



CHALMERS



The Car That Reverses Safely

Batchelor's Thesis of Autonomous Emergency Braking

SSYX 02-14-18

Björn Nilsson,	880302
Markus Carlander,	921027
Magnus Törnqvist,	890724
Niklas Westerlund,	900921
Nils Nordeman,	920430
Viktor Palmquist, Berndtsson,	920723

Abstract

This thesis deals with the development of an active safety system that autonomously can brake a reversing car if it is about to collide with an obstacle. The goal is to develop a system that produces consistent safety margins to the object regardless of the starting velocity but also a system that does not intervene too much with the driver. In the development of this project a rig was constructed to accommodate six external ultrasonic sensors used to measure the distance to obstacles. Different braking strategies have been evaluated and tested in order to find out if a late, hard braking strategy works better than a smooth long.

The strategy that implements a late and hard braking is the one that works the best, in regards of counteracting misuse and unintentional braking sequences. Moreover, this strategy consistently resulted in that the car stopped near predefined stopping distance. The car has been tested in a variety of situations that could pose a problem, and the emergency braking system coped with all situations without any unnecessary interruptions and brakes were only activated when needed.

Sammanfattning

Denna avhandling handlar om utvecklingen av ett aktivt säkerhetssystem för autonom bromsning av en backande bil. Målet är att utveckla ett system som ger konsekventa säkerhetsmarginaler för hinder oavsett utgångshastighet, men också ett system som inte ingriper när det inte är nödvändigt. I utvecklingen av detta projekt har en rigg behövt byggas på vilken de externa ultraljudssensorer, som används för att mäta avstånd, monterats. Olika bromsstrategier har utvärderats för att ta reda på ifall till exempel en sen, hård bromsstrategi fungerar bättre än en längre mjukare inbromsning.

Bromsstrategin som tillämpar en hård och sen bromsning är den som fungerar bäst, med avseende på att motverka att systemet missbrukas och att onödiga ingrepp undviks. Bilen har testats i flera kritiska situationer där systemet skulle kunna få problem och panikbromsen ingrep inte någon gång i onödan utan bara när kollisionrisken var överhängande. Bilen stannar också konsekvent på fördefinierat säkerhetsavstånd, oberoende av ingångshastighet.

Contents

1	Introduction	3
1.1	Goals and Objectives	4
1.2	Problem Declaration	5
1.3	Limitations	5
2	Methodology	6
2.1	Presentation of Hardware	6
2.1.1	Hardware Used in the Project	6
2.1.2	Hardware Setup in the Car	9
2.2	Presentation of software	10
2.2.1	Software Used in the Project	10
2.3	Using Ultrasonic Sensors	11
2.3.1	Maxbotix LV-Maxsonar-EZ0 MB1000	11
2.3.2	Method for Reading of Sensor Values	12
2.3.3	Low Pass Filtering DC Power supply for Sensor system	13
2.4	Rigging Up the Sensors	13
2.4.1	Developing the Sensor Rig	14
2.5	Dynamics of the Brake Process	15
2.5.1	Limitations in Brake Dynamics	15
2.5.2	Brake Scenarios at Different Initial Velocities	16
2.5.3	Complete Theoretical Model for Simulation	19
2.6	Data Gathering for Modelling	20
2.6.1	Onboard Sensors	20
2.6.2	Data Logger	21
2.7	Simulink Model Implemented in the Car	21
2.7.1	Decision Making Process While Reversing Toward a Stationary Object	22
2.8	Brake Strategies	22
2.8.1	Brake Dynamics Test	23
2.8.2	Brake Strategies Test	23
2.8.3	Setup of Braking Tests	23
3	Results	25
3.1	Developed Hardware	25
3.2	Determining of Brake Parameters	27
3.2.1	Results From the Cars On-board Sensors	27
3.2.2	Results From the Data Logger	28
3.3	Results of Simulation	29
3.4	Brake Strategy Tests	29
3.5	Real World Testing of the System	31
3.5.1	Collision Avoidance	31
3.5.2	Brake Margin Continuity	32
3.6	Sensor Stability With and Without Low Pass Filter	33
3.7	Sensor Triggering Method Used	33
3.8	Achieving Autonomous Emergency Braking	33

4	Discussion	35
4.1	Choice of Hardware	35
4.1.1	Recognition of Objects	37
4.1.2	Avoiding Collision of a Moving Object Crossing the Cars Trajectory	37
4.1.3	Avoiding Collision With Object While the Car is Turning	38
4.2	Simulations	38
4.3	Reasons for Few Braking Strategies	39
4.4	Gathering Data Using the Cars Onboard Sensors	39
4.5	Interface Between the Sensor System And The MicroAutoBox	40
4.6	Further Development of Object Positioning and Tracking	40
4.7	The Autonomous Braking System From an Environmental Point of View	42
5	Conclusion	43
	References	44
6	Appendix	45
6.1	Different Sensor Triggering Methods	45
6.1.1	Results of Triggering Methods	46
6.2	Sensor testing	47
6.3	Test Protocols and Comments to Those	51
6.4	Principles of Triangulation	54
6.5	Additional Graphs on Sensor Fluctuations With and Without Low Pass Filter	56
6.6	Arduino Code for Managing the Sensors	56
6.7	Simulink Models and Code	59
6.7.1	Code for Calculation of Critical Distance	59
6.7.2	Simulink State Decider	61
6.8	Simulink Model	62
6.8.1	Simulink Overview	62
6.8.2	Inner Simulink model	62
6.8.3	Simulink UDP Block	63

1 Introduction

Many modern cars are today equipped with systems that while reversing alerts the drivers that they are approaching an object and assists in judging the distance to it. This system works through a sound signal triggered by ultrasonic sensors that intensifies the closer to the object the car gets. Despite that, there are numerous accidents every year that injure pedestrians and/or cause damage to the car which involve insurance companies and costs both time and money. In Sweden, the department of transportation has highly set goals and visions to improve traffic safety. "Nollvisionen" is the basis for the Swedish road safety efforts and means a desire to adapt roads and vehicles so that people are not killed or injured for life in traffic (Trafikverket, 2012). This with consideration taken that people are not perfect and make mistakes. "Nollvisionen" accept that accidents happen but not that they result in death or serious injury. This goes hand in hand with Volvo's strive to develop systems implemented in their cars that not only protects the driver and passenger, but also people in other cars in the traffic environment. Volvo also have the same short term goals as the Swedish department of transportation to build cars that does not kill or seriously injure anyone in traffic (Volvo, 2010).

It is therefore desirable to develop and implement a system that prevents accidents by for example automatically braking the car before any damage is done. An implementation of a system like this would besides preventing injury affect the insurance premium of the car. A car that is less likely to be involved in an accident would simply be cheaper to insure.

This report is a part of a bachelor thesis written at Chalmers University of Technology. The assignment is issued by Volvo and tested and executed on a Volvo S60 that has a rapid prototyping computer used by the automotive industry to test different control software. The computer is called MicroAutoBox II and will hereafter be called MicroAutoBox. The project involves Chalmers students from different backgrounds: electrical engineering, automation & mechatronics and mechanical engineering.

During this project the financial assets available were limited to 5000 SEK. The money was spent on ascertain components and material not already available to the project from Chalmers.

1.1 Goals and Objectives

The main goal of this project is to design and test a system that can detect obstacles while reversing and subsequently apply emergency braking if necessary at speeds normally achieved during reversing. It will also be essential to implement functions that can effectively counteract misuse in the means that the driver put too much trust into the braking system.

It is not possible to read values from the ultrasonic sensors that are installed on the car today. Therefore, external sensors had to be installed. Moreover, to position the sensors at suitable locations, a rig had to be developed.

Another goal is to design the system so that the final product in great extent is applicable to pre-existing Volvo systems and therefore the use of sensors with similar specifications like the ones installed in the cars would be preferable.

If time, resources and technical specifications of the sensors and setup allow it, the sensor systems performance could be improved with a triangulation algorithm for determining objects positions.

1.2 Problem Declaration

Today's reverse assistance systems are becoming very reliable and this causes drivers to put too much trust in them. These systems are still not foolproof, and this overly trust generates a lot of unnecessary parking damage to the cars and surroundings and expenses for the insurance companies, and in some cases personal injuries. Therefore the main task is to develop and implement an automatic anti-collision system which will aid the driver in avoiding accidents while reversing, without intervening in situations which the driver would consider unnecessary. A big problem is that people could become comfortable and rely too much on the system to always stop the car before hitting anything, thus it must be designed so that it at great extent prevents both intentional and unintentional misuse, without compromising the safety benefits of the system. These goals will be accomplished by programming robust software code, good algorithms and through the development of an external rig of sensors. The rig has to be built since the signal from the reverse assist sensor signal is unavailable due to limitations in the car to MicroAutoBox interface.

Additional Issues

In addition to solving the main problem, it would also be preferable if this task could be solved by using a sensor configuration with similar specifications as the one already present in the reverse assistance system.

1.3 Limitations

Normally reversing speeds are low, often in the range of 1 to 5 *km/h*, but to make sure the system has a large operating margin it would be desirable if it was working up to a maximal velocity of 15 *km/h* or 4,5 *m/s*. Since traction varies widely from season to season in some regions, and friction between the road and the tires of the car is a very difficult condition to predict, this will not be taken into account in the project. The road conditions for testing will be specified to dry tarmac. Two factors that will not impact much, and thus will not be considered, are air pressure and wind speed. Ultrasonic waves translates at different speeds depending on air pressure and wind speed affect the propagation of the waves. For this application, the precision of the distance and resolution are not critical enough for these variables to be taken into account.

During this project the financial assets available are limited to 5000 SEK. This sum will be used to ascertain components and material not already available to the project from Chalmers.

2 Methodology

In this chapter three aspects of the methodology are presented: the hardware and software used, how the extensive testing was conducted and lastly how the virtual simulations were performed.

2.1 Presentation of Hardware

This section addresses the hardware used in the project. Some of the hardware is domain-specific why it is considered desirable to present it explicitly.

To make programming, testing and development easy, a rig onto which the sensors attached to was constructed. The sensors were then connected to an Arduino micro controller unit which processes the sensor signals and sends the data via UDP protocol, a quick data communication protocol, to the car's installed rapid prototyping computer, the MicroAutoBox.

Since the budget for the project was limited to 5000 SEK, the price of the electrical components has been a crucial factor in choosing hardware. In the budget there must also be room for all fittings, rigging and cables.

2.1.1 Hardware Used in the Project

Volvo S60

The Volvo is a standard issue S60 that have been fitted with an external control box for tapping into the cars control system. No physical modifications have been done to any part of the car. A picture of the Volvo S60 is presented in figure 1.



Figure 1: Volvo s60

MicroAutoBox II

In order to enable easy access to the car's main computer through the CAN bus network, an expansion box adapted for automotive environments has been connected and is reachable from the luggage compartment. With a computer fitted with MATLAB, a custom Simulink model can be created and uploaded to the MicroAutoBox in order to access and control the cars different systems. Figure 2 display a picture of the MicroAutoBox.



Figure 2: MicroAutoBox II

Data Logger Racing Technologies D11 Mk2

This data logger uses a GPS and accelerometer to measure acceleration in different directions. The data is saved to a computer from where it later can be exported to MATLAB for analysis. It is possible to get a lot of different data from the logger, for example, lateral and longitudinal acceleration, speed and position. This is used to model the cars braking dynamics more reliably.

Arduino Uno rev. 3

The Arduino Uno, displayed in figure 3, is a micro controller with several analog and digital inputs. It is easily programmed by using the Arduinos own programming environment and language similar to ordinary C code.

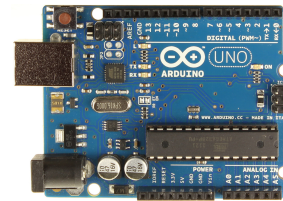


Figure 3: Arduino Uno Rev. 3

Arduino Ethernet Shield

For communication between the Arduino, which administers the sensors, and the MicroAutoBox an Ethernet interface is used. This method was used and stated as convenient in earlier projects with the car with Arduino units connected to the MicroAutoBox. The Arduino Uno has no Ethernet connection in its basic configuration so it will have to be supplemented with an Arduino Ethernet shield to allow this connection. A picture of the Arduino Ethernet shield is presented in figure 4.

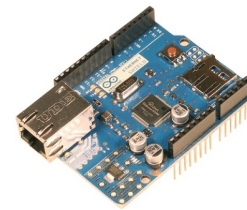


Figure 4: Arduino ethernet shield

Maxbotix LV-Maxsonar-EZ0 MB1000

The LV-Maxsonar-EZ0 ultrasonic sensor was found to possess many of the characteristics desired in this project. They have a wide field of vision compared to many others considered, as well as a high sample rate at 20 Hz and a, for the purpose, adequate resolution. Figure 5 displays a picture of the Maxsonar sensor.



Figure 5: LV MaxSonar-EZ0

Low Pass Filter

The electric systems in cars are often exposed to fluctuations due to the many appliances connected to it. The sensors and the Arduino will then be affected by this since it is interconnected to the cars powering and CAN-BUS system. This noise will make the sensors' supply voltage fluctuate which in turn will result in incorrect measurements from the sensors since their output voltage is proportional to the supply voltage (see Section 2.3.1). To reduce the fluctuations of the supply voltage a first order low pass filter was constructed. Information of appropriate cut of frequency was gained through recommendations from the manufacturer Maxbotix (Bonar, 2012a).

Because of the limited time and component availability, an alternative filter than the one recommended was constructed. The cut off frequency is still the same as the recommended filter though. The schematics of the filter can be seen in Figure 6 below.

