would mean doing it in a higher level where the features are already created by UFEX. The main advantages of working at this level are the functions already implemented in the map as the merge and deletion of features. However as it is already known, the map used by the system is an static map, what means that all the assumptions and functionalities currently working won´t be useful in order to solve the problem of dynamic objects. Therefore, the solution considered was to create a dynamic map.

- UFEX layer

  This means, to use the previous layer and integrate the code taking as input the features created in this level. However, it was important to consider that all the features are created based on static assumptions. That is why if the solution is implemented in this layer, only the hypotheses used to extract features have to be considered. The advantages of using this layer is that line and point hypotheses already have the association algorithm implemented and in the case of lines hypotheses there are some extra information such as the angle and the length of the line which could be useful in order to detect a car driving in front of the host car.

- SGWF

Finally, there is the option of using the signal way layer. This means to use directly the distances provided by the sensors. The main advantage of implementing the code at this level is that there is no assumption done before, this means that the data are not modified by any previous static hypothesis, it is raw data. The most important disadvantage that has to be taken into account in this layer is that it means working with no filtered data, wrong and noisy measurements and missed measurements and therefore the code to create has to be able to detect all these cases and minimize its impact.

6.2.2. Code basis

In order to develop a code that could classify the object detected by the sensors of the host car, there were two possible bases: one based mainly in the distances and the timestamps and another based on the relative speed of the object.

In a first approach, as it was explained in the previous section, a possible solution could be based on the time, the speed of the host car and the first echo distances.

On the other hand, as the system already had a functionality that calculates the relative speed of the detected object, a code based on this variable could also be a feasible option.

6.3. Adopted solution

As it was explained in chapter 6.2.1, there were different layer options. However, as the goal of the code was to take into consideration static and dynamic objects, every level in which some static assumptions were already made, an extra difficulty was added because the input to the code would have been transformed.

The idea of using the map layer by creating a whole dynamic map was the first one to be discarded. First of all, create a dynamic map was not profitable to do it as a task for a master thesis and moreover, it was also difficult to develop it because in the previous levels there are already some static hypotheses taken into consideration.

Furthermore, the next layers were considered. UFEX has not only the advantage that the association algorithm is already implemented and wrong echoes are already deleted, but also the extra information supplied by the features hypotheses. On the other hand, there was SGWF module which input is the raw data with any data transformation or filter. The fact of using one module or the other does not make a very remarkable difference in terms of developing a profitable solution, however, the grade of difficulty highly increases when the code has to work with the UFEX hypotheses. That is the reason why the layer that has been chosen to develop the code in, is signal way filter.

Moreover, it has been previously mentioned that there has been considered two different strategies to make the code, one based on timestamps, distances and speeds and another one based on relative velocity. During the realization of the code both solutions were elaborated, starting by the one based on the relative velocity.

However, after a first approach, it was noticed that this first solution was not very reliable as the relative velocity of an object detected for the first time was initialized to the negative vehicle velocity and in order to have a solid relative speed the system has to precede subsequent associations. This means that if two echoes are associated, the relative velocity is the result of an alpha filter that takes into account the value of the previously stored relative velocity and the speed of the current echo and if the echoes are not associated, the relative speed is not certain.

The alpha filter has the following structure:

$$New_{relative\ velocity} = alpha \cdot old_{speed} + (1 - alpha) \cdot current_{speed}$$

For performance reasons, the alpha is a term of the type (1/ (2^n)) and to calculate the current echo speed it the delta distance and the delta time are used.

Therefore, the whole algorithm was changed to the second possible strategy taking this one as final.

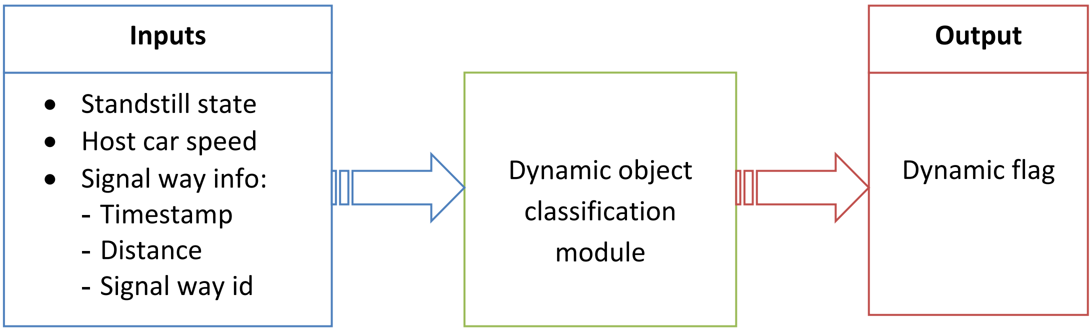
6.4. Code development

6.4.1. Module interfaces

In order to create a new module is important to know how are the different interfaces between the module and the rest of the system. This means it has to be clearly determined what the input of the module is and what would be the output.

The adopted solution is based on timestamps and distances, therefore the module input is in each measurement the speed of the host car, the standstill state and the information of the signal ways. As the goal is to detect dynamic objects driving in front of the host car, the signal ways taken into account are only the four located in the front bumper. In every measurement, it is checked if each of these sensors has received an echo and then the next information is supplied: timestamp, distance and id of the signal way which allows distinguishing from which sensor comes the information.

On the other hand the output of the module is to determine the movement state of the object detected in front of the car for each of the four signal ways analyzed. Then, once the four states are set, two conditions are considered in order to determine if the "*Dynamic flag*" should be set to TRUE. When the system tries to brake, this flag is checked and only in the case where this flag is equal to TRUE the car will brake.



The two conditions taken into account to set the "*Dynamic Flag*" to TRUE are:

- Two or more states are "*Approaching to static*" or "*Against*"
- At least one of the states is "*Approaching to static*" and the rest are "*Not enough info*"

The different ten states that have been defined depending if the car is in standstill or not are represented in the table below:

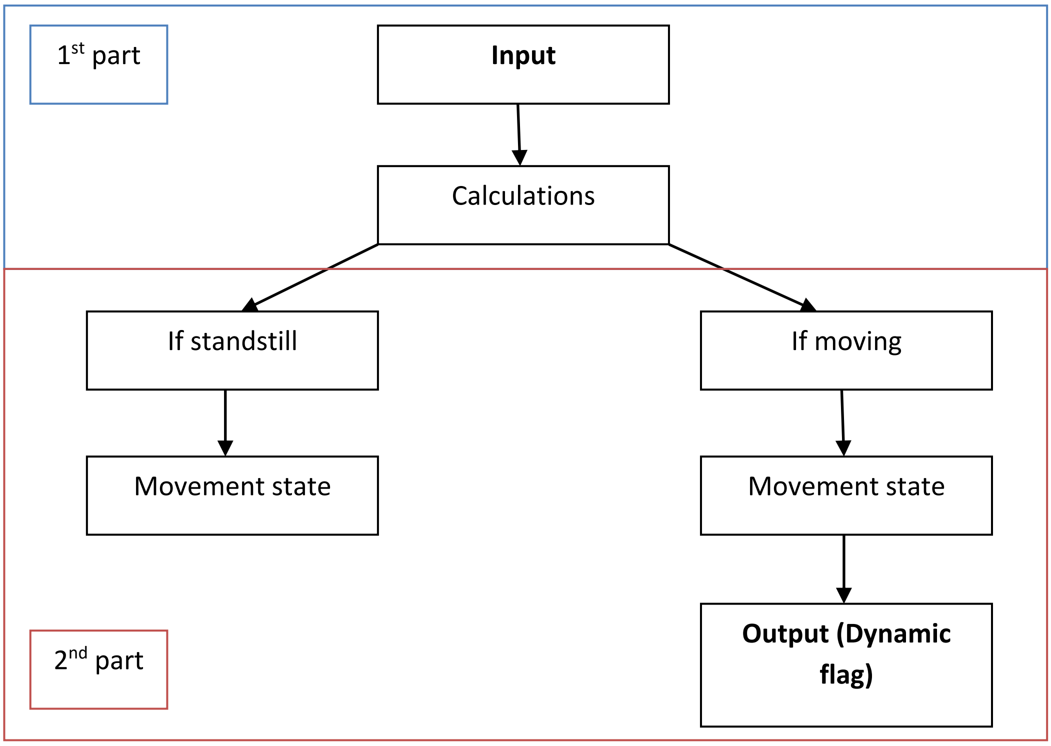
Table 4.- Module states

General states	While ego car is in standstill	While ego car is moving
Not enough info	Object not moving	Approaching
No object	Object departing	Approaching to a static object
		Approaching to a slower object
		Object moving faster
		Same Speed
		Against