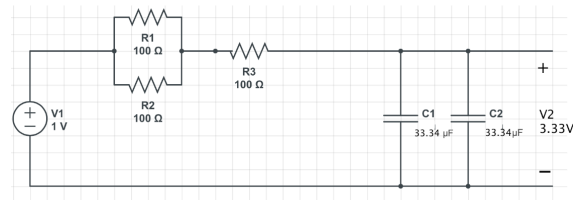


Figure 6: Low pass filter

Cable

Interconnections between the sensors and Arduino are made with a 8 pin (solid single corded) TP cable. Using this type of cable ease the handling as using separate cables for all sensor connections would result in tangling and difficulties distinguishing different wires. It is also crucial to use a cable with sufficiently low inner resistance to provide the Arduino unit with a correct analog voltage signal. A picture of the TP cable is presented in figure 7.



Figure 7: Cable

Sensor Rig

During the time of this project other projects were being executed on the car simultaneously, so to allow for easy assembly and to make testing convenient, a rig was constructed so that the sensors could be easily attached to and detached from the car in order to perform tests when the car was unavailable. The rig can be seen in Figure 8 below. This test rig was constructed with a perforated u-profile rail mounted on a bicycle transportation mount fitted to the cars tow bar. The perforation allows sensors to be easily arranged in a variety of positions on the rail to make different sensor configurations possible without having to resort to major mechanical alterations. The construction and 3D-printing of swivel mounts made it possible to redirect the sensors direction of detection. While testing the sensors, the need for a cone for protecting the sensor from wind, vibration and consequently constrict the sensing area was revealed so this function was integrated into the sensor mountings on the swivel mounts. How the rig and mountings were created is explained further in Section 2.4.1

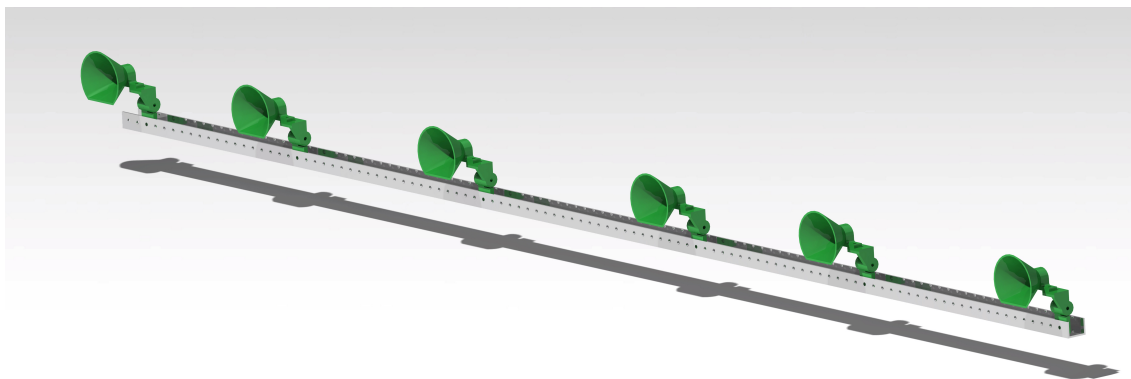


Figure 8: The sensor rig

2.1.2 Hardware Setup in the Car

In the luggage compartment in the car there is a test platform mounted where system components are bolted to a plywood board in order to make everything easily accessible for repairs and connecting of cables. On the center console of the car an emergency stop button is installed which shuts down the MicroAutoBox immediately if there is something wrong with the program or if the car does not react as intended.

2.2 Presentation of software

In addition to the domain specific hardware, domain specific software was also used as well as some more common programs. These are presented below.

2.2.1 Software Used in the Project

MathWorks Simulink

Simulink is a modeling software with a graphical block interface. Each block represent a mathematical function usually with variable parameters. The software is able to simulate and analyze multivariate domain dynamic systems. These blocks represents the foundation of the program for the car. To connect the program to the car an interface made for the specific car is used. The interface that is used in this project is supplied by Volvo and through this the cruise control with an implemented auto brake can be controlled. The block will from now on be referred to as the GCDC block, a name supplied by Volvo. This block and the rest of the blocks developed in this project can be observed in Appendix 6.8.

dSpace - ControlDesk

To control the car from the MicroAutoBox a software called ControlDesk is used. With ControlDesk it is possible to upload a Simulink file to the MicroAutoBox, which then can be executed and tested to see if it works as intended. From ControlDesk every input and output signal is controllable or writable. The software is also able record and save data from the car while tests are carried out. These recordings can later be analyzed in MATLAB.

Race Technology Analysing Software

Racing Technology is a manufacturer of instruments, sensors and data loggers which are aimed at analysing vehicle dynamics with great precision. The analyzing software package includes a collection of programs for connecting the data logger to a computer for real time measurements and for analyzing data. The Race Technology software has a built-in function to export collected data to MATLAB which simplifies evaluation.

Arduino

Arduino have developed their own program for interfacing with and programming the Arduino micro controller. This runs a variant of the c programming language.

CATIA V5 r20

CATIA V5 r20 is a commercial and industrially used CAD program. This is used to design and produce model parts of sensor mounts for rapid prototyping through 3D-printing.

2.3 Using Ultrasonic Sensors

Through the project different sensors and ways of configuring these have been evaluated through testing, to substitute the stock sensors installed in the test car by Volvo. This was done using predetermined testing protocols, which are attached in Appendix 6.2 and 6.3

2.3.1 Maxbotix LV-Maxsonar-EZ0 MB1000

As the built in sensors of the test vehicle were not easy accessible one important step of the project was to find sensors as alike these as possible to test the system with. This process mainly consisted of estimating the Volvo stock sensors characteristics and finding substitutes for these. After scanning the market it was found that the MB1000 fitted the description, budget and availability best.

Sensors' Sensing Width

Volvo has, as part of the standard equipment, installed four sensors in the rear bumper for the proximity warning system. These have such a wide cone of detection that four of them is enough to satisfactory cover the width of the car. After comparing the data sheets and testing the angle of detection for the MB1000 sensors, it was approximated to 48 degrees. (See Appendix 6.3 for further knowledge of how the tests were performed.) The sensors actually had differences in the width of the "cone" at different distances, but for easy handling, and to be able to simplify calculations and reasoning, the value of 48 degrees was set, which gives a little margin and means that the sensing volume can be estimated to be a symmetrical cone.

Resolution

The MaxBotix MB1000 has a resolution of an inch while in analog read mode. Any higher resolution than this would result in a higher price tag per sensor and unnecessarily fluctuating values due to vibration from test car and sudden changes in temperature and wind. The sensors from MaxBotix are able to send readings by analog and digital with pulse width modulation, PWM. Trying to read with PWM would result in higher resolution and precision values but it would make the simultaneous reading of the sensors hard-some, which is essential because getting

values sequentially would result in too slow update of proximity data compared to the cars dynamics and ability to decelerate. **Refresh Rate**

The limitations set for the project states that the system should work up to speeds of 15 km/h (approx $4,2 \text{ m/s}$). Since there are delays in the systems for a variety of reasons such as the time it takes for the brakes to apply after a signal is sent from the MicroAutoBox, there is a need to get readings at a high enough rate so that the distance the car travels between two readings is not so great that a collision might occur. For example the refresh rate of 10 Hz would result in the traveled distance of 0.42 m between readings. The MB1000 has a refresh rate of 20 Hz , which would equal a traveled distance of 21 cm/reading at 15 km/h . As this refresh rate of 20 Hz , was about the fastest discovered in ultrasonic sensors available on market this was found most probable to be equal to the Volvo sensors.

Shapes and Materials

Because it is important that the emergency braking system should be able to work in a common car environment, effort went into tests checking how effectively the sensors works with different materials and shapes. In short, the material of the objects determine at what distance they can be detected, given that they have a surface perpendicular to the sensor. This is because different materials absorb different amounts of sound.

Voltage Scaling MaxSonar Sensors

The sensors' output is a continuous voltage which varies with the distance to an object. The voltage is measured by the Arduino which is then using a scaling factor to convert it to a distance. The scale factor functions are described in Equation 1 and 2 below. (Bonar, 2012b)

$$V_i = \frac{V_{supp}}{512} \quad (1)$$

$$R_i = \frac{V_m}{V_i} \quad (2)$$

where V_{supp} is the supply voltage, V_i is voltage per inch which gives the sensors' resolution, V_m is output voltage from the sensors and R_i is the distance in inches.

2.3.2 Method for Reading of Sensor Values

By deliberately triggering the sensors to initiate their work cycle in synchronisation, a cycle rate of 20 Hz is achieved. This is accomplished by connecting all triggering

pins of the sensors to the same digital output on the Arduino. For a more in depth explanation and evaluation of the different triggering methods see Appendix 6.1 and for the Arduino code see Appendix 6.6. Figure 9 describes how the simultaneous measurement method is implemented with the Arduino.

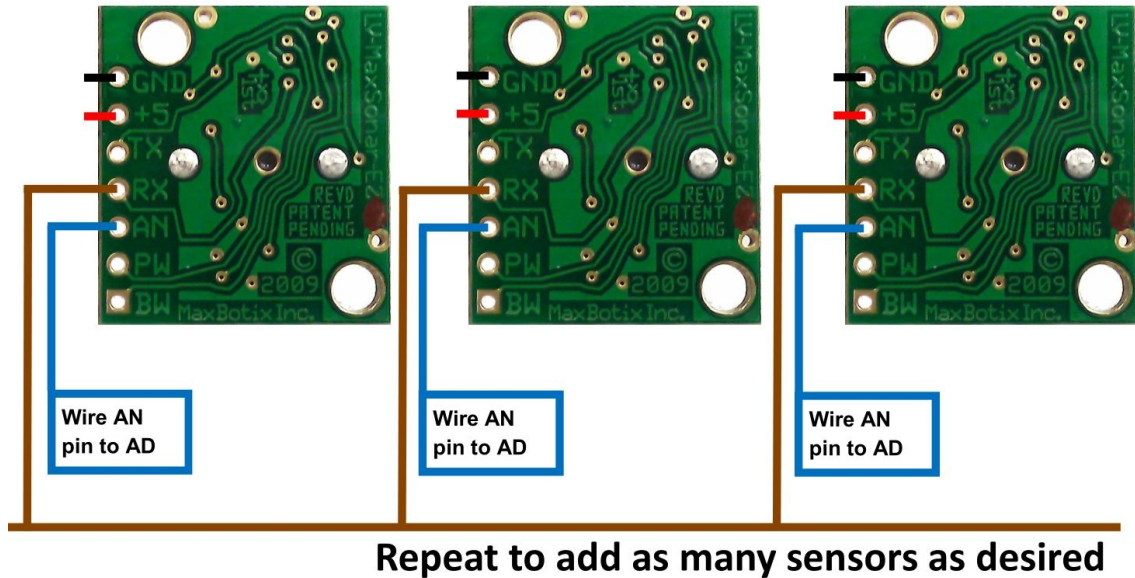


Figure 9: To enable simultaneous readings, all triggering pins are connected to the same digital Arduino output.

2.3.3 Low Pass Filtering DC Power supply for Sensor system

To eliminate DC power fluctuations, a Low Pass filter was connected between the power supply and the sensors. An unsteady current seemed to affect the readings since the Arduino was powered by the cars 12 V system, that varies slightly depending on what in-car systems are running. The filter design and specifications were recommended by MaxBotix for some applications, and verified as functional through testing. It was also discovered that connecting the ground of the Arduino-sensor system to the common ground of the car, got rid of a voltage difference between these. This connection stabilized stochastic fluctuating sensor readings that was found in some of the sensors.

2.4 Rigging Up the Sensors

Since ultrasonic sensors are angle dependent, there was a need for being able to test different angles to fine tune the cluster of sensors. This was accomplished by developing a rig that is able to hold the sensors and that enables easy repositioning of them both vertically and horizontally.

2.4.1 Developing the Sensor Rig

Firstly it is important to point out the purpose of this rig. It is in no sense meant as a prototype for implementation on Volvo cars available for consumers. It is solemnly purposed to be used to accommodate the sensors, and to solve this projects main problem declaration, which summed up is to test different braking strategies for an emergency braking system. This makes the characteristics of the development somewhat different from the development of a typical consumer product which could demand higher standards of sturdiness and usability.

Expected Functionality of the Rig

The demand on this piece of equipment was that it was to be developed with no considerations for anything other than to fulfil the most basic demands put on it in this project. The final design of the rig consisted of a aluminium rail with a cars width in length bolted to a bicycle stand with tow bar mount, which allowed for easy on and off assembly when not conducting tests. For early stage testing the use of sticky tape was sufficient for fastening the sensors to the rail. But after concluding that fluctuating sensor readings could occur due to direction and angle towards the ground a need for easier adjustments of this angle was obvious. therefore the aluminium rail was attached to make it easier to attach the sensors. Tests with the sensors screwed on sheets of metal resulted in lots of amplification of vibrations from the running car. As an alternative the use of 3D-printed mounts was considered. This manufacturing method was easily accessed and fast, and would likely get rid of vibration amplification problem.

The needed functions for the mount was evaluated and hence a model was constructed in CATIA. This was then sent for print and an PLA plastic component was manufactured. The resulting design was a three piece, 3D adjustable mounting, as can be viewed in Figure 10 below.

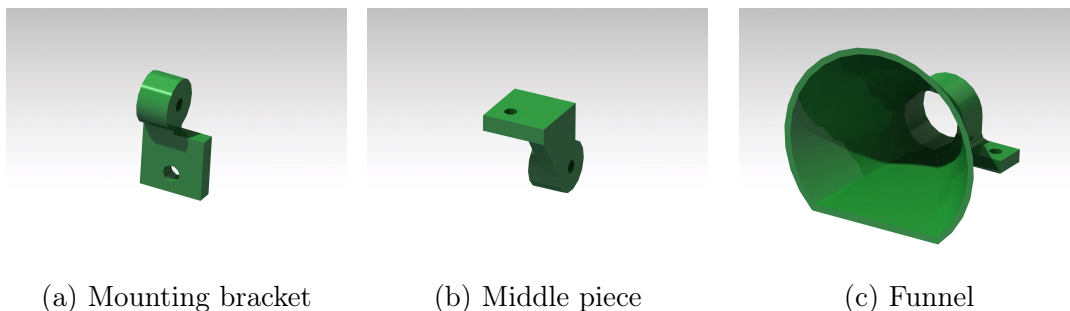


Figure 10: The complete sensor funnel, piece by piece

By placing a cone on an acoustic element, it is possible to alter the characteristics, for example range and width. This was proven to be convenient for controlling the vertical sensing angle, and a reflector was built into the cone for the sensors to limit the sensing area downwards.