- *Not enough information:* the sensors have detected for first time an object and the module does not have enough information to determine a reliable state.
- *No object:* The sensors do not receive any echo back, this means that there is freespace in front of the car.
- *Object not moving:* an object is detected, the echo distances remain constant and the state of the host car is Standstill.
- *Object departing:* the echo distances of the object detected are increasing from one timestamp to another while the host car is not moving.
- *Approaching:* the distance of the detected object has decrease but the module does not have enough information to determine if the car is approaching to a static object or to a dynamic object that is driving with a lower speed than the host vehicle.
- *Approaching to a static object:* the echo distances are decreasing from one timestamp to the next one fulfilling the conditions of static objects.
- *Approaching to a slower object:* the echo distances are also decreasing however the static conditions are not fulfilled.
- *Object moving faster:* in this case, the echo distances increase from one timestamp to the next one while the host vehicle is moving as well.
- *Same speed:* the detected object is a dynamic object that is moving with a similar speed as the host car. This is determined by the echo distances that remain constant while the host car is driving with a certain speed.
- *Against:* the case means that two dynamic objects are moving against each other. The pattern is that the echo distances are decreasing over the time with delta distance higher than what the car has moved.

6.4.2. Module fundament

It can be distinguish that the algorithm is divided into two. First, for each signal way measurement the necessary calculations are made and then, once the module has all the information needed, it determines the movement state of the object detected in front of the car depending if the car is moving or is at standstill.



The variables that are taken into consideration in order to later determine the state of the measurement are:

- *Freespace:*

  This Boolean determines if there is any object detected or there is freespace in front of the sensor.

- *Variation of actual measurements:*

  Taking into account the previous measure the following deltas are calculated

  - Delta time (s)
  - Delta car speed (m/s)
  - Delta object speed (mm/s)
  - Delta echo distance (mm/s)

- *Car movement:*

  The car longitudinal movement (in mm) is calculated taking into account the speed of the car obtained from the input of the module and from the delta time previously calculated.

- *Speed of the object:*

  Taking into consideration the driven distance of the car (car longitudinal movement), the difference between the distances measured by the sensor (Delta distance) and the period of time during two consecutive timestamps (Delta time), the speed of the object driving in front of the host car is estimated.

As soon as the module has all the variables calculated, the algorithm determines the movement state of the object detected by the ultrasonic sensors. First of all, the code distinguishes between standstill and free driving because the states depend also on this condition.

On one hand, if the car is in standstill there are three different scenarios depending on the difference between the distances measured in two consecutive timestamps: higher, lower or equal to zero. If the delta distance is equal to zero, as it will be explained later in the section “problems considered in the code”, the state remains the same as in the previous timestamp. Furthermore, if the delta distance is lower than zero it can only mean that the detected object is moving towards the host car or the object is not moving but there are deviations in the measurements. Moreover, the third possible scenario can be either the object is dynamic and is departing from the host vehicle or it is a static object but the limitations of the sensor lead to little deviations in the measurements. In addition, it has to be remarked that when the delta distance is higher or lower than zero the method is able to notice if the measurement is an invalid measure, that it is also explained in the following section of the report.

On the other hand, when the host car is moving free the algorithm becomes more complex as there are many different possible cases and states.

Following the scheme explained in standstill, during free driving the algorithm is also first based on the delta distance. If the difference between the distances of two consecutive measurements is equal to zero, the state remains as the preceding one. However, if the delta distance is higher or lower than zero, there are three different ways to define the state of the detected object:

- *Based on previous moving state*

  The previous measurement led to a moving state and based on it, the code has certain conditions to determine the next moving state.
- *Based on previous standstill state*

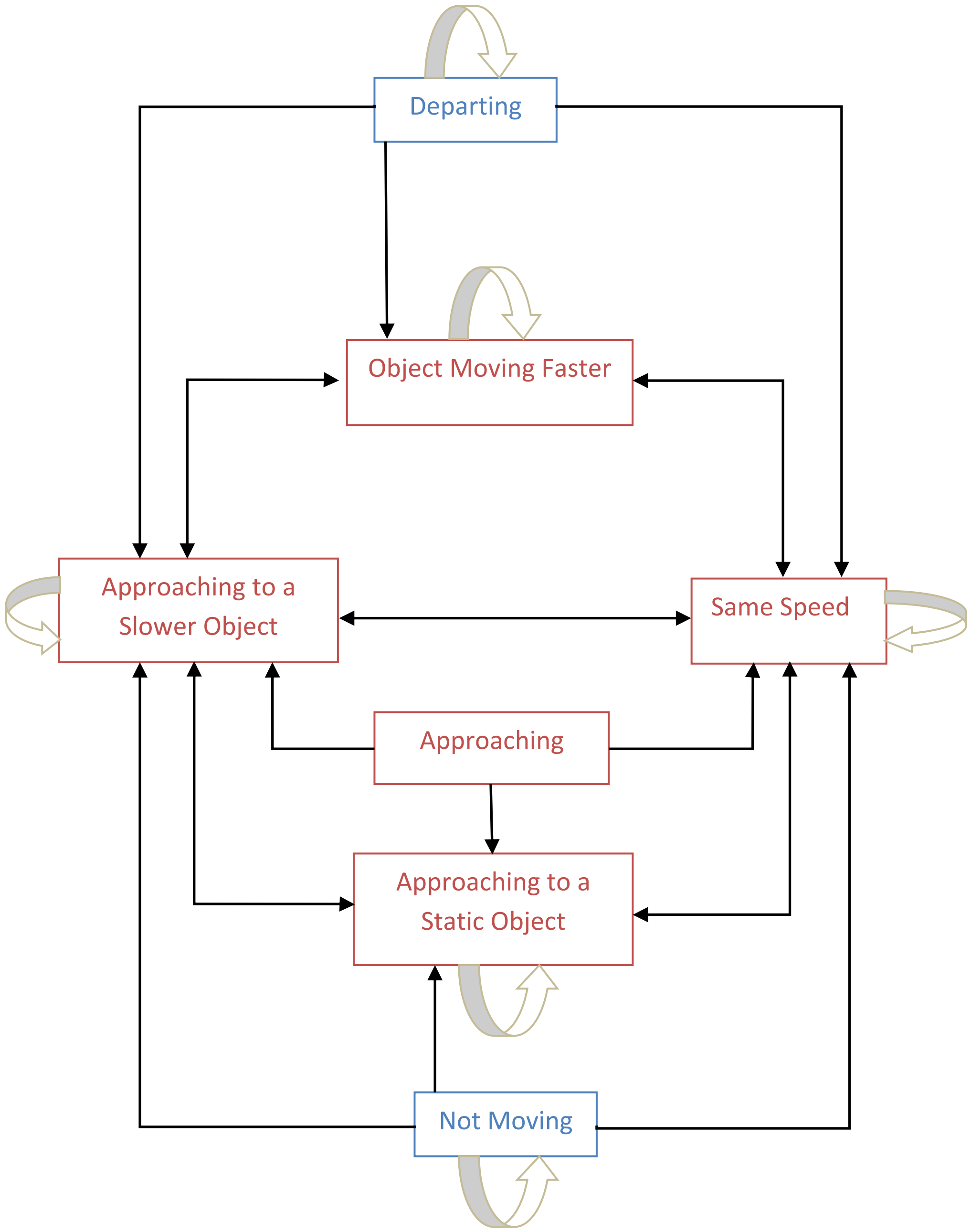
  The previous state is not a moving state because in the precedent measurement the host car was in standstill. However, the standstill sate can be useful to define the first moving state.

- *No previous information*

  In this case, there is no previous information which means that the preceding state is "Not enough information". That is why the conditions that determine the state are based only on the information of the actual measurement.

In each way there are different conditions that have to be fulfilled in order to determine the new state. It is essential to remark that when the code is based on previous states it follows a state machine. This means, it is not always possible to change from one state to another without having an intermediate state. For example, if the object detected by the system is previously defined as a dynamic object driving with a higher speed it is not possible that in the next measurement the same object would be static and therefore determine that the host car is approaching to a static object. In order to go from *Object moving faster* to *Approaching to a static object* state it is obligatory to be in between in the *approaching to a slower object* or *same speed* state.