2.5 Dynamics of the Brake Process

The stopping distance for a car can be described from general formulas of speed, time and acceleration. The general formula is described in Equation 3 and 4. The formulas for calculating the dynamics of the car are taken from Physics for Scientist and Engineers with Modern Physics (John W. Jewett, 2014).

The braking time can be calculated from the general formula, described by Equation 3. The initial velocity is represented by v_0 and the acceleration is represented by a . The variable v represents the final velocity which in this specific case is equal to zero.

$$v = v_0 + at \quad (3)$$

The distance an object travels during acceleration can be described by Equation 4. v_0 is the initial velocity for the object. t_0 is the time it takes to start the acceleration, namely the time the object moves before it starts to accelerate and t_a is the time the object is affected by acceleration.

$$s = v_0 t_0 + \frac{at_a^2}{2} \quad (4)$$

2.5.1 Limitations in Brake Dynamics

The braking process is highly dynamic and dependent of many external factors and limitations that affect the vehicle's deceleration. Maximum deceleration depends on many factors such as aerodynamics, gradient of surface and velocity as well as traction between the tires and road surface. This mathematical formula does not take any of these external factors in consideration. Instead the vehicle is assumed to have a average deceleration due to braking a_{avg} .

Full acceleration is not obtained directly but increases gradually. The rate of change of the acceleration is in physical terms called jerk and describes the third deviate of position (m/s^3) as shown in Equation 5. In the same way as the acceleration, the jerk is also dependent on several parameters and therefore the vehicle is assumed to have an average jerk, j_{avg} .

$$a_{avg} = \int j_{avg} dt \quad (5)$$

In the brake system, there is a delay from when the brake signal is sent until the vehicle starts to decelerate. This delay is given by the constant t_{del} and during this

time the vehicle travels a distance s_{del} which is formulated in Equation 6.

$$s_{del} = v_{in}j_{avg} \quad (6)$$

2.5.2 Brake Scenarios at Different Initial Velocities

The jerk j_{avg} is active in two different scenarios, in the first the velocity decreases to half the initial velocity v_{half} before maximum deceleration is reached a_{avg} . This means that the jerk is increasing the deceleration from when the braking begins until the moment when half initial velocity is reached and then shifts to decrease the deceleration.

A second scenario appears when the initial velocity is sufficiently great to let the acceleration reach its maximum before the speed have been halved. In this scenario the braking process starts by the jerk increasing the deceleration to maximum level, after which the vehicle continues to slow down with the achieved deceleration until the deceleration decreases to zero and the vehicle is standing still.

To calculate the stopping distance it is assumed that average deceleration a_{avg} , jerk j_{avg} and time delay j_{avg} are given constants. The jerk is also assumed to be equally great during increase and decrease of acceleration, but with different signs. The value of these constants have been generated by tests and can be found in the results chapter.

Scenario 1, Full Brake Acceleration Not Reached

In this scenario, the stopping distance s_{stop} can be divided in three components. The distance the vehicle travels during time delay in the system s_{del} , the distance the car travels during increasing accelerations $s_{jerk_{beg}}$ and the distance the vehicle travels during decreasing acceleration $s_{jerk_{end}}$.

In this scenario the acceleration is built up during time $t_{v_{half}}$ which by modification of Equation 7 is given in Equation 8. The equation describes the time it takes to decrease the initial velocity v_{in} by half but it also describes the time to decrease the speed from half initial velocity to standing still.

$$\frac{-v_{in}}{2} = \frac{j_{avg}t_{v_{half}}^2}{2} \quad (7)$$

$$t_{v_{half}} = \sqrt{\frac{-v_{in}}{j_{avg}}} \quad (8)$$

Equation 3 can be modified to describe the actual velocity ($v_{jerk_{beg}}$) at the moment deceleration reaches its maximum level. The formula is given in Equation 9.

$$v_{jerk_{beg}} = v_{in} + \frac{j_{avg}t_{v_{half}}^2}{2} \quad (9)$$

By integrating the velocity given in Equation 9 the distance $s_{jerk_{beg}}$ is given in Equation 10. At the starting point of this formula the vehicle travels at initial velocity.

$$s_{jerk_{beg}} = \int_0^{t_{v_{half}}} v_{jerk_{beg}} dt = v_{in}t_{v_{half}} + \frac{j_{avg}t_{v_{half}}^3}{6} \quad (10)$$

The distance the vehicle travels during the decreasing deceleration is given by Equation 11.

$$s_{jerk_{end}} = v_{jerk_{beg}}t_{v_{half}} + \frac{a_{v_{half}}t_{v_{half}}^2}{2} - \frac{j_{avg}t_{v_{half}}^3}{6} \quad (11)$$

Finally the total stopping distance can be described as Equation 12 with the three components s_{del} , $s_{jerk_{beg}}$ and $s_{jerk_{end}}$ from Equation 6, 10 and 11.

$$s_{stop} = s_{del} + s_{jerk_{beg}} + s_{jerk_{end}} \quad (12)$$

Scenario 2, Full Brake Acceleration Reached

The stopping distance s_{stop} in this scenario can be divided in four components. The distance the vehicle travels during time delay in the system s_{del} , the distance the car travels during accelerations increasing $s_{jerk_{beg}}$, the distance the car travels during a constant deceleration s_{cont} and the distance the vehicle travels during acceleration decreasing $s_{jerk_{end}}$.

The jerk is in processes for the time $t_{a_{avg}}$ which is given by Equation 13. This equation gives the time taken to build up from zero to average acceleration as well as to decrease the acceleration.

$$t_{a_{avg}} = \frac{a_{avg}}{j_{avg}} \quad (13)$$

When full brake acceleration a_{avg} is reached the vehicle decelerates continuously during the time t_{cont} as given by Equation 14.

$$t_{cont} = -\left(v_{in} + \frac{j_{avg}t_{a_{avg}}^2}{a_{avg}}\right) \quad (14)$$

combining of Equation 3 and 13 gives the velocity at the moment when full brake acceleration is reached, which can be described as Equation 15.

$$v_{jerk_{beg}} = v_{in} + \frac{j_{avg} t_{avg}^2}{2} \quad (15)$$

The velocity at the moment when acceleration starts decreasing from a_{avg} to zero is given in Equation 16 as a combination of Equation 14 and 15.

$$v_{cont} = v_{jerk_{beg}} + a_{avg} t_{cont} \quad (16)$$

By integrating the velocity $v_{jerk_{beg}}$ given in the Equation 15 the distance $s_{jerk_{beg}}$ can be described as Equation 17. The formula gives the distance the car travels during the jerk is in process to increase the acceleration.

$$s_{jerk_{beg}} = \int_0^{t_{avg}} v_{jerk_{beg}} dt = v_{in} t_{avg} + \frac{j_{avg} t_{avg}^3}{6} \quad (17)$$

By integrating the velocity v_{cont} given in equation 16 the distance s_{cont} can be expressed as in Equation 18.

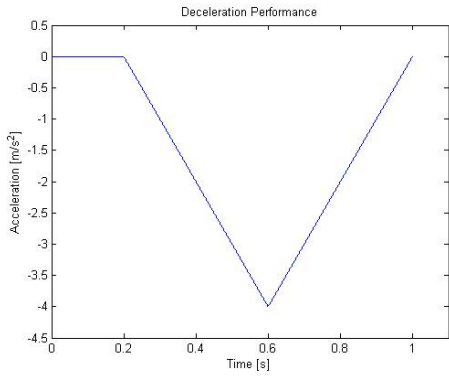
$$s_{cont} = \int_0^{t_{avg}} v_{cont} dt = v_{jerk_{beg}} t_{cont} + \frac{a_{avg} t_{cont}^2}{2} \quad (18)$$

The last part of stopping distance is given by Equation 19. The equation describes the distance the vehicle travels during acceleration decreasing.

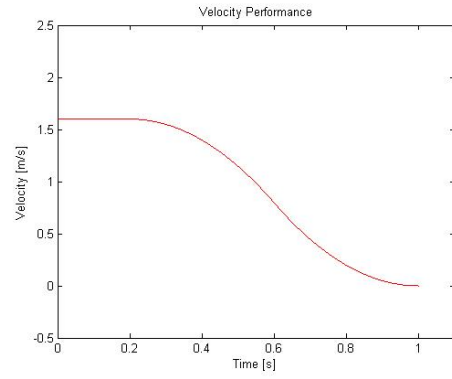
$$s_{jerk_{end}} = v_{cont} t_{avg} + \frac{a_{avg} t_{avg}^2}{2} - \frac{j_{avg} t_{avg}^3}{6} \quad (19)$$

The total stopping distance is formulated in Equation 20 as a sum of s_{del} from Equation 6, $s_{jerk_{beg}}$ from 17, s_{cont} from 18 and $s_{jerk_{end}}$ from Equation 19.

$$s_{stop} = s_{del} + s_{jerk_{beg}} + s_{cont} + s_{jerk_{end}} \quad (20)$$

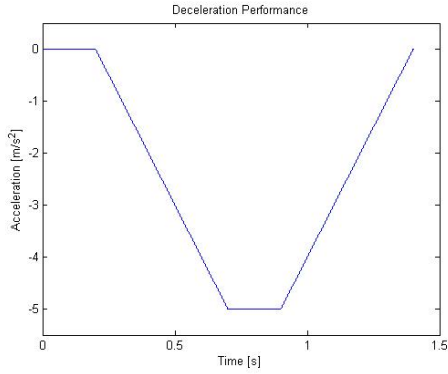


(a) Deceleration

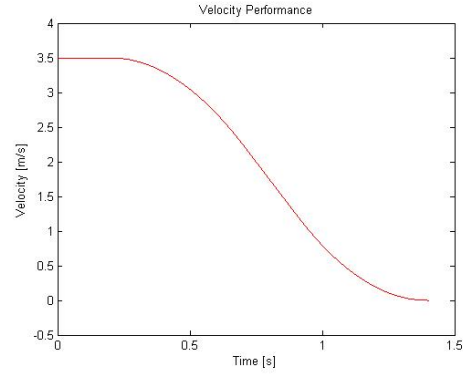


(b) Velocity

Figure 11: This is an example of the deceleration process in scenario 1. The initial velocity is set to $1,6 \text{ m/s}$ and the jerk to 10 m/s^3 and average acceleration to 6 m/s which is not reached.



(a) Deceleration



(b) Velocity

Figure 12: This is an example of the deceleration process in scenario 2. The initial velocity is set to $3,5 \text{ m/s}$ and the jerk to 10 m/s^3 and average acceleration to 5 m/s which is reached for $0,2 \text{ seconds}$.

2.5.3 Complete Theoretical Model for Simulation

To be able to verify and test different braking strategies a graphic simulation model was created and imported to the model based simulation program Simulink. Based on the cars' braking dynamics the model was modified to be able to determine if the distance to an object is less than the braking distance for the car at that speed. The calculations are based on the stopping distance s_{stop} from Equation 20 in case of brake scenario 2 with the addition of a safety margin (s_{safety}) to prevent a collision. This is presented in Equation 21.

$$s_{stop} = s_{del} + s_{jerk_{beg}} + s_{cont} + s_{jerk_{end}} + s_{safety} \quad (21)$$

This yields a limit to where last possible brake point is at different velocities and this is graphically represented in Figure 13 below.

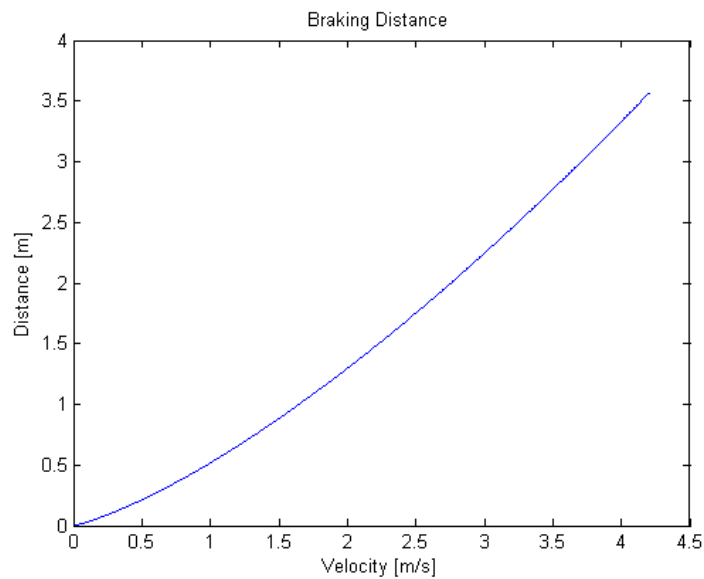


Figure 13: Last possible brake point if the car should be at a standstill at s_{safety}

By adjusting the required acceleration level and safety margin the model can simulate different brake strategies.

Data such as distance to object, speed, acceleration and braking signal were analyzed through so called "scopes", a function in Simulink. These scopes mainly plot data relative to time, but can be configured to whatever the analysis calls for. A visual representation of the car approaching an obstacle based on the simulation was also created for easier identification of the success of the test.

2.6 Data Gathering for Modelling

In order to get a representation of how the car reacts to sudden impulse braking, data gathering in real life conditions had to be done.

2.6.1 Onboard Sensors

A car has a lot of sensors for the ABS, traction control and other safety systems to be able to work. Via the analysis program "Control Desk" these sensors can be accessed and their data can be logged to a file for later analysis. Variables of interest for the purposes of this project was lateral acceleration, braking request and speed of the car. To get repeatability in the results the car was reversed in a straight line at different predetermined speeds. To make sure consistent data was acquired the tests were performed several times at the speeds 5 *km/h*, 10 *km/h* and 15 *km/h* respectively.

The data gathered from the tests were then saved to a struct, which is a multi-level array. These files were then loaded into MATLAB and can from there be plotted for further in depth analysis.

To record an experiment Control Desk has to be prepared in such way that it knows when to start the recording and what variables it is supposed to record. After that it is just a matter of starting the recording with a button, performing the test and stopping the recording and the variables are automatically exported to a MATLAB struct for analysis.

2.6.2 Data Logger

To make sure the data gathered with the cars sensors were correct an external data logger was used. They are more accurate since it uses accelerometers. To be able to compare the tests with the inboard sensors they were made under the same conditions.

The data from the logger was fist saved to a program called Race Technology Analysing Software, where the data could be analysed. To get a good plot over the data that is printable the logged data was exported to a MATLAB matrix.

2.7 Simulink Model Implemented in the Car

The output that is interesting in this project is brakes. The interesting input signals are speed, acceleration over ground and raw data from the ultrasonic sensors. These are the main inputs but there are also additional signals that are going to be used, for example if the driver pushes the brake or acceleration pedal.

The signals on the CAN-bus are hard to interpret without knowledge of how the messages were encoded in the first place. Extensive testing would have had to be conducted in order to find the specific message sequences for each individual input request and that is not reasonable for a project of this small scale. To work around this problem Volvo has constructed a block in Simulink that gives the ability to demand deceleration and a lot of other variables to be used just by numbers. This Simulink block is basically a couple of inputs and outputs that can be controlled and no other access to what goes on "behind the scenes" is possible. This makes it easy to program Simulink code since everything done to brake the car is done automatically. No implementation of ABS or anti spin control has to be done.

The Simulink model in the car uses the equations from Chapter 2.5. These are combined into a block that calculates the critical distance to collision. This block then sends a signal that tells if the distance is critical to a state flow block. State flow is a sequential flow chart programmer which in this case decides the state of the program. Depending on the state the car either prepares to brake or brakes for an obstacle. To apply the brakes on the car, a signal with the required deceleration is sent to the GCDC block which controls the adaptive cruise control. A signal that requires parking brake is also sent to the same block. When the car have come to a

complete stop it waits for the driver to push the brake pedal. At this point the car releases the parking brake. For Simulink model and MATLAB code see Appendix 6.7 and 6.8

2.7.1 Decision Making Process While Reversing Toward a Stationary Object

The current speed of the car is extracted from the MicroAutoBox and is used to calculate the critical distance i.e the distance the car requires to avoid a collision and stop at a chosen distance margin. To determine if the braking system should engage, the critical distance is compared with the distance measurements received by the sensors. If the measured distance is less than the critical distance, the braking system is engaged. The decision process which is implemented in the car is visualized in the flow chart in Figure 14 below.

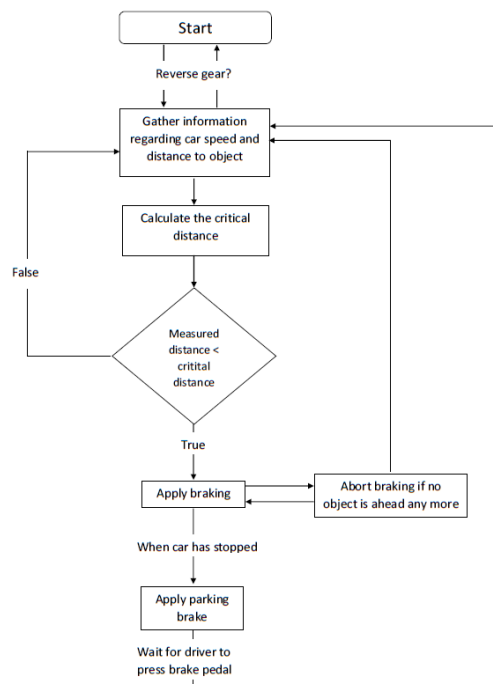


Figure 14: Flow chart of the decision making process

2.8 Brake Strategies

To prevent misuse of the auto braking system, different types of braking patterns were tested. The idea was to try different brake strategies to see which was the best way to prevent misuse and create the safest braking strategy. When tests were carried out it was discovered though, that the interface between the car and Simulink has a very strong amplification which means the the car brakes hard in the beginning and then levels out to the desired value.

2.8.1 Brake Dynamics Test

To determine the brake parameters for use with the calculations, tests were made to collect data with use of both the car inboard sensors and the data logger. The tests were carried out by supplying a requested deceleration step of 10 m/s^2 and log the cars deceleration. These tests can be acknowledged as the braking systems step response. To perform the tests a light Simulink model were made to control the requested deceleration signal manually.

2.8.2 Brake Strategies Test

Primarily, another light Simulink model had to be produced to perform these tests. In the same way as the "Brake Dynamics Test" the Simulink models are used to set requested deceleration manually. As the access to the braking system was limited, a way of altering deceleration stats that had to be investigated was the use of one Simulink model that produce deceleration trough pulse width modulation.

Other brake strategies that are interesting to test is to increase the deceleration with a ramp resulting in higher comfort as the car will decelerate more smoothly. Another strategy would be to increase the deceleration exponentially to smoothing the braking even more.

In a final product a variety of different brake strategies may be used. Depending on for example different velocities, different brake patterns may prove more effective or necessary for consistent performance. One strategy that probably always should be implemented is the ability to brake hard and late because this both prevents misuse and accidents with objects suddenly moving in behind the car.

2.8.3 Setup of Braking Tests

In order to get consistency in the tests a controlled environment is important. To control all parameters, the experiments have to be conducted multiple times at the same speed and at the same angle to the obstacle. Then when the system is stable in these conditions other scenarios can be tested to see if it works as intended.

The friction between the tarmac and tyres varies from day to day, and the changes are the most obvious when the asphalt is wet. Since there is no good way of measuring friction that could be implemented on a car, this parameter has to be left out from our calculations. Therefore the scenario that the system is tuned to handle is dry tarmac. Although, the velocities that are likely for a reversing car are lower than the velocities were any difference in stopping distance, due to wet conditions, is noticeable. The car stops almost instantly when the emergency brake control signal is set since the brakes are good enough to stop the car with out locking up the wheels.

For the purpose of reversing the car onto an object that would not damage the car in case the system failed a cardboard box was used as obstacle. This gave a

good enough signature for the sensors to detect it and did not damage the car when collisions occurred.

3 Results

3.1 Developed Hardware

Throughout the project the need for some components and functions led to the development of prototypes to solve these. The hardware structure as a whole is presented below and the developing of this is presented under method. The rig as a whole, with tow bar mounted aluminium rail, wiring between sensors, Arduino, Ethernet shield and low pass filter, can conveniently be dismantled from the car when not in use.

Summary schematic diagrams of the system and a couple of its key components can be found below. Figure 15 shows fundamentals how the wiring of the MicroAutoBox and some of the cars systems are made. Figure 16 shows the addition of this projects developed system and the essential wiring of this.

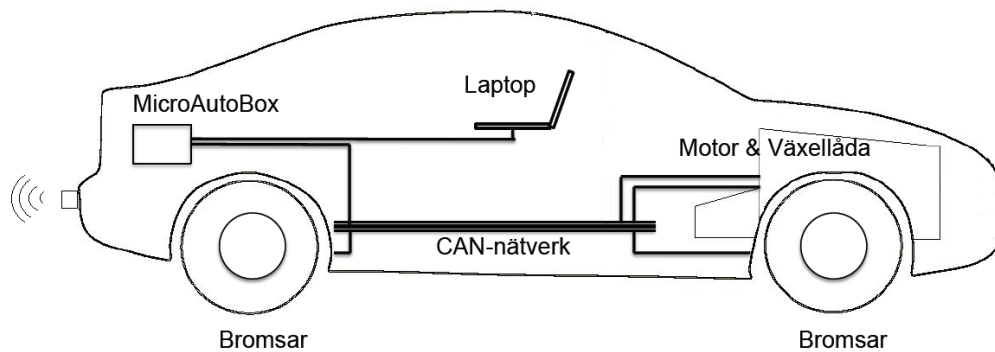


Figure 15: Schematics of essential parts of the whole car.

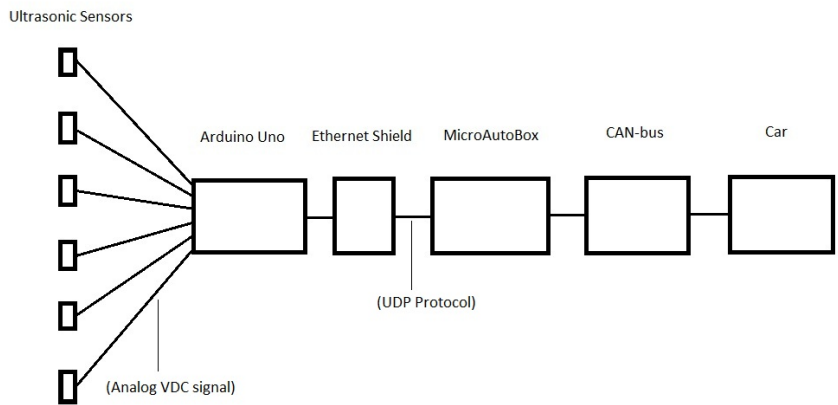


Figure 16: Schematics of essential parts of the developed system

In Figure 15 the sensor mounts are assembled with sensors wired on the rig. For easy recognition the cables running from the sensors were color marked.

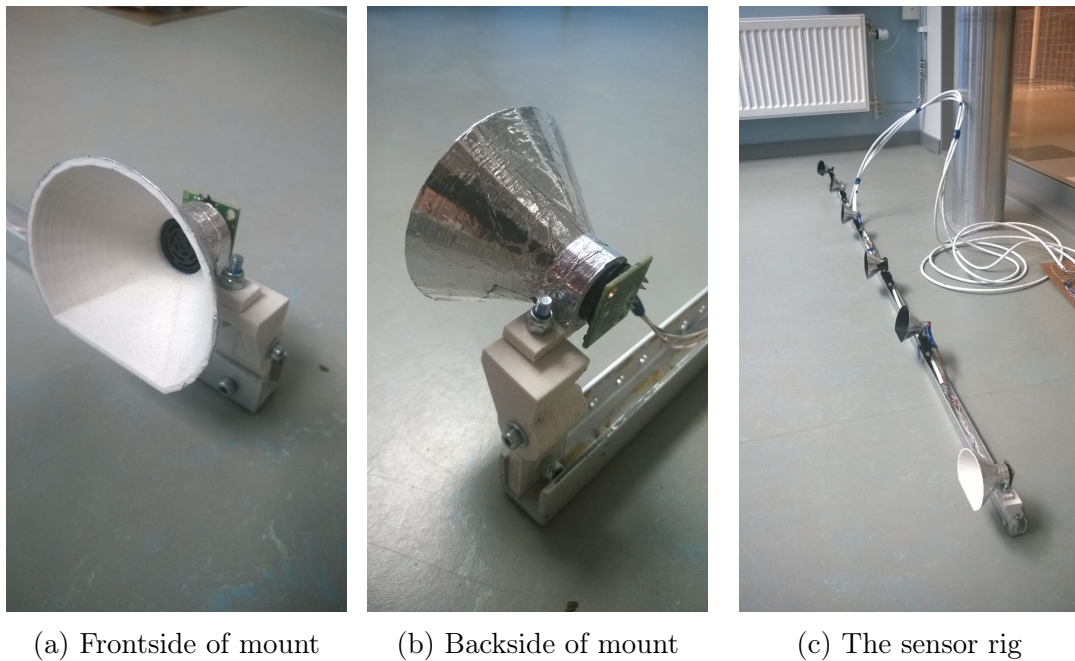


Figure 17: The sensor mount and the whole rig

The breadboard for connecting the sensors to the Arduino unit is visualised in figure 18. The open design made connecting, testing and troubleshooting of the wiring easy and comprehensible.

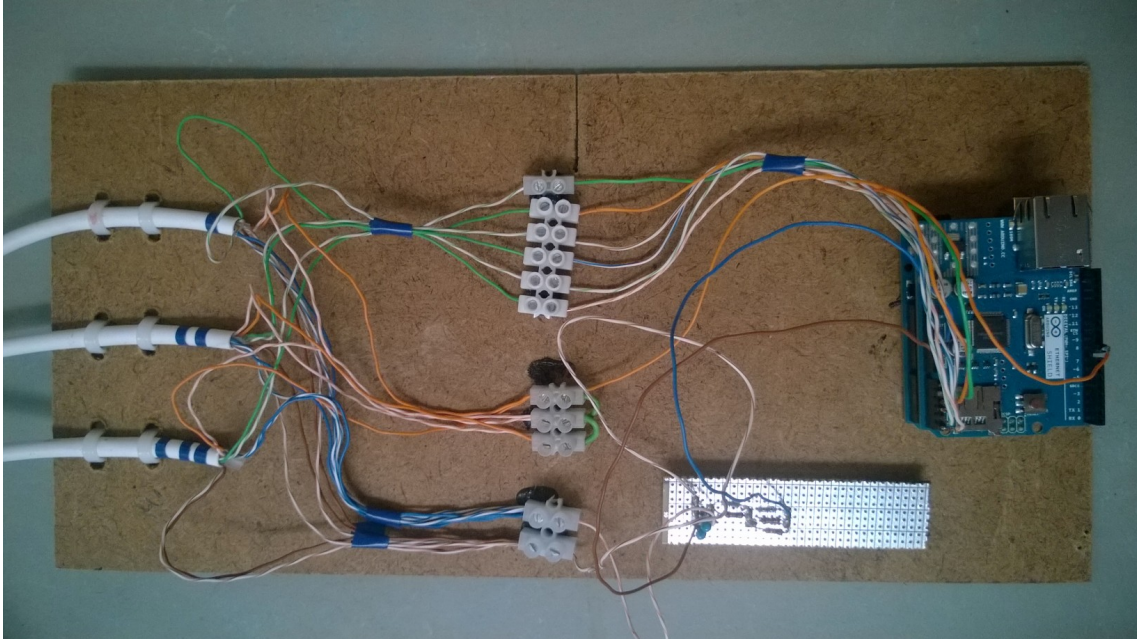


Figure 18: Connections between the MicroAutoBox, Arduino unit and sensors are made on this breadboard.