The next figure shows the different changes ("State Machine") that are enabled while the car is free driving:



Furthermore, it is necessary to explain the different conditions that have to be fulfilled in order to determine one state or another:

- *Approaching to a static object condition*

  In order to determine that the car is approaching a static object, the difference of distance between two consecutive measurements shall be similar to the distance driven by the car in this period of time. That is why the condition that needs to be fulfilled is that the distance approached by the object is between 0.8 and 1.32 times the distance driven by the host car with contrary sign. These values were determined by analyzing real traces and assuming deviations on the accuracy of the sensor measurements.

  Moreover, when the previous state is already approaching to a static object, in the next measurement the state remains likely the same, that is why the margin changes from 1.32 to 1.4.

- *Moving faster conditions*

  This state should be only reached when the delta distance is positive. However, the code also considers the possibility of little deviations due to the limitations of the ultrasonic sensors when the delta distance is also negative.

  Then, when delta distance is positive and the condition for same speed is not fulfilled or delta distance is negative but a very low value and the previous state is already moving faster, this state is set.

- *Same Speed condition*

  When the object in front of the host car is a dynamic object driving with a similar speed, both vehicles should advance a similar distance between two timestamps. As the speed of the object is previously calculated with the delta distance, this state is reached if the speed of the object is between 0.9 and 1.1 times the speed of the host car. These values were adjusted after analyzing real traces.

- *Approaching to a slower object conditions*

  If the delta distance between two measurements is negative, it means that the car is probably approaching an object. If the conditions of approaching to a static object or to an object moving with the same speed are not fulfilled, the only option left is to determine that the host car is approaching to a dynamic object that is moving in front of it with a lower speed.

- *Against condition*

  This state is reached if in two consecutive measures the delta distance is negative and higher than the distance moved by the car. It has to be confirmed in two consecutive measures in order not to interpret wrongly an invalid measure.

The detailed flow diagram of the algorithm can be found in Appendix 2.

6.4.3. Problems considered in the code

It is important to contemplate within the code the limitations of the information provided by the ultrasonic sensors in order to create a reliable code. In addition, it is essential to understand that the final decision of braking will highly depend on the output provided by the module that is why the code has to be 100 % reliable.

Therefore, the different problems taken into consideration are:

- *Invalid measures*

  As it is known, ultrasonic sensors does not supply always solid information due to wrong reflections. That is why, detecting during the actual measurement that it has to be set as invalid is one of the highest challenges of the module.

  These invalid measures, or jumps, have to be detected either the delta distance is positive or negative and the car is in standstill or moving.

  - Delta distance is negative and the car is moving

    A car can approach another car that is driving with a slower speed, a static object or another car driving against it. In the first situation the delta distance should be less than the distance moved by the host car, in the second, the distance droved by the car should be similar to the delta distance with opposite sign and in the third the delta distance should be higher than the distance moved by the host vehicle. However, it is important to distinguish between the third situation and an invalid measure. Therefore, it is assigned invalid measure when the absolute value of the delta distance is greater than 30% of the distance moved by the car and it only happens once.

    In addition, it is also considered invalid measure when the difference of the object speed between two consecutive timestamps is greater than 4.1 km/h. This condition was considered because it is very difficult for a car to change its speed so much during a short period of time (delta time usually is around 300 ms). Moreover, this value (4.1 km/h) was assumed based on the real traces recorded in a real car.

Furthermore, if the previous state determines that the object was moving faster than the host car and in the actual measurement the delta distance is negative, it can only mean that the car has slow down. Therefore, if the delta distance is similar to the distance moved by the car it would lead to the state approaching to a static object and this is not feasible.

- Delta distance is positive and the car is moving

  Following the strategy explained when delta distance < 0, if the difference of the speed of the object driving in front of the host car varies more than 3,8 km/h between two timestamps, the module assumes that it is a jump and it classifies the measure as an invalid measure.

  Moreover, with the calculation of the relative speed between the host car and the detected object, the relative movement is calculated. If the delta distance between two consecutives measurements is higher than 1,7 times the relative movement, the module triggers the flag of invalid measure.

- Delta distance is positive and the car is in standstill

  Based on real traces, if the in the previous measurement the object in front of the car was not departing and the delta distance is greater than 155 mm the module considers the new measurement as a jump.

  In addition, if the difference of speed of the detected object is higher than 2 km/h, the invalid measure flag is also triggered.

- Delta distance is negative and the car is in standstill

  The conditions do not change dramatically from the ones described when the delta distance is positive. However, in this case, it is considered if the previous state is object not moving or object departing. In the first case, if the difference of distance is lower than -150 mm, the measure is classified as invalid. On the other hand, if the previous state is "Object departing" and the delta distance is lower than -300, the module triggers the "*invalid measure*" flag. The ranges have been set according to real traces.

- *Missed measurements (again seen)*

  In the case presentation (Situation 2: driving behind another car) it was presented a missed measurement. This means, there is an object but the ultrasonic sensor missed it what leads the module to think that there is not an object in front of the car.

  These missed measurements can not be solved, however the code, by the “*again seen*” flag, is able to notice in the next step that the previous measure was missed and rely on two measures before in order to determine the next state.

- *Data types*

  In a first approach, as the code pretended to be as accurate as possible, most of the variables were defined as float. However, in order to test the code in a real car, floats consume a lot of time and can make the system very slow. Therefore, the data types were changed to unsigned and signed variables making sure that the resolution of the results were not minimized.

  Once this change was made, an exhaustive analysis was made to assure that there were any overflows of the variables.

- *Accuracy of the sensors in high ranges (>3000mm)*

  Ultrasonic sensors have a limited distance to measure. The sensors used by this system can accurately measure distances in a range 0.5 and 2.5 m. They can measure longer distances but the accuracy is minimized, that is why this phenomenon has to be taking into account within the code.

  It can be appreciated in the code that some of the conditions can be fulfilled only if the distance measured is less than 2.5 meters. In addition there is a special condition when the delta distance is greater than 0 mm, the measured distance is over 3 meters and the previous state is approaching to a static object.

  If the module detects from far away that the car is approaching to a static object and the speed is high, it is very important that wrong measures do not interfere in the assignation of a state. This condition highly avoids these cases.

- *Delta distance equal to zero*

  Normally, even if the object detected by the ultrasonic sensors is a dynamic object moving with the same speed, the distance changes from one measurement to the next one due to the accuracy limitations of the ultrasonic sensors. Therefore, when the difference of distance between two consecutives measurements is equal to zero, it does not mean obligatory that the state must be “same speed”. That is why, when this case happens, the module maintains the previous state not considering this measure remarkable. This situation appeared due to a internal software issue.

6.5. New function development: Speed braking

6.5.1. Function´s goal

The second target of the thesis is to develop a speed braking function taking into consideration the information supplied by the dynamic object classification module developed as first target.

The goal of the function is to send the activation order to the brake with the appropriate deceleration value taking into account if the detected object is static or dynamic and, in the second case, if the car in front is not at standstill but just slower, the target is to brake down to the same speed as the other car without crashing it.

6.5.2. Fundament

First of all it is necessary to distinguish between full and speed braking. Full braking is applied when the application triggers the brake and the dynamic object classification module confirms that the object detected in front of the host car is a static object. In this case the deceleration applied is a constant and corresponds to the maximum value -9000 mm/s^2.

On the other hand, speed braking is applied when the object detected by the ultrasonic sensors is dynamic, the module indicates that the object is driving with lower speed and the time to collision is lower to 5 seconds. This value has been determined in order to not have very high decelerations and make the brake more comfortable for the driver.

To calculate the time to collision, information from the dynamic module such as the speed of the object, the distance and the speed of the host car, is required. With this information, the following formula is applied:

$$TTC = \frac{Dist}{(V_c - V_o)}$$

Where:

- TTC = Time to collision
- V_c = Speed of the host car
- V_o = Speed of the object detected
- Dist = Distance to the object

Once the braking order is set up, it is necessary to determine which deceleration is appropriate. The first approach was setting it to a constant equal to -500 mm/s^2. However, this solution is only suitable when the speed of both cars and the difference between them are not very high. In this case, this deceleration is not enough and the host car crashes the vehicle driving in front of it.

The second approach was to determine three different decelerations taking into account the time to collision already calculated. The three different levels were:

- If TTC < 2s:
    - deceleration = -1000 mm/s^2
- if 2 < TTC < 4s:
    - deceleration = -700 mm/s^2
- if 4 < TTC < 6:
    - deceleration = -500 mm/s^2

This second solution worked for the range of speeds needed however, it is not as accurate as it can be that is why a final solution has been implemented.