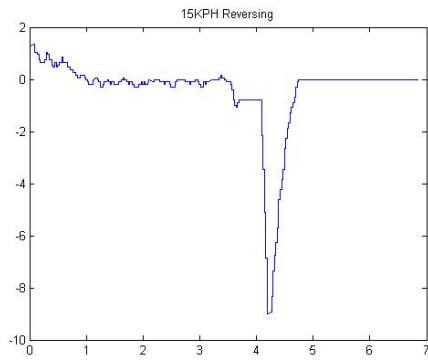
When all parts of the rig was printed, mounted and all interconnections was securely connected, it was functioning properly and according to plan. The 3D-printed sensor mounts made it easy and convenient to test different sensing angles and settings.

3.2 Determining of Brake Parameters

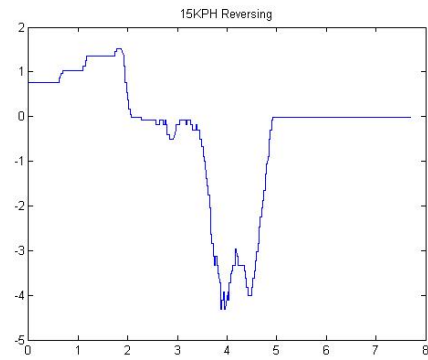
The tests were made on a dry plane asphalt surface with good traction. In each test at least 5 samples of random selected velocities between 3 - 10 km/h were recorded.

3.2.1 Results From the Cars On-board Sensors

Tests with readings taken from the cars onboard sensors gave ambiguous results, in form of two distinct variants of braking dynamics. Figure 19 displays the data gathered from the cars' onboard sensors when the same deceleration was demanded. As can be seen in the graph, there is no continuity in the braking performance. The steep peak in Figure 19a is the result of the wheels locking and therefore does not represent the deceleration of the car. To gain accurate measurements on the cars deceleration, an external measuring device from Race Technologies was used.



(a) One variant of deceleration



(b) Another variant of deceleration

Figure 19: This is example of the two distinct variants of deceleration logged by the car onboard sensors.

3.2.2 Results From the Data Logger

The data logging of the cars braking dynamics was made with Race Technologies Data Logger 1 MK2. One of the data logging samples can be seen in Figure 20. Based on the collected data, the average jerk was calculated to 15 m/s^3 and the average acceleration to 10 m/s^2 .

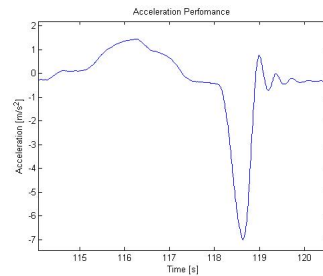


Figure 20: Example of one of the samples made by the data logger

3.3 Results of Simulation

The simulation provided information about how the car would react to a brake impulse step. From the simulation a graph of critical distance, and a graph of the deceleration and how the speed decreases were obtained. The graph over distance in Figure 21 shows that the car stops at a constant distance, the safety margin, regardless of the speed. The simulated car starts to brake when the graph starts to behave non-linear at around 0,2 *m*.

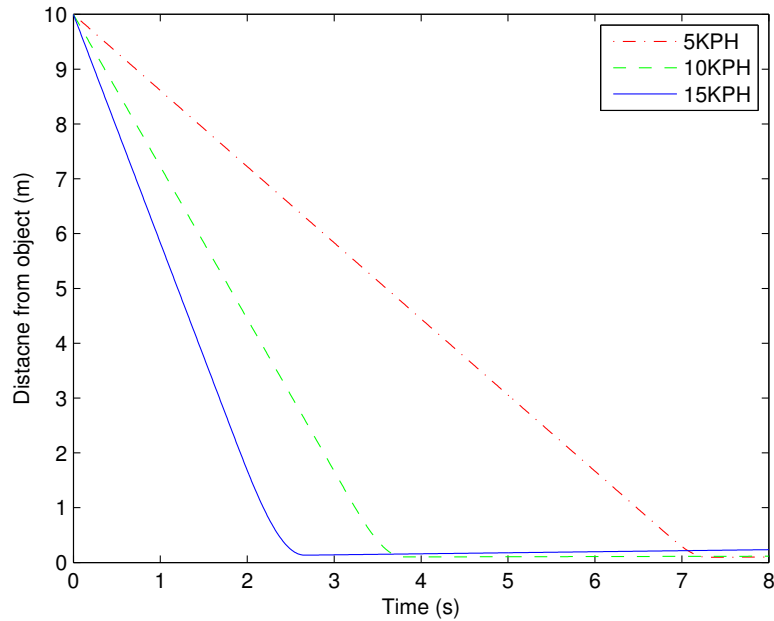


Figure 21: Simulation of distance between car and object. Notice the non-linear transitions of the different lines at around 0,2 *m*

3.4 Brake Strategy Tests

A total of three different brake strategies were tested in the project. Of these three strategies only one worked sufficiently well for further development. The tests results are presented below.

Pulse Width Modulated Brake Signal

The pulse width modulation strategy was tested with three different carrier frequencies (2, 5 and 10 *Hz*) as braking signal. In all tests where pulse width modulation was used, the car was requested to decelerate with 10 m/s^2 . The duty cycle was set to 50 percent of the period time. When the carrier frequency was set to 10 *Hz*, the deceleration was smooth and reached a maximum of 0.3 m/s^2 . The deceleration performance with a carrier frequency of 10 *Hz* is shown in Figure 22a. When 5 *Hz*

was used, the deceleration reached a maximum of 0.4 m/s^2 as shown in Figure 22b. the test of 2Hz frequency resulted as shown in figure 22c

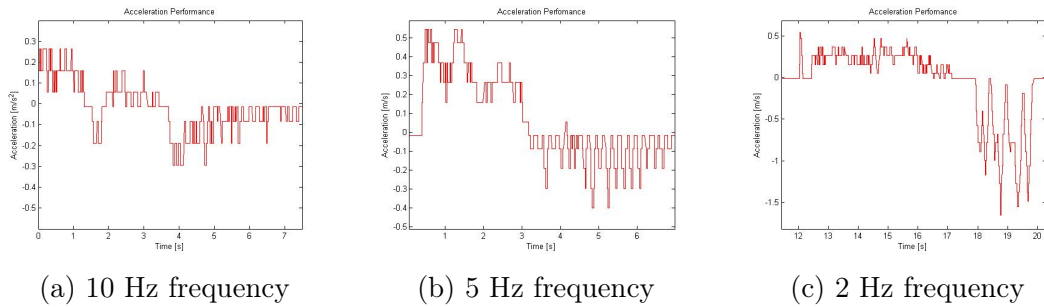


Figure 22: The figures display the three different frequencies tested with use of pulse width modulation.

Braking hard and late

These tests were made to determine how the car brakes with a requested deceleration of 10 m/s^2 . Results from different tests are found in Figure 20 which was logged by the cars' onboard sensors and in Figure 19b which was logged by the Race Technologies data logger device.

Braking soft and early

In these tests the car was requested to decelerate with 2 m/s^2 . Figure 23 display the acceleration result from one of these tests. All tests resulted in a hard deceleration above 2 m/s^2 in the beginning of the brake process before it smoothed out at the desired level.

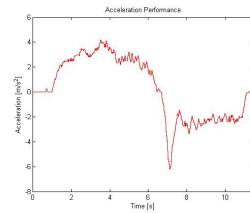


Figure 23: Example of requested deceleration of 2 m/s^2 .

3.5 Real World Testing of the System

When the car reverses perpendicular to an obstacle it brakes and stops within the safety margin and at a constant distance from the obstacle. If the car turns towards an obstacle the sensors work in the same way as the car reverse perpendicular, which means that the car stops at a constant distance. If the driver wants to park between two cars the system does not interfere unnecessarily. The sensors cannot detect low obstacles like a concrete block at close distances, which means that the car will not brake for it. This is a result of the sensor placing.



(a) A harmless obstacle to test the system against

(b) A real life scenario

Figure 24: Testing the system

3.5.1 Collision Avoidance

As described in Chapter 2.8.3, tests were conducted when reversing straight towards a cardboard box at different velocities to see if the car stopped at its predetermined safety margin. As can be seen in Figure 25 below the car reacts as intended.

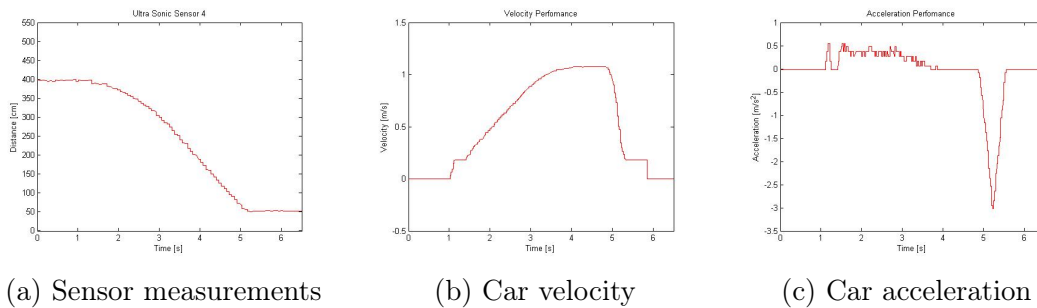


Figure 25: The figures display data gathered when the car was reversed toward an object and applies the auto brake. The step behaviour between 5,2 s and 5,9 s in the graph of car velocity might be explained as a sensor error. A more in depth explanation can be found in Chapter 4.4.

3.5.2 Brake Margin Continuity

Tests were performed to measure the continuity in performance of the auto brake program. The results from the tests can be found in Table 1 below. The reason for the increased safety margin at velocities above 6 km/h is that there is a implemented break point at 6.4 km/h where the safety margin is increased due to make sure there is a margin for error.

Velocity [km/h]	Sensor distance [cm]
4	50
5	52
5	51
5	50
5	46
6	61
7	52
7	54
8	56
8	60
8	54
9	66
9	60
10	61
10	58
10	68
10	63
13	65
13	62
15	66

Table 1: Table of braking results. The right column displays the velocity while the left shows the distance registered by the sensors after the braking procedure when the car is standing still.

3.6 Sensor Stability With and Without Low Pass Filter

Figure 26 displays the results from the low pass filter test. When the low pass filter is used, the fluctuations of sensor measurements is less frequent. The distance scales in the graphs differ from each other due to a lower power input using the filter but not adjusting the scaling factor accordingly. The supply voltage is 5 V without the low pass filter and approximately 3.4 V with the filter. However, correct scaling factor is not significant for interpreting this result.

Additional graphs of sensor fluctuations can be found in Appendix 6.5.

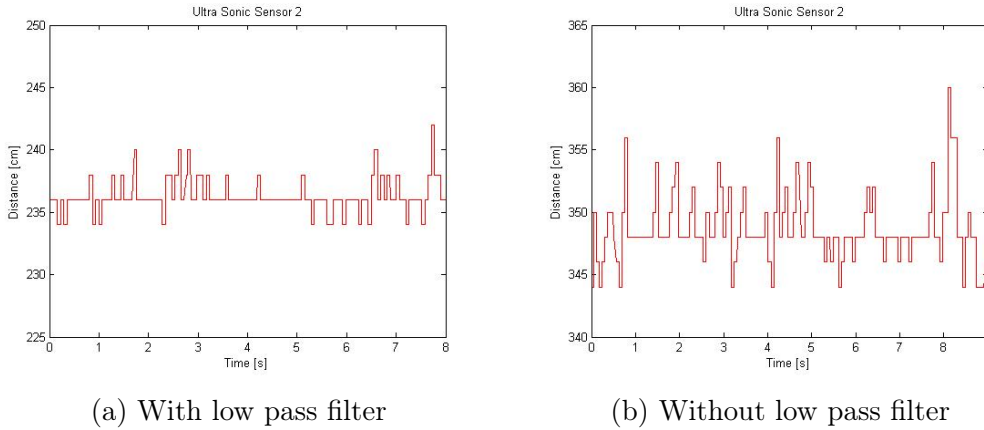


Figure 26: Sensor fluctuations with and without the use of a low pass filter

3.7 Sensor Triggering Method Used

The Maxbotix sensors have as stated several different ways of triggering, each with its advantages and contradictions. Results of tests on other triggering-reading methods can be found in Appendix 6.1.1. **Simultaneous Sensor Triggering and**

Measurement

The method where all sensors are triggered to begin sending and receiving at the same time provided accurate and stable measurements when six sensors were used. To gather measurements from all of them takes approximately 50ms. For convenient programming of the reading Arduino unit it is considered necessary to use analog output signal from the sensors, as it would be troublesome to make the Arduino read the pulse width of the sensors output simultaneous.

3.8 Achieving Autonomous Emergency Braking

With the current setup, with the sensors on the rig directed straight backwards, the car stops by itself when coming to close to an obstacle. It does this with a very

uncomfortable jerk, which would deter most people from relying on the system to do the braking for them while reversing.

4 Discussion

4.1 Choice of Hardware

According to time plan, the hardware would have been selected and implemented into the system quite early on in the project process. All the components and much of the software is dependent of each other, so it was therefore important to have a good idea of how everything was supposed to work with everything else before settling on the specific components to use. **Sensor Location**

Much of the reasoning about how to design the emergency braking system has been revolving around the current reverse system that has four sensors directed backwards. The ultrasonic sensors attached to cars have a relatively long range of detection, but more importantly very wide. This allows for four sensors to cover the entire vehicles width, which means that most of the points in space behind the car is only covered by one sensor.