The final solution consists on applying the “Uniform Accelerated Motion” speed formula:

$$V_f = V_o + a \cdot t$$

Where:

- V_f = Final/Desirable Speed (mm/s)
- V_o = Initial Speed (mm/s)
- a = acceleration (mm/s^2)
- t = Time (s)

In order to apply this formula the object speed is considered the desirable speed as the host car should brake until it reaches this speed, the initial speed is the speed of the host car and the time is the time to collision already calculated.

With all the information the suitable deceleration in mm/s^2 for each situation is calculated:

$$Deceleration = \frac{V_{obj} - V_{vehicle}}{TTC}$$

In order to have some safety time, the deceleration applied is 10% more than the value calculated.

6.5.3. Problems considered

- *Side objects*

The main problem presented is that the dynamic module takes into account raw data and this means that it can detect that there is an object in a certain distance, but not exactly where. Therefore, as it can be seen in the next figure, if there is a car driving in the side lane of the road with a lower speed, the module detects it as a dynamic object driving slower and the braking is triggered even if the car is not located in the driving area of the host car.

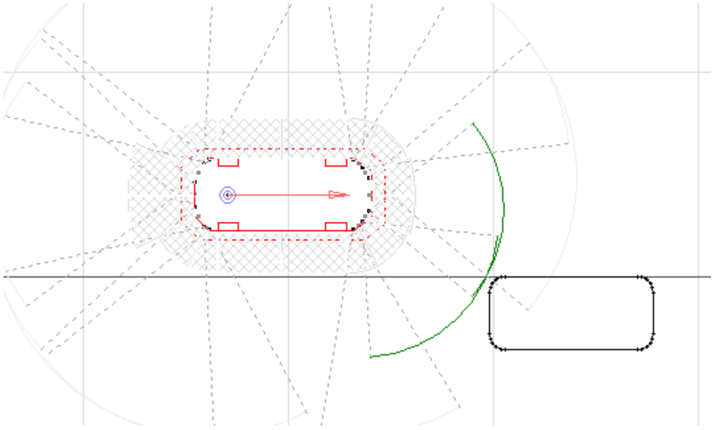


Figure 71.- Side Object detection

The solution implemented to solve this issue has been a function that detects if the object is located in a side of the car taking into consideration the difference between the distances measured by the two external sensors of the front bumper. When an object is placed in the side of the car, the sensor closer to this side measures a lower distance than the opposite external sensor. Based on simulations done with VS6, the difference between distances set to determine if the object is a side object is 375mm for a specific car configuration from the first Client. In addition, the distances of the side sensors have been taken into consideration. If the object is located in a side of the host car, the sensor of this side will detect it but the opposite won´t. This second solution is 100% reliable only if there is an object in one of the sides but not if there are two objects in both sides at the same time.

- *Object´s Velocity and Distance*

As the dynamic module works with the information provided for four different sensors, that is why in order to supply a single object speed and distance to the speed braking function, an average between the four signals has to be made.

However, it is important to consider that there are wrong or invalid measurements and they should be avoided in order to supply information to the new function. The function in charge of this checks if the actual measure of each sensor has been considered as invalid measure by the dynamic module and in this case it doesn´t take it into account to calculate the mean distance and mean object speed.

- *Hysteresis*

In both full and speed braking cases, a hysteresis condition is applied in order to maintain the brake until the condition to release the brake is fulfilled.

The full braking condition to release the brake is reaching standstill. On the other hand, the condition for speed braking is to reach a lower speed than the 97% of the object´s speed.

6.5.4. Corner cases

- *Driving between 2 close objects*

When the car has to drive between two objects which separation between them is a bit higher than the weight of the car, the actual application sometimes triggers the brake and as the dynamic module determines the object as static the system brakes even if there is enough space to drive between both objects.

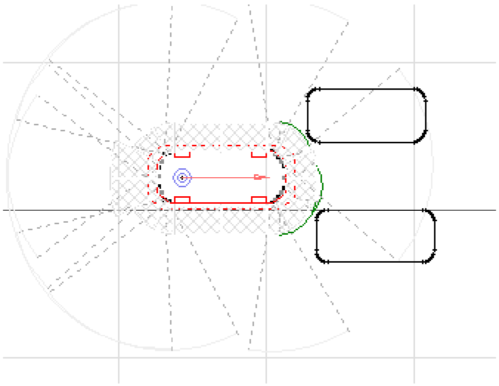


Figure 72.- Corner case 1

This use case is also part of the cases where the function in charge of detection side objects can not identify them properly because each side sensor will detect one of them.

- *Car crossing perpendicular*

When the car is driving and another car crosses perpendicular, neither the application nor the dynamic module is able to detect it properly in 100% of the cases that is why the system does not always brake on this scenarios.

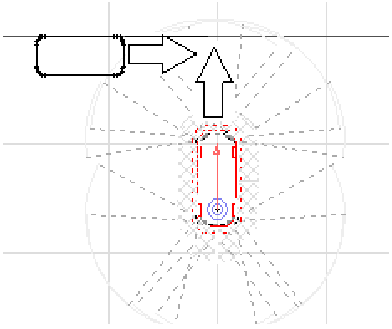


Figure 73.- Corner case 2

- *High speed differences while driving with high speed*

Ultrasonic sensors can measure a limited distance. Therefore when the car is for example driving with 14 km/h and the car in front of it drives with a speed of 2 km/h, taking into consideration that the minimum detection distance is around 3,5 - 4 meters, the time to collision is around 1-1,2 seconds and that is not enough time to brake until the object speed before crashing it due to system limitations.

In general, the function works correctly in every speed until 12 km/h. From 13 km/h the algorithm does not support high speed differences between both cars, which means that the car in front of the host vehicle has a speed lower than 3-4 km/h.

7. Results

7.1. First target: Dynamic object classification module

In order to interpret the result of the module, around 150 real traces supplied by three different clients have been analyzed.

First of all, the traces were run with the standard code in order to see how the statistics were without implementing any new code.

Then, the new algorithm was included in the application and all the traces were run again.

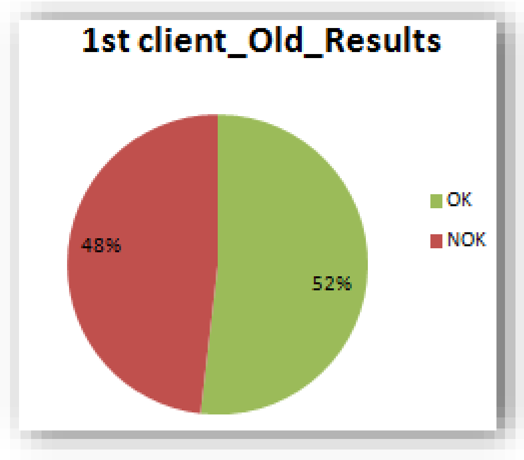
It is important to remark, that the traces were not only recorded in traffic red light situations or in open road, but also driving against a wall. The goal of the module is to avoid false positives during free driving but it is also important to check if when the host car is driving towards a wall it triggers the brake.

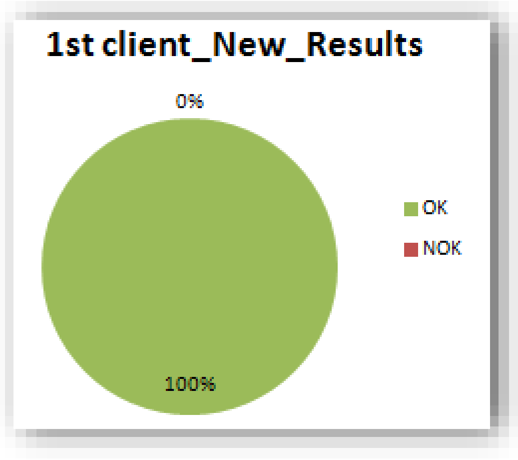
The first comparison is made distinguishing between the different cars:

- *1st Client results:*

Table 5.- 1st client results

		Old Algorithm		New Algorithm	
		Number of traces		Number of traces	
	Results	OK	NOK	OK	NOK
1st Client	Against Wall	10	0	10	0
	Behind Car	0	5	5	0
	Vehicle in Front	4	8	12	0
	TOTAL	14	13	27	0
	Percentage	52	48	100	0