Since one of the projects extended objectives was to determine the position of an object, and because of the reasoning above, it is not enough with only four sensors to determine where an object is located. Four sensors provide accurate measurements for how far away from a sensor an object is, and can only in small areas, where the sensors' fields overlap, detect the object with two sensors simultaneously. This can be seen in Figure 27.

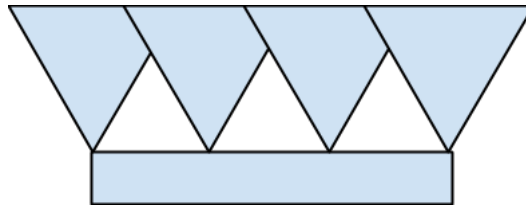


Figure 27: Illustration of current ultrasonic sensor setup. The areas where the detection areas are overlapping is small.

To get more accurate positioning on where an object is located, there are two general options:

1. Many sensors that each covers a narrow area, which is narrow enough to provide a sufficiently accurate resolution of the object. This setup can be seen in Figure 28.
2. Sensors with overlapping areas of detection that makes it possible to calculate where an item is with triangulation. For a more in depth explanation of triangulation principals see Appendix 6.4. This can be accomplished with wider sensing area sensors, or different sensor configurations. This option can be seen in Figure 29.

Option 1 requires many sensors. This can be a problem if/when the system is to be implemented, as it can be expensive and aesthetically unattractive with so many sensors in the bumper of the car.

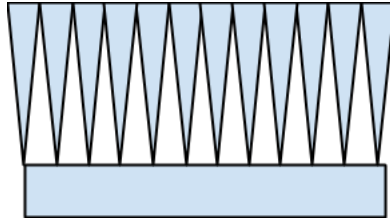
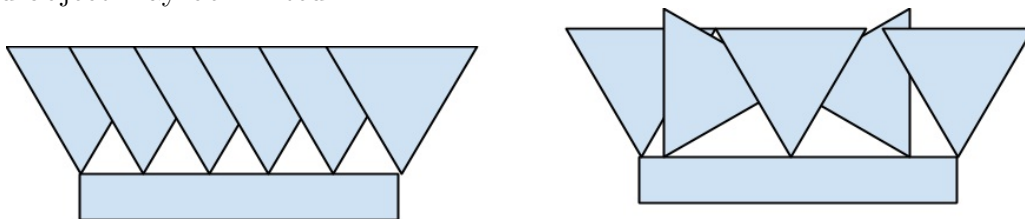


Figure 28: By using many narrow sensors, adequate lateral resolution may be acquired by knowing which sensor is detecting an object.

Option 2 requires fewer sensors than option 1, and various possible locations of the sensors results in different configuration possibilities. Depending on where in the field of detection the object is located the accuracy of the distance between the car and object may be limited.



(a) Using wide sensors, detecting an object with multiple sensors makes it possible to determine the lateral position via calculations.

(b) Another possible alternative sensor setup.

Figure 29: Different sensor setups

For a complete system to be more commercially viable with the demands for aesthetics and low costs it was decided that it is option 2 which will be the pursued one.

There is however not one "best" way of arranging the sensors, so some thought was given to finding the optimal one for this project. In Figure 29 two options are displayed, and Figure 30 displays one that was found to be very promising because of the complete coverage of the area behind the car. Both configurations in Figure 29 lack coverage close to the car, which means the system would be relying solemnly on calculations in certain circumstances, such as reversing close up against a pole.

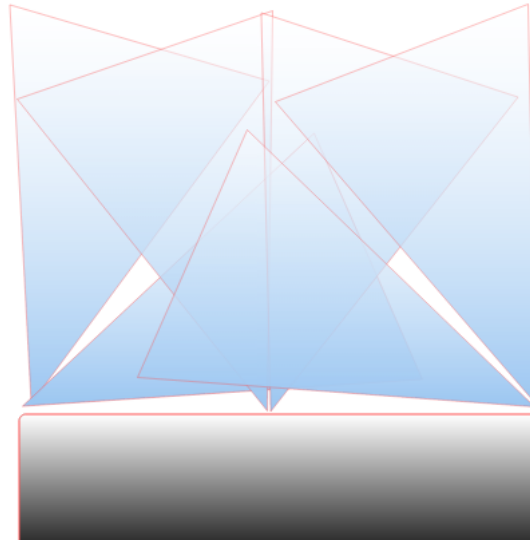


Figure 30: Using the sensors' in pairs makes it possible to have basically the whole area behind the car covered. A configuration like this would require calculations to establish a distance to an obstacle though.

4.1.1 Recognition of Objects

It was realized that it would be beneficial to have more knowledge of a detected object. There is ways to analyze the objects with frequency interpretation of the returning signal. The sensors now being used have built-in control electronics that can not be modified in a simple manner. These work great at detecting objects, but it is not possible obtain the raw signal from the ultrasonic element to independently analyze this. Therefore priority has been to develop a prototype system that works, in the sense that the car will stop for obstacles.

An alternate way of developing the sensor rig would therefore be to use sensors without built in control electronics and process the raw signal separately to gain better control of the sensors characteristics, configuration and usability. This was however considered to far from the original project description and too time consuming to implement.

4.1.2 Avoiding Collision of a Moving Object Crossing the Cars Trajectory

With the developed system, the car will brake when an object is located within a certain distance from the car, and this distance varies depending on the cars speed. However, this strategy does not take into consideration whether the object is moving and can therefore not determine if the object will pass the cars trajectory in such manner that braking no longer is required. By frequently updating the objects position, relative to the cars, and compare it with earlier positions it would be possible to determine the objects current velocity. By knowing its current velocity,

the objects trajectory could be calculated. By comparing the objects trajectory with the cars, decisions whether collision would occur or not could be made. Moreover, for situations when a collision will occur at current velocity of the car and object, different braking strategies could be implemented to ensure that the object passes the cars trajectory. For example, the car could reduce its speed so that an object passes the trajectory of the car before a collision occurs.

4.1.3 Avoiding Collision With Object While the Car is Turning

In normal traffic situations a driver makes turns while reversing. With the aid of the MicroAutoBox it is possible to extract the value of the cars wheel angle. With the knowledge of the wheel angle, the trajectory of a turning car can be calculated. The trajectory can then be matched with the objects in the same manner as described above. For example, consider the situation visualised in Figure 31 where a car is reversing into a parking slot where neighbouring slots are taken by other cars. During the parking the car will come close to the other cars. Without the knowledge of the wheel angle and therefore the cars trajectory, the autonomous braking system could consider this as a collision course, resulting in an unnecessary braking manoeuvre. To solve this a triangulating system as described above or sensors that can be turned inwards in same manner as the wheel or have a electronically controllable sensing angle could be used.

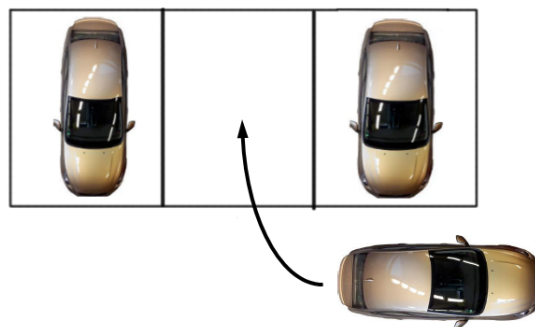


Figure 31: Car revering into a parking slot

4.2 Simulations

The results from the simulations gave a good indication that the car could handle the different scenarios it would be exposed to. A realistic simulation of the cars characteristics is crucial for the development of the final model. Being able to have an indication of how the car will behave in the real world makes the development of the first algorithms much more accurate. Unfortunately there is some difference between the simulation and the reality. Based on the data that has been collected there is possibly a jerk to stop acceleration which for now isn't implemented. The parameters of maximum acceleration and jerk is not in this moment clearly defined as a secondary fault of the problem of gathering data.

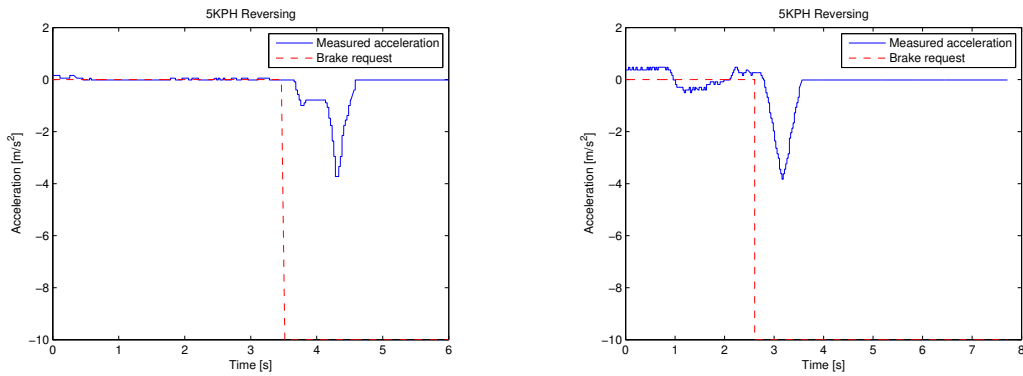
4.3 Reasons for Few Braking Strategies

Since it is only possible to demand a deceleration value in m/s^2 no knowledge of how the signal is translated into output to the hydraulic brake actuator is known. Their pre-defined regulation sequence has a very large amplification and because of this the car, no matter how low you set the deceleration demand, always responds with a violent brake at first and then smoothens out towards the set point value. This has made it hard to test different brake strategies. Experiments were made with pulse width modulation (PWM) of the signal, but since there is a buildup time, or jerk, until the brakes operate at full power these trials only resulted in that the car had a very jerky brake course. The only chance at testing other strategies would have been if Volvo had provided a new interface with other variables to control. If, for example, the signal to the hydraulic pump that control the automatic braking could be set directly strategies that would have had a lot smoother brake course could have been tested. It would maybe have been preferable if the power of the brakes would have increased the closer to a object you reversed, giving the driver a lot better sense and control over the progression. Say for example that you had to reverse a bit into a bush to be able to turn the car around. Today the system could mistake the bush for some other object not letting the driver reverse any further than the safety margin is set to.

Another strategy could have been that the braking sequence would start a lot earlier with lower brake power to give a less aggressive event. The problem with this could be that it becomes to comfortable and the driver puts to much trust in the system which is not preferable from a safety standpoint. It is believed that the setup today, with a late and hard braking, is the best in terms of not intervening with the driving experience. Most people that have driven the car with the system as it is set up today have thought they reversed too closely to the obstacle to avoid a collision when in fact they have had a couple of decimeters left until collision. That confirms that the system works as intended in that it is only activated when the driver clearly has missed an obstacle. This with a combination of a intensifying beeping warning, much like the one installed today, would considerably reduce the number of unnecessary collisions that occur today.

4.4 Gathering Data Using the Cars Onboard Sensors

The data gathered with the onboard sensors of the car is not completely accurate because it uses wheel tics to calculate acceleration. This means that if the wheels lock up the the wheel tics will give a constant value and the calculated acceleration gives a constant value which can be seen in Figure 32a. This only occurs if the wheels lock up, when it does not, the graph looks like Figure 32b. Since it is different every time it is also impossible to determine acceleration and jerk. This is why the data logger was used. Since the data logger uses accelerometers to calculate the acceleration it still works even if the wheels lock up.



(a) Deceleration step response at 5 km/h with disturbance. The vertical line at 3,5 s represent the signal to activate the brakes. Instead of the expected linear acceleration, a plateau is registered at 4 s. This may be caused by the ABS intervening due to issues with traction.

(b) Deceleration step response at 5 km/h . The vertical line at 2,5 s represent the signal to activate the brakes. The linear acceleration at 3 s is the anticipated behavior of the brakes.

Figure 32: Step responses

4.5 Interface Between the Sensor System And The MicroAutoBox

Initially the plan was to program the Arduino to handle all data processing i.e. recording all the readings from the sensors, calculating where an object is located and taking the decision of whether to brake or not. This ended up not being the case mainly because of two reasons. The first being that there was concern about however the Arduino was able to make all the calculations quick enough, and by this meaning still being able to run all the sensors at their maximum refresh rate and make the calculations at the same time. The second reason was that a possible application in the real world would most probably only take readings from the sensors and do all the processing in a central computer.

Regardless of if the Arduino had been programmed to make the calculations or not, communication between the Arduino and the MicroAutoBox is necessary anyway. The communication works through UDP protocol since that is the standard supported protocol in the MicroAutoBox.

4.6 Further Development of Object Positioning and Tracking

One issue of great debate has been how to angle the sensors in relation to each other and the surroundings. The sensors detection width is a big advantage in many ways, since it allows for the use of fewer sensors to cover the area behind the car. However, since the sensors range is rotationally symmetric, it provides one small

problem to overcome. If the sensor is positioned having its center line parallel to the ground, it will detect small object lying on the ground, and in cases of rough road surfaces, perhaps even the ground itself. This could be the cause of unwanted activation of the system. This is avoided by angling the sensor upwards, or placing a deflector on the sensor. By doing this it is possible that low objects that could damage the car would remain undetected. This poses with a decision to be made, to determine the optimal balance between not registering the ground and still detecting possibly harmful obstacles. The optimal upwards tilting angle will be the one that at maximal registering range stops detecting a curb.