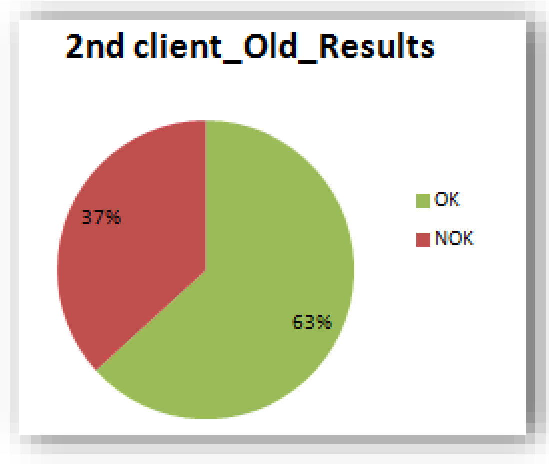


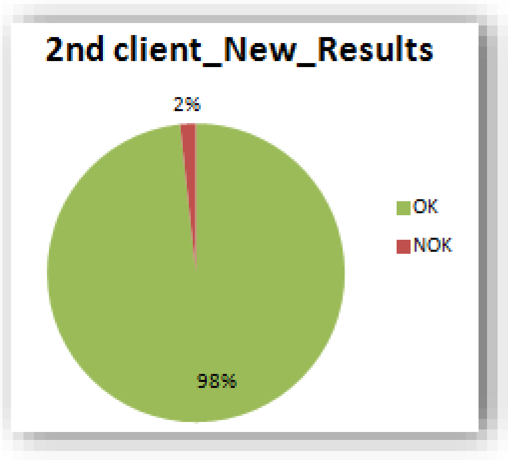
As it can be seen in the charts, the implementation of the new code improves dramatically the results of the traces in the first Client car. The results of the new module are 100% OK in the three different situations while with the standard code only 52% of the traces where OK, being in the situations where the host car was driving behind another car the ones that are remarkably improved. Moreover, it can be also noticed that when the brake must be triggered (Against wall situation) the new code does not introduce any error.

- *2nd Client results:*

Table 6.- 2nd Client results

		Old Algorithm		New Algorithm	
		Number of traces		Number of traces	
	Results	OK	NOK	OK	NOK
2nd Client	Red Light	3	13	16	0
	Against Balloon Car	45	0	45	0
	OpenRoad_160310	16	13	29	0
	OpenRoad_160208	5	7	10	2
	OpenRoad_160122	5	5	10	0
	OpenRoad_160118	0	2	2	0
	OpenRoad_160115	0	2	2	0
	OpenRoad_160805	0	1	1	0
	TOTAL	74	43	115	2
	Percentage	63	37	98	2





For the 2nd client traces, more were provided distinguishing between the situations where the car is approaching to a “wall”, in this case was a balloon car, traffic red light situation and traces recorded in open road.

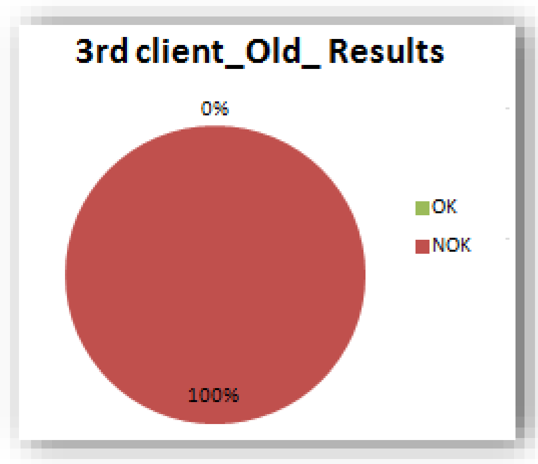
As it can be seen, the improvement is also remarkable both in open road and traffic red light cases increasing the percentage of OK traces by 34% and to overall 98% OK traces.

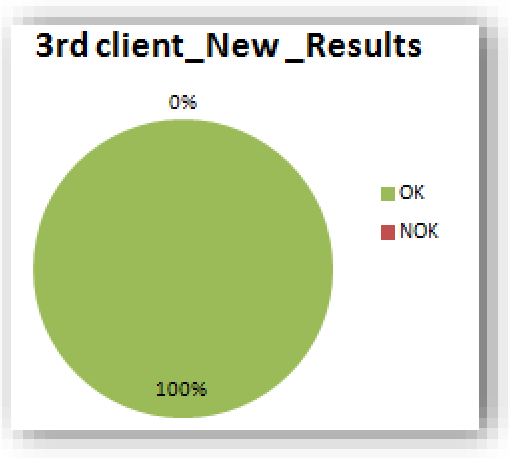
On the other hand it is important to analyze the two traces where the results with the new algorithm continue being NOK. In both traces, it appears false echoes during 4 consecutive cycles approaching to our car that fit exactly with a static object. These echoes can appear for example if there are some fumes coming from a sink of the road. In these 4 cycles the application triggers the brake and as it matches with a static object, the algorithm classifies them also as static, set the “*Dynamic Flag*” as TRUE and brakes.

- *3rd Client results:*

Table 7.- 3rd Client results

		Old Algorithm		New Algorithm	
		Number of traces		Number of traces	
	Results	OK	NOK	OK	NOK
3rd Client	Open Road	0	4	4	0
	TOTAL	0	4	4	0
	Percentage	0	100	100	0





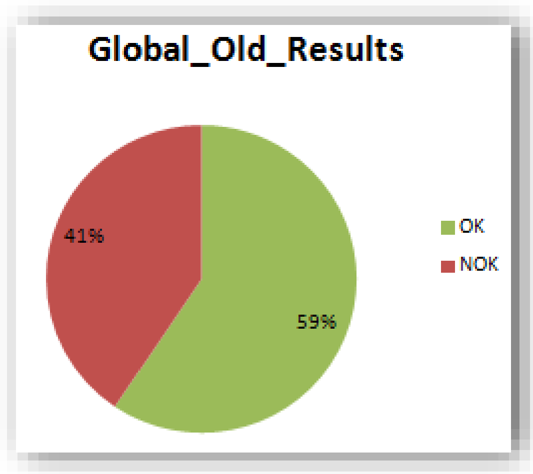
The traces supplied by the thrid Client team where only four but full of false positives. When the traces were run with the new algorithm all the false positives were corrected.

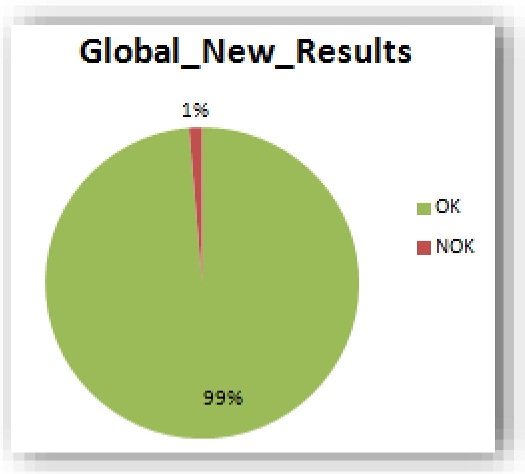
This means that the new code improves in 100% the results obtained.

- *Global results:*

Table 8.- Global results

	Old Algorithm		New Algorithm	
	Number of traces		Number of traces	
	OK	NOK	OK	NOK
TOTAL	88	60	146	2
Percentage	59	41	99	1





In a global analysis, it can be appreciated the high improvement that the new algorithm introduces increasing from 59% to 99% the number of OK cases.

Moreover, an analysis depending on the expectation of the brake system has been done.

- *Expectation: Brake*

Table 9.- Results according to brake expectation

		Old Algorithm		New Algorithm	
		Number of traces		Number of traces	
	Expectation	OK	NOK	OK	NOK
Against Wall	Brake	10	0	10	0
Against Baloon Car	Brake	45	0	45	0
OpenRoad_160115	Brake	0	2	2	0
	TOTAL	55	0	55	0
	Percentage	100	0	100	0

It can be appreciated that when the expectation of the system is “to brake” the new algorithm also triggers the brake. This means that the brake function keeps working correctly in the situation where the host car is approaching to a static object.

- *Expectation: do not brake*

Table 10.- Results according to not brake expectation

		Old Algorithm		New Algorithm	
		Number of traces		Number of traces	
	Expectation	OK	NOK	OK	NOK
Behind Car	Don´t Brake	0	5	5	0
Vehicle in Front	Don´t Brake	0	12	12	0
Red Light	Don´t Brake	3	13	16	0
OpenRoad_160310	Don´t Brake	16	13	29	0
OpenRoad_160208	Don´t Brake	5	7	10	2
OpenRoad_160122	Don´t Brake	5	5	10	0
OpenRoad_160118	Don´t Brake	0	2	2	0
OpenRoad_160805	Don´t Brake	0	1	1	0
OpenRoad_3rdClient	Don´t Brake	0	4	4	0
	TOTAL	29	62	89	2
	Percentage	32	68	98	2

On the other hand, when the expectation is “not brake”, the implementation of the new module increases the OK results from 32% to 98%.