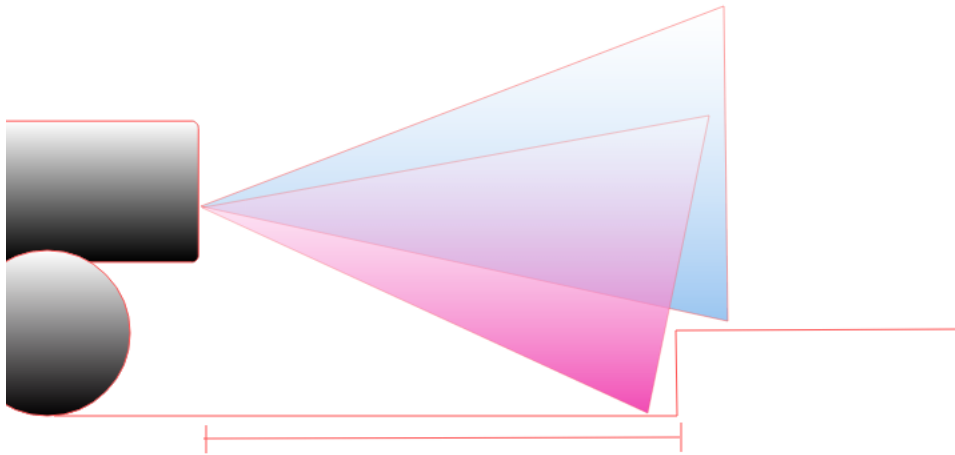


Figure 33: With the sensors aimed too low, the sensors' will register a harmless object close to the ground like a curb.

In some circumstances, for example when the car is moving slowly and the braking occurs late, it is possible for low objects, that still have enough height to damage the car can go out of the sensors range before the braking begins, and therefore gets hit. To avoid this two possible solutions have been found. The first is to extrapolate its position and the second is to place sensors lower down on the car as well as the normal bumper height with the lower sensors trailing the ground. The one major problem with low sensors is where to put them on a physical car.

To further develop the autonomous braking system another method of object detection could possibly be developed. By presenting the information of a objects position in lateral and longitudinal coordinates, additional manoeuvres to avoid objects could be implemented. To gain the lateral and longitudinal position at least two sensors need to be able to detect the object, putting high demands on sensor reliability and sensor positioning. In this project focus has been to develop the braking strategies with a sensor system similar to the one installed today, but an alternate way could be to develop a system using other sensor techniques, such as radar, IR, laser etc. Likely a combination of these techniques would be preferable,

but this is not investigated in this project.

Since the sensors transmit ultrasonic pulses at the same time there might be some minor error in the distance measurements due to the fact that the sensors transmit at the same frequency and therefore cannot determine from which sensor an ultrasonic pulse originated. The situations described above can occur when an object is detected by two sensors. In situations where a sent pulse is lost and one from another sensor is registered this might lead to incorrect readings.

4.7 The Autonomous Braking System From an Environmental Point of View

Today a lot of car parts are being scrapped due to minor damages obtained from low speed collisions. In a unpublished report from the insurance company Folksam it is estimated that repair costs from reversing crashes accounts for 21% of the total repair cost of all their analyzed vehicle crashes (Ydenius and Rizzi, 2014). This in combination with that 25% of their analyzed data set contained reversing accidents (Ydenius and Rizzi, 2014) tells us that this is a big problem area with much room for improvement. Probably the number of crashes could be drastically reduced if our current system would be installed in cars and in doing so reducing the environmental impact by a great measure.

Because of the sensors already installed in many cars, integrating the autonomous braking system in a car will not add to the environmental impact significantly since beside the sensors, the rest of the system works purely on software which wont require any additional components as it can run on the on board computer.

5 Conclusion

All in all, the developed system is working. Sensors, programmed Arduino unit, MicroAutoBox with Simulink model and connections between all these are running smoothly at the end of the project time. The car is executing the expected braking sequence when it is necessary in situations stated as normal in a traffic environment.

It is although concluded that there is room for further development and fine tuning of some aspects of the project. One area of obvious room for improvement is the more precise positioning of objects. As this was not one of the projects main goals this alternate function was pretty much developed but not implemented as the time was running out. The matter of whether the system can effectively prevent misuse is somewhat difficult to measure in discrete terms, but after testing the system for some time, it can be concluded that a more steep braking sequence is more uncomfortable and therefore should in the greatest extent possible, without being dangerous for the driver, prevent the misuse. It can be concluded that the parameters is somewhat arbitrary and, consequently for perfection, could be the matter of some more subjective customer panel testing and surveying.

There is also some obvious limits with the ultrasonic sensors used in this project. The systems precision would probably benefit from the use of a combination of sensor types and techniques, like radar, laser and IR, so that emergency braking is not mistakenly executed when not necessary.

It was established far into the project that access to some of the systems essential for implementing different brake strategies was somewhat limited. This clearly have had impact on the possibilities of creating the perfectly smooth and completely satisfying emergency braking system. This limitation was due to confidentiality preferences of Volvo's system and the cost of resources it would take to develop a new and more suitable interface.

References

- Tom Bonar. Troubleshooting guide @ONLINE, October 2012a. URL <http://www.maxbotix.com/articles/035.htm>.
- Tom Bonar. Finding distance using analog voltage @ONLINE, October 2012b. URL <http://www.maxbotix.com/articles/032.htm>.
- Raymond A. Serway John W. Jewett. *Physics for Scientist and Engineers with Modern Physics*. Physical Sciences: Mary Finch and Physics and Astronomy: Charlie Hartford, 20 Channel Center Street Boston, MA 02210 USA, physics for scientists and engineers with modern physics, ninth edition edition, 2014.
- Trafikverket. Nollvisionen @ONLINE, August 2012. URL <http://www.trafikverket.se/Privat/Trafiksakerhet/Vart-trafiksakerhetsarbete/Trafiksakerhetsmal/Nollvisionen/>.
- Volvo. Vision 2020 @ONLINE, September 2010. URL http://www.volvocars.com/intl/top/about_volvo/corporate/volvo-sustainability/safety/pages/vision-2020.aspx.
- A Ydenius and M Rizzi. Backing crashes with passenger cars - effectiveness of ultrasonic sensors. Unpublished paper from insurance company Folksam, 2014.

6 Appendix

6.1 Different Sensor Triggering Methods

There are a variety of different ways to obtain sensor values

Free Run Method

To gain the desired sample rate and accuracy from the sensors, different measurement strategies were tested. The first test was with a single sensor, using the analog output and letting the sensor run freely. Free run is an operation which allows the sensors to make continual measurements and data delivery to the computer. The free run operation is dependent on the sensors internal clock which implies that if multiple sensors are to be used, the chance for interference due to not being perfectly synchronized is high. If the sensors internal clock are not synchronised the sensors will, in relation to each other, take measurements at different times, resulting in inaccurate readings due to interference. This method, with the use of one sensor, were mainly used for testing the sensors detection area e.i., determine the angle and distance which an object could be detected. The free run method was also tested with pulse width as data delivery method from the sensors. The length of each pulse is scaled to distance with the Arduino.

The next step in the process of choosing a valid method for range measurements was to try this method with two sensors to see if interference would occur. The sensor interference were studied, first in a normal situation where an object were placed centred and one meter behind the two sensors and then in an extreme situation, where the both sensors were placed at a distance of one meter from each other and directed toward each other.

Daisy Chain Method

A method that eliminate the effects of interference, when multiple sensors are used, is by placing the sensors in a so called Daisy chain. In a Daisy chain only one sensor is allowed to take a range measurement at the time. When a sensor has completed its measurement it tells the next sensor in the chain to measure distance by applying a voltage to the Tx gate, which is connected to the Rx gate on the next sensor in the chain. Figure 34 visualise a chain with 3 sensors that is connected through a Daisy chain.

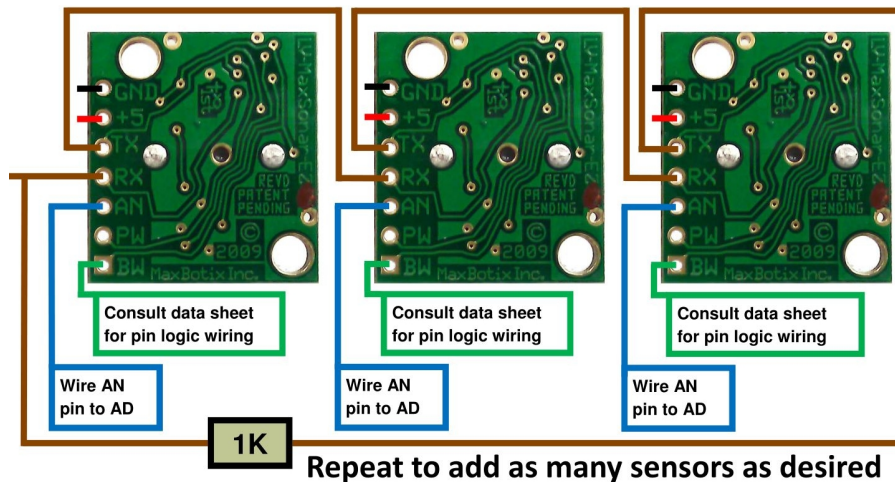


Figure 34: Daisy Chain. The figure display how the sensors are connected.

This method works for a large number of sensors, provide accurate measurements and does not require the sensors to be synchronised. However, since only one sensor is allowed to take a measurement at the time, each taking approximately 50ms , gathering measurements from all sensors take to time. This strategy was tested with the use of two sensors in the same manner as the normal situation as in the free run method were used.

6.1.1 Results of Triggering Methods

Below are the results of the two methods not used in the project.

Free Run Method

For one sensor this method provides accurate distance measurements and can do this at a sample rate of maximum 20 Hz . When two sensors are active at the same time in close proximity of each other the sensors go out of synchronisation with each other after some time, which results in interference which in turn results in stochastically varying measurements.

Daisy Chain Method

This trigger method provides accurate and stable measurements when two sensors are used. As the sensors complete measurements sequential, a cycle of reading all the sensors take approximately the number n sensors times the cycle time of one sensor (approximately 50 ms). This slow process makes this method of triggering unsuitable for this application, as the time between readings affect the reaction time and therefor safety margins.

6.2 Sensor testing

Width and Height Test

It was believed that it was possible that the width/height of the volume of detection varied depending on where in the volume an object is located, and it was something that had to be taken into account when positioning the sensors on the rig.

Width Test

For testing this the approach was to place the test object at a physically measured distance from the sensor and bring it from directly in front of the sensor to where it could no longer be detected. The angle between the perpendicular line straight from the sensor and a line drawn between the sensor and the object was recorded.

Precision Test

By holding an object on a physically measured distance from the sensor and then reading the computer measured distances a difference was obtained which could be used to calibrate the final system.

The approach will be to place the test object:

- Directly in front of the sensor
- On the boundaries of the sensor's angular sensing range
 - horizontal
 - vertically

within the sensor's range.

Testing of a Single Sensor

The following reasoning circulates around Arduino sensors, but the principles apply to all ultrasonic sensors. This early testing made clear what aspects of testing that generated important results instead of large amounts of worthless data with occasional significant numbers.

Before the test, it was believed that the Arduino sensors would have specifications similar to those mounted on the Volvo car, which uses four sensor units to cover the entire vehicle's width and has an estimated reliable range of about 2 *m*. It soon turned out that Arduino's ultrasonic sensors had not at all the same sensing width and length, and pre-test fiddling with a sweater sleeve revealed that a sensor had a width and height of 12 degrees and a range of about 1 *m*.

The sensor range was largely the result of two material properties: ultrasound absorption and size of the surface perpendicular to the sound wave. This meant that the hard, flat material such as sheet metal could be detected at a relatively long distance, in a, in the context, broad cone, but only when the plate was carefully angled towards the receiver. Beads of metal gave no reading at all until they were placed very close to the sensor, and then only those of larger diameter, because of the surface that reflected sound back to the receiver was very small.

Instead of felt lined bead sweater sleeves were used, which showed to be equivalent in terms of ultrasound absorption, and probably a better idea since clothing is what will be detected in real life as well. It became clear that a certain size was needed before any reflexes could be detected. This size was approximately a cycloid with a diameter of 5 *cm*. Since it is a cycloid, it is detectable no matter how it is orientated and therefore gives an echo wherever it is located. A plate for example would make angling it correct important to give reliable readings

The Arduino sensors gave stable readings up to about 1 *m* with a width of ± 6 degrees. These are the specifications that will be used to assess the sensors, because it is in this area a pedestrian could be discovered.

Distance [<i>cm</i>]	Width Left [<i>cm</i>]	Width Right [<i>cm</i>]
30	4	3
50	5	5
70	6	7
100	11	10
120	11	12
150	17	-

For this test, it appeared that for Arduino's own sensors to be used a great number of units will be required to cover the car's width, and the sensors' range may still be limited. This was not yet possible to say with certainty, it could depend on how the Arduino sensors could be controlled and if provided with output signals from several sensors simultaneously could result in stronger echoes and thus receiving a stronger echo. A stronger echo would mean greater range and ability to detect objects.

Testing of Two Parallel Sensors

With the results of the test with a sensor in fresh memory it was now only interesting to see if the two sensors in any way would interfere with each other, and hopefully increase their scope and width thereof.

By connecting the two sensors pointing in the same direction at a distance of 10 *cm* apart the only thing examined was if and how the two sensors interacted in any way. As the measuring object, the same sweater sleeve from the one sensor tests were used. The tests were conducted by directly in front approaching the sensors from a distance and noting at what distance detection happened. To test for interaction, the sleeve was brought in a semicircle with center right between the two sensors from approximately 45 degrees from the central line into the middle of the ultrasonic field, while looking for irregularities in the readings.

After finishing these experiments, it was determined that the range was the same as for a single sensor, and no interference between the two sensors could be detected.

Each individual Arduino sensor's width also appeared not to be affected by multiple sensors mounted next to each other.

What this means is that it is not possible to increase the scope or width by connecting multiple Arduino sensors next to each other and in that way send stronger audio

signals.

Early Testing Plans for Ultrasonic Sensors

There was an extensive test plan for the Arduino sensor and a serious ambition to go through with it, but when it became clear that there was basically two factors affecting the performance of a sensor it was abandoned to save time. Instead, the objects used for testing was exclusively a sweater sleeve and a metal pole. Below is the table of shapes and materials that were to be tested and motivation for the different objects.