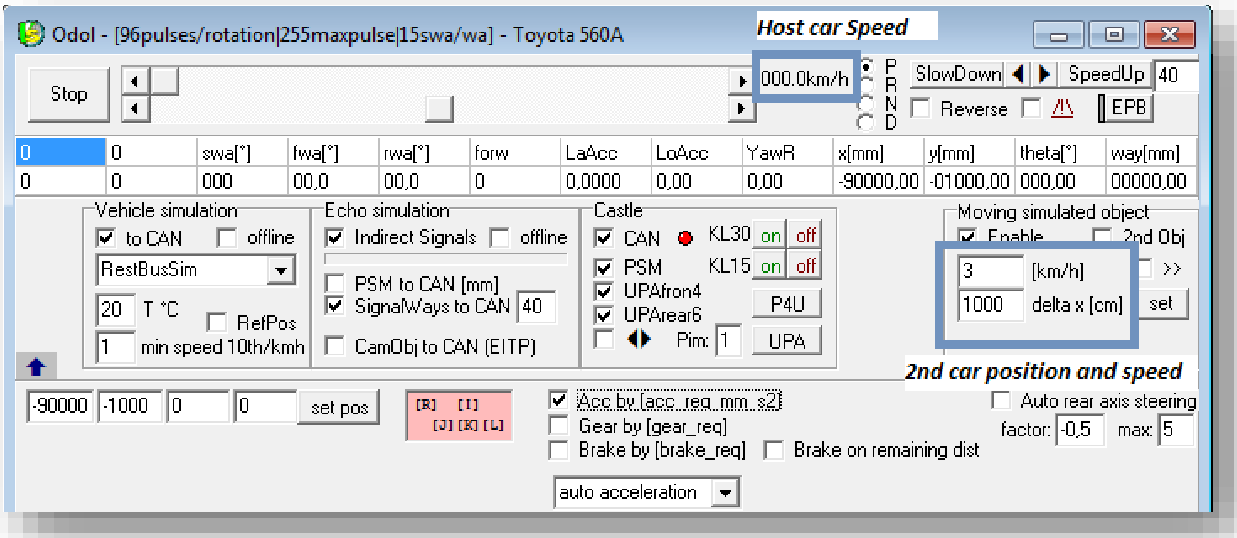
The complete analysis is attached in Annex 3, where it can be found the results of the traces and the final results overview.

In addition, is important to remark that the algorithm was tested not only in a test environment but also in a real car. The code was integrated in the whole application and was provided to the third client. This client tried it on their test fields and gave us positive feedback. That means that the final algorithm developed during this thesis would be used by real automotive companies.

7.2. Second target: Speed braking functionality

In order to evaluate the results of the new function, a simulation environment had to be set up. This tool is called VS6. It simulates echoes that are provided to a real ECU in which the algorithm is running.

In the next figure it is presented one of the panels from VS6. With it, it is possible to control the speed and position of two different dynamic objects.



Odol - [96pulses/rotation|255maxpulse|15swa/wa] - Toyota 560A

Host car Speed

000.0km/h

SpeedUp 40

0	0	swa[°]	fwa[°]	rwa[°]	forw	LaAcc	LoAcc	YawR	x[mm]	y[mm]	theta[°]	way[mm]
0	0	000	00,0	00,0	0	0,0000	0,00	0,00	-90000,00	-01000,00	000,00	00000,00

Vehicle simulation
☑ to CAN ☐ offline
RestBusSim
20 T °C ☐ RefPos
1 min speed 10th/kmh

Echo simulation
☑ Indirect Signals ☐ offline
☐ PSM to CAN [mm]
☑ SignalWays to CAN 40
☐ CamObj to CAN (EITP)

Castle
☑ CAN ● KL30 on off
☑ PSM KL15 on off
☑ UPAfron4
☑ UPArear6
☐ Pim: 1

Moving simulated object
☑ Enable ☐ 2nd Obj
3 [km/h]
1000 delta x [cm]

2nd car position and speed

-90000 -1000 0 0 set pos

☑ Acc by [acc_req_mm_s2]
☐ Gear by [gear_req]
☐ Brake by [brake_req] ☐ Brake on remaining dist
auto acceleration

☐ Auto rear axis steering
factor: -0,5 max: 5

Figure 74.- VS6 control panel

- *Speed braking:*

As it can be seen in the following figure, in the first situation simulated, two cars are placed one in front of another and both start driving. The one that is in the front starts with higher speed in order to leave some space between both cars and at certain point its speed is reduced, reaching a lower speed than the host car leading to a brake of it.

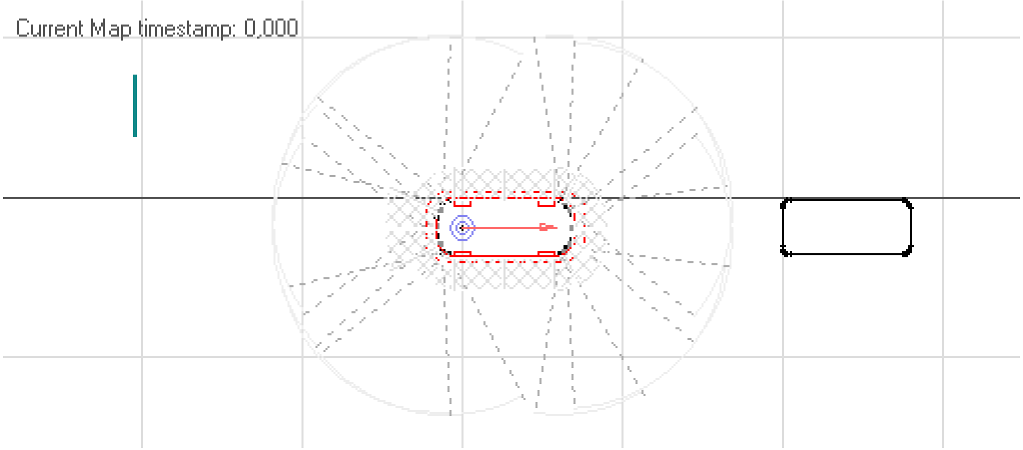


Figure 75.- Scenario 1: Speed braking

In order to verify the proper functioning of the function, the speed of the first car is reduced to different values and using canoe it can be confirmed if the speed of the host car has been properly reduced.

In the following images it can be seen three different signals extracted from CANoe:

1. Speed of the host car (km/h)
2. Brake status (Active or not)
3. Brake deceleration (m/s^2)

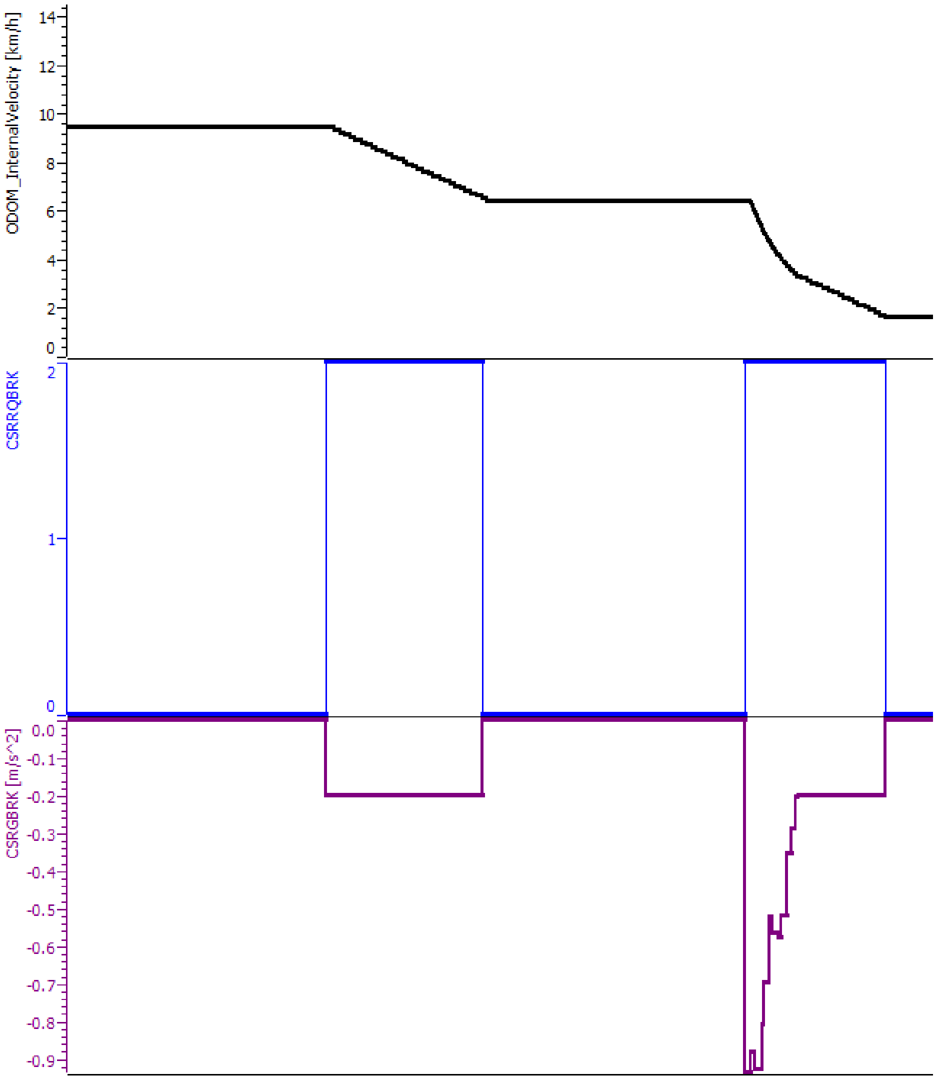


Figure 76.- Speed braking results 1

In this first figure it is presented speed braking examples which starting point is: the host car is driving 9,8 km/h and the car in front reduces its speed to 7 km/h. When the host vehicle detects it and the brake is triggered, it can be seen that the deceleration is constant and equal to -0,2 m/s^2 ,which is the minimum deceleration value, and it maintains it until the speed is under 7 km/h, in this case 6,4 km/h. Moreover, the speed of the second car is again reduced to 2 km/h and as in this case the difference between speeds is higher, the deceleration needed is also increased starting with a high value and decreasing over time until the speed of the car is 1,6 km/h.

In this second figure presented below, it can be seen two speed brakes from 14 km/h to 7 km/h who requires a higher deceleration, another from 7 km/h to 4km/h and at the end of the trace it can be distinguished a full speed represented with a deceleration of -9 m/s^2 which means that the car driving in front ended in standstill.

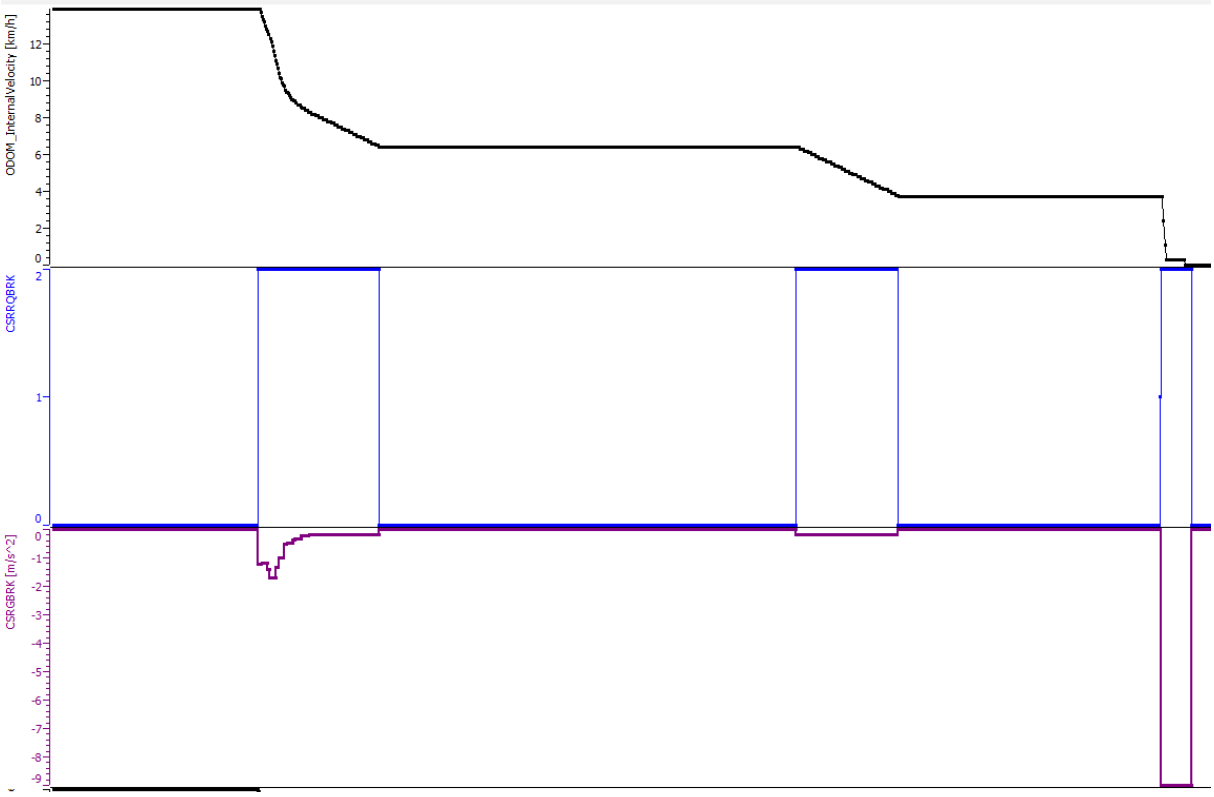


Figure 77.- Speed braking results 2

- *Bicycle*

This situation consists on driving behind a bicycle in order to confirm if the speed braking function is able to brake on thinner objects.

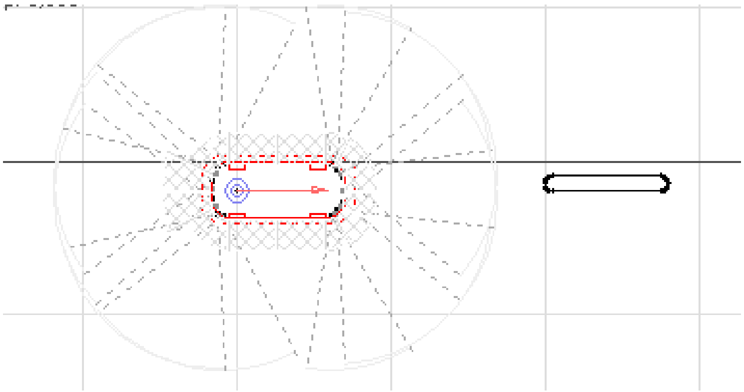
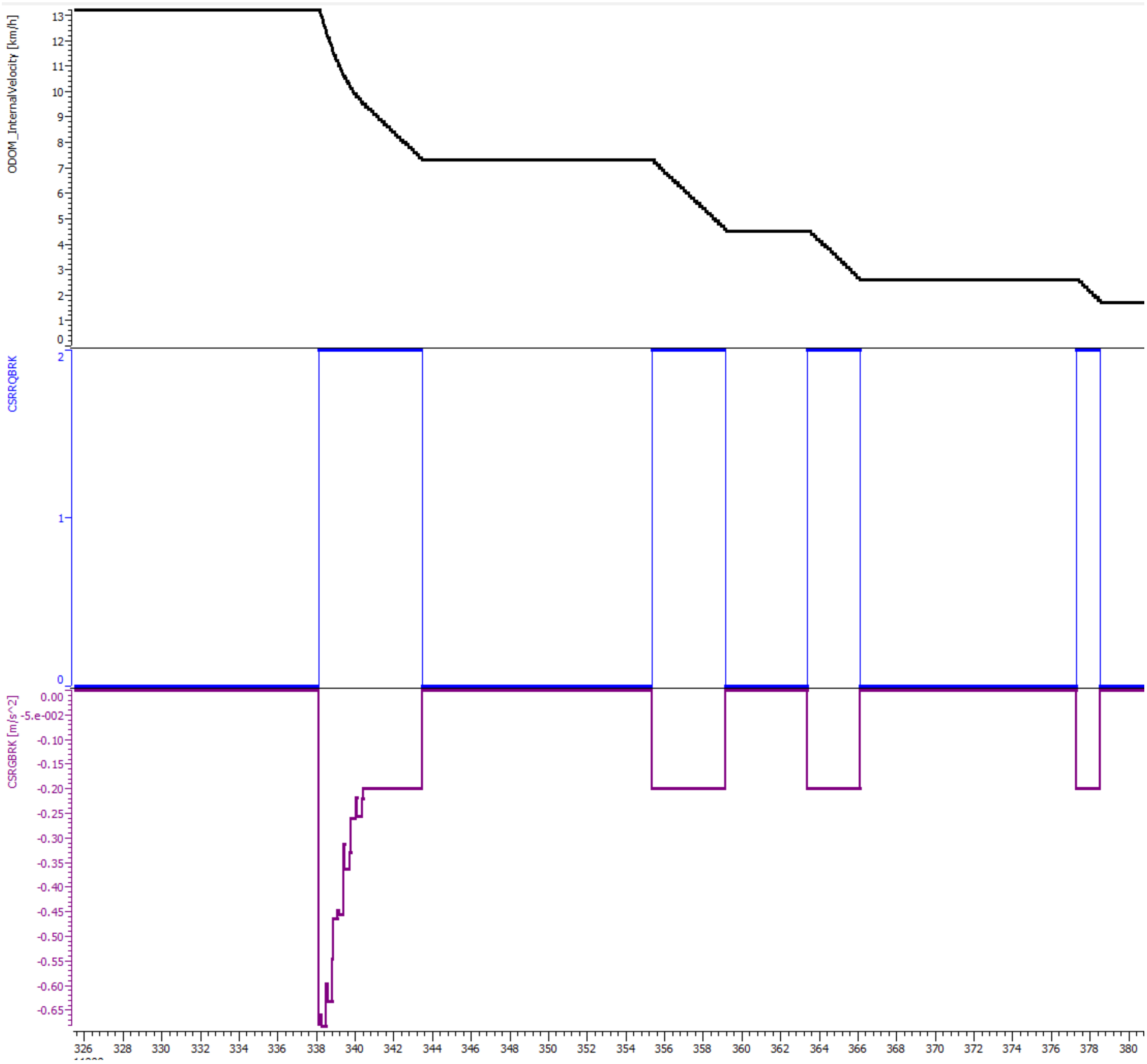


Figure 78.- Scenario 2: Bicycle

The simulation case is similar to the one explain before. A bicycle reduces its speed and it has to be confirmed that the host car brakes properly on it.



As it can be appreciated, the starting speed of the car is 13 km/h while the bicycle is driving 8km/h. When the car gets enough close to the bike it starts braking with a high deceleration that is reduced over the time until 7,4. After this first brake, the bicycle changes its speed to 5 km/h then to 3 km/h and finally to 2 km/h. These variations produce three brakes on the host car from 7,4 km/h to 4,6 km/h, from 4,6 km/h to 2,7 and finally from 2,7 km/h to 1,8 km/h. The difference that can be appreciated in these three brakes is the constant value of the deceleration equal to -0,2 m/s^2 . As it was already explained in the previous scenario when the deceleration calculated does not reach this minimum value, the deceleration is set to this constant value.

- *Side Objects*

As it was explained in section 6.5.3, dealing with side objects was one of the challenges of this function. In order to simulate them two different situations are proposed.

The first situation, represented by the following figure, is the typical parking scenario where there are static objects and a slot in between.

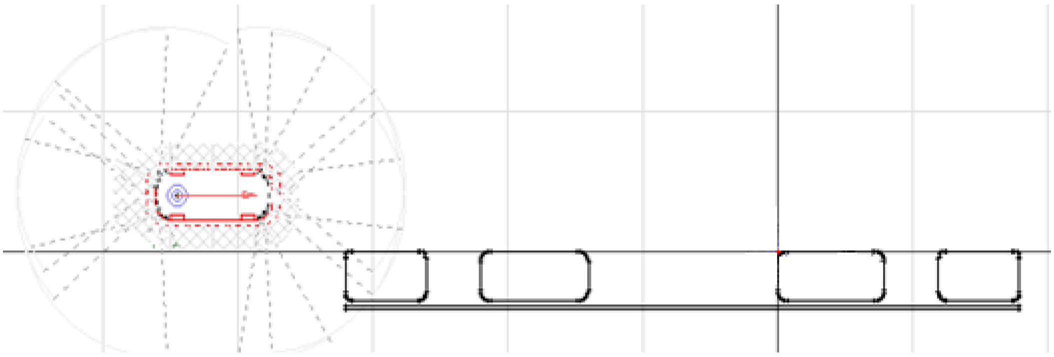


Figure 79.- Scenario 3: Static side objects

Simulations with each speed from the brake range (1 to 15 km/h) were done and it was confirmed that the function is not braking in this kind of situations.

In addition, the next figure shows another possible scenario where the function should not brake. In this situation there is one static object placed in one side and a dynamic object placed on the other.

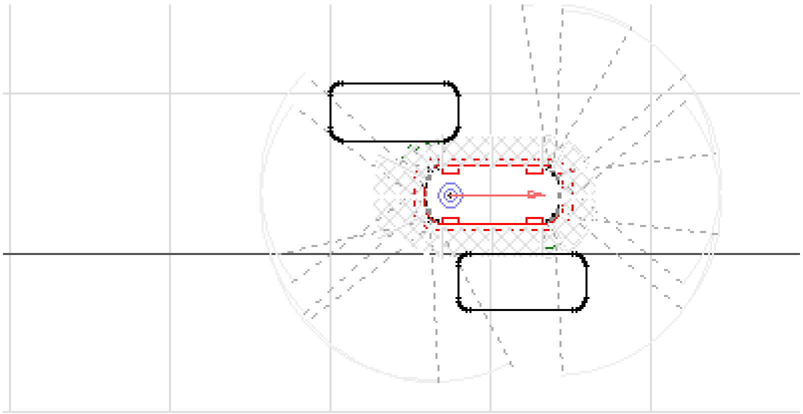


Figure 80.- Scenario 4: Static and dynamic side objects

There has been done simulations with different speeds for both host car and dynamic object and it has been verified that if the vertical distance between cars is more than 75 cm, the function does not brake on this objects.

8. Conclusions and future development

8.1. Conclusions

The first aim of this thesis was to develop a module with the information supplied from the ultrasonic sensors able to detect and classify the movement of dynamic objects. Its goal was avoiding false positives of the brake system during free driving in traffic red light or driving behind other car scenarios.

The proposed module considers the information provided by the four front sensors of the vehicle and it is able to identify if the object located in front of the car is static or dynamic and, in the second case, it also calculates the speed of the other vehicle. Once the code has been realized, it has been simulated with 148 traces supplied by the three different client´s cars and it has been confirmed that it improves the emergency braking decision 39% which in dynamic scenarios represent a 66% and which means an overall OK rate of 99%.

On the other hand, the second main purpose of the thesis was to realize a speed braking function able to make the vehicle brake with the appropriate deceleration in situations where the vehicle in front of the host car drives with a lower speed using the information provided by the dynamic object classification module already developed. In order to verify the function different scenarios have been simulated such as driving behind a vehicle or a bicycle and driving with static and driving objects in the side lanes. These simulations confirmed the correct functioning but also showed the weakness of the system in situations like a car crossing perpendicular, car driving very slowly (<3 km/h) when the host vehicle has a high speed (>13 km/h).

However the system is not perfect. Due to the limitations of the ultrasonic sensors the accuracy of the calculations is limited and can sometimes lead to wrong measurements and consequently, wrong statements.

To sum up, both targets were reached showing 99% of correct results of the module developed and including a complete new function to the application.

8.2. Future development

As ultrasonic sensors have many limitations that is why including in the function information supplied by other sensors could improve the accuracy of the function developed.

Moreover, a side object detection based on the raw distances provided by the sensors could be improved by including the physical measures of the car and the location of the sensors. With this information the value of the difference between measures of the side sensors that determines if an object is a side object would be more accurate. In addition, some logic should be introduced in order to detect if there are objects in both sides at the same time.

Furthermore, another way of improving the system would be developing a dynamic map where all the information is stored and will make the system more reliable in situations that deals with dynamic objects.

In addition, the actual system has some corner cases as the explained in chapter 6.5.4 which should be solved. When two cars are driving against each other the system does not always detect it so it can be added in the speed braking function as another condition to active a brake.

From another point of view, the same logic can be implemented with the rear bumper sensors in order to use it also when the host vehicle is driving backwards.

9. Bibliography

- http://www.autoalliance.org/era-of-innovation
- http://www.valeo.com/en
- Valeo ULS Sensor Basics Training
- Valeo OBJD4 Training
- Valeo ULFX Training
- Valeo DAS Architecture Training
- Valeo DAS Product Training
- Valeo THAS-IEBM Training