Shape	Diameter [mm]	Covering
Bead of steel	3	-
Bead of steel	10	-
Bead of steel	25	-
Bead of steel	50	-
Bead of steel	3	Covered with blanket
Bead of steel	10	Lined in felt
Bead of steel	25	Lined in felt
Bead of steel	50	Lined in felt
Aluminum plate	-	-
Square rod of steel	-	-
Square rod of wood	-	-

The dimensions of the beads were chosen to give a wide representation of various possible obstacles. Steel reflects ultrasound well, which was thought to represent the hard objects such as poles, walls and other cars, while the felt absorbs sound waves and acts as an analog for example clothes.

A Square rod is desired to see if any reflexes are obtained if the rod is angled so that there are no sides perpendicular to the sound wave.

The plate was chosen to give a strong reflection to test maximum length and width of the signal.

Technical Problems That Were Worked Around

What this section is all about is the problems that were going to have to be sold if a different path had been chosen. The reason this is left in the report is that these problems will most likely occur if any of the suggested further developments is pursued

1. Develop means of dealing with asymmetric sensor placements on the rig
2. Gaps in the sensor's detection area

Possible Solutions

1. This step can be solved mathematically, by feeding the sensor data into the Arduino or auto box for processing. Which device is selected to make the calculations depends on how fast the Arduino can work. It is desirable to keep the whole subsystem which manages the sensors and handles the signals

separate from the MicroAutoBox for development purposes. By having the sensor rig and signal processing separate from the MicroAutoBox, other systems can be used for development than those found in the car, which increases its flexibility when access to the car is not always possible. If it turns out that the rate at which readings from the sensors are adversely affected by the calculations performed, it may be necessary to handle the signal processing in other ways, for example by using the Arduino to control sensors and one to handle the calculations.

2. The braking strategies that were being developed had the intention to stop the car close to the object to avoid the system triggering unnecessarily, an object might pass through areas where no sensors monitor. This mainly applies when the object is close to the car. Two methods of dealing with this are:

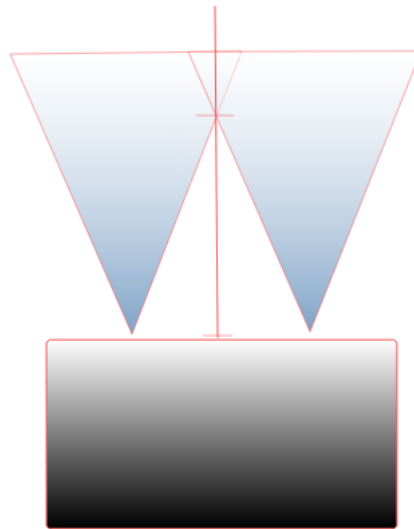


Figure 35: With certain sensor configurations it is possible for objects to enter gaps in the sensing area, meaning they now are "invisible".

- (a) Cover all blind spots with sensors by arranging them in asymmetrical ways, see figure 30. The challenge with this approach is to program the algorithms determining where the object is, which probably will require that sensors be directed along the car instead of away from it.
- (b) Base the approximate position of the object on calculations. As an object passes through the sensor's detection area, the Arduino registers the sensors readings and calculates the objects location. These locations can be integrated, and the object's to the car relative speed and direction can be obtained. By utilizing this and the car's built-in sensors such as position sensors (wheelticks) and speedometer an objects route can be extrapolated. Based on this, appropriate brake strategy can then be performed.

6.3 Test Protocols and Comments to Those

Width test of Maxbotix LV-Maxsonar-EZ0 MB1000

Distance [<i>cm</i>]	Width Left [<i>cm</i>]	Width Right [<i>cm</i>]
100	47	45
200	90	92

Calculating an angle based on the table above gives a width of ± 24 degrees. The sensors are symmetrical, which means this angle applies to the height as well.

Test of 1		All data in cm									
Range	Straight ahead	Far	Near								
	Sweater sleeve	303	16								
	Metal pipe	501	16								
	Bumper	Depends on angle									
Range	outer boundry	Far	Near								
	Sweater sleeve	300	16								
	Metal pipe	495	16	Range is not affected by angle							
	Bumper	-	16								
Precision	Distance from	20	50	100	Good enough?						
	Sweater sleeve	21	51	101	yep						
	Metal pipe	21	48	101	yep						
	Bumper	18	51	99	yep						
Width	Distance from	5	10	20	30	40	50	75	00.00	00.00	150
	Sweater sleeve	-	-	-	-	-	-	-	-	-	-
	Metal pipe	-	-	-	-	-	-	-	-	-	-
	Bumper	-	-	-	-	-	-	-	-	-	-
	See separate test. Width is the same no matter the material as long as an echo is recieved										
Test of 2 sensors											
Range	Straight ahead	Far	Near								
	Sweater sleeve	300	16								
	Metal pipe	495	16								
	Bumper	See note above									
Range	outer boundry	Far	Near								
	Sweater sleeve	298	16								
	Metal pipe	495	16								
	Bumper	See note above									
Precision	Distance from	20	50	100	Good enough?						
	Sweater sleeve	18	51	101	yep						
	Metal pipe	21	48	99	yep						
	Bumper	21	51	99	yep						

Width		0,05	0,1	0,2	0,3	0,4	0,5	0,75	01.00	01.25	1,5
	Sweater sleeve	-	-	-	-	-	-	-	-	-	-
	Metal pipe	-	-	-	-	-	-	-	-	-	-
	Bumper	-	-	-	-	-	-	-	-	-	-
Interference		Sweep at 50			Sweep at 100						
	Sweater sleeve	nope			nope						

The results from the width test, which was not made at the same time as the others can be found in section 6.3.

The conclusions that can be drawn from this test is:

- Since the sensors measures in inches, which is then converted to cm, which makes not hitting the precision marks no big deal.
- In the tests with one and two sensors the sound wave was propagating from more or less one point and because of the irregular shape of the bumper only erratic readings were obtained. With sensors placed further apart, there are more sensors to pick up the echo and would give more stable readings.
- It doesn't matter where an object is placed as long as it produces an echo that can be picked up by the sensor.
- How absorbent a material is is the primary factor affecting from how far it can be detected. Shape is naturally of great importance too since a very reflective, but without any surfaces perpendicular to the sound will be difficult to localise.
- The main reason for testing two sensors simultaneously was to determine whether they would affect each other in any way. This was not observed in any of the tests.

6.4 Principles of Triangulation

To provide longitudinal and lateral position, triangulation is required. Triangulation means that the triangles geometry is utilized to determine a point in the plane's position relative to two other. Figure 36 describe a scenario where an object is detected at a point C in the plane. The length of a and b are gained through measurements from the sensors, placed at the points A and B. The points A and B are placed symmetrically relative the cars central line. By using the law of cosine, described in equation 22, the angle φ and length l can be determined. By knowing the angle φ and length l , the point C can be described by x and y coordinates. The mathematical steps are provided by equations 22– 28

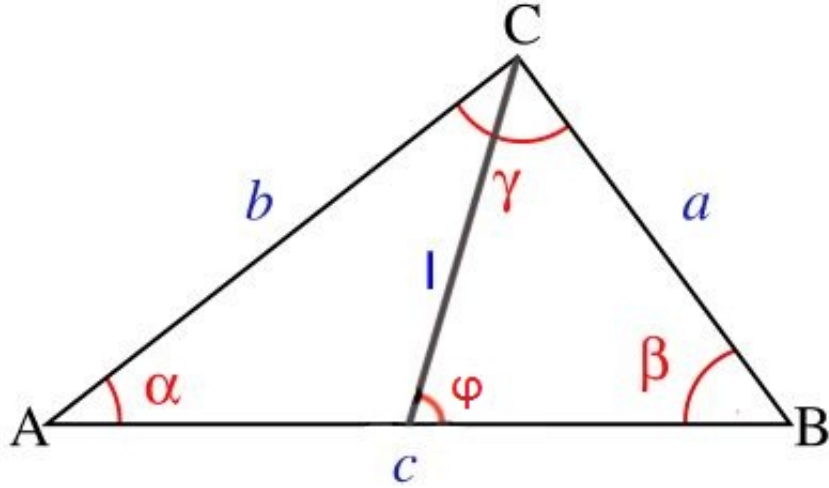


Figure 36: Triangle used to explain the calculations below.

$$b^2 = a^2 + c^2 - 2ac \cos(\beta) \quad (22)$$

$$\cos(\beta) = \frac{a^2 + c^2 - b^2}{2ac} \quad (23)$$

To determine the angle φ , the length l needs to be known. By viewing the triangle constructed by the length a, l and $c/2$ the length l can be obtained by the law of cosine as described in equation 24.

$$l^2 = a^2 + \left(\frac{c}{2}\right)^2 - 2a\frac{c}{2} \cos(\beta) \quad (24)$$

$$a^2 = l^2 + \left(\frac{c}{2}\right)^2 - 2l\frac{c}{2} \cos(\varphi) \quad (25)$$

$$\cos(\varphi) = \frac{l^2 + \left(\frac{c}{2}\right)^2 - a^2}{2l\frac{c}{2}} \quad (26)$$

$$x_{coordinate} = l * \cos \varphi \quad (27)$$

$$y_{coordinate} = l * \sin \varphi \quad (28)$$

Triangulation and knowledge of the sensors' positions relative to a given point, in this case the bumper on the vehicle's center line, can provide both longitudinal and lateral position.

6.5 Additional Graphs on Sensor Fluctuations With and Without Low Pass Filter

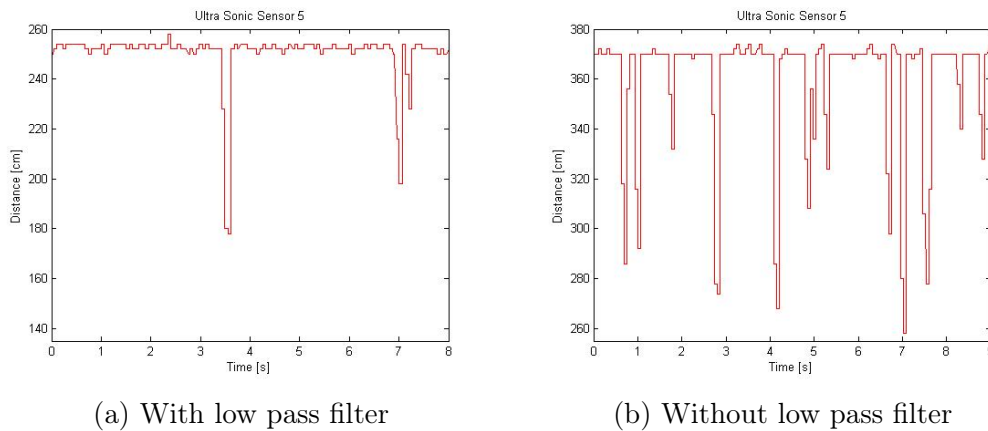


Figure 37: Sensor fluctuations with and without the use of a low pass filter

6.6 Arduino Code for Managing the Sensors

```
#include <Wire.h>
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>

//=====

// In case of more sensors, change following variables:
int numSensor = 6;           // Number of sensors
int ultraSoundSignalPins[]={0,1,2,3,4,5}; // Sensor port number
int langd[]={0,0,0,0,0,0}; // Array to store data
byte sendBuffer[] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
// Data to send via UDP 4 byte/sensor

// Init setup for sensors
//-----
int i=0;
unsigned long time = 0;
int ultraSoundTriggerPin = 8;
// Output pin to start Ultrasound signals

// Init setup for UDP & Ethernet
//-----
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x51, 0xC2 };
// Mac for Arduino
```

```

IPAddress ip(192,168,188,200);
// IP for Arduino
IPAddress remoteIP(192,168,188,41);
// IP to MicroAutoBox
unsigned int localPort = 0255;
// Port to listen on
unsigned int remotePort = 50390;
// Port to send on

EthernetUDP Udp;

//=====

void setup()
{
  Serial.begin(9600);
  // Config of sensors
  //-----
  pinMode(ultraSoundTriggerPin, OUTPUT); // set this pin to output
  //give the sensors time to boot up
  for(int i=0; i < numSensor; i++) {
    pinMode(ultraSoundSignalPins[i], INPUT);
    // Switch signalpin to input
  }
  delay(250);

  // Config of UDP & Ethernet
  //-----
  Ethernet.begin(mac,ip);
  Udp.begin(localPort);
}

//=====

void loop() {
  // Recive data from sensors
  //-----
  digitalWrite(ultraSoundTriggerPin, HIGH);
  delayMicroseconds(1000);
  digitalWrite(ultraSoundTriggerPin, LOW);

  delay(15);

  for(i=0; i < numSensor; i++) {
    langd[i]=(int) (analogRead(ultraSoundSignalPins[i])*2.00);
    if(langd[i] > 500) {

```

```

        langd[i] = 500;
    }
    sendBuffer[i*4] = (langd[i] & 0xFF);
    sendBuffer[i*4 + 1] = ((langd[i] >> 8) & 0xFF);
}

delay(35);

// Display data in monitor
//-----
for(i=0; i < numSensor; i++) {
    Serial.print(langd[i]);
    Serial.print(" ");
}
Serial.print(" ");
time = millis();
Serial.println(time);

// Send data on UDP
//-----
    Udp.beginPacket(remoteIP, remotePort);
    Udp.write(sendBuffer,numSensor*4);
    Udp.endPacket();
//
    delay(10);
}

```

6.7 Simulink Models and Code

6.7.1 Code for Calculation of Critical Distance

```
function critical = fcn(acc,state,velocity, gear,Sensor1,
Sensor2, Sensor3, Sensor4, Sensor5, Sensor6, sm, a_avg,
j_avg, t_del)

v_in = velocity;
reverse = 1;

v_half = v_in/2;           % half initial velocity
t_v_half = sqrt(-v_in/j_avg); % time to decelerate to v_half
a_v_half = j_avg*t_v_half;  % achieved deceleration at v_half
t_a_avg = a_avg/j_avg;      % time to achieve a_avg

% If the cars deceleration reduces the velocity more than v_half
during t_a_avg. The car will not achieve a_avg, instead the
maximum is a_v_half

if(-(j_avg*t_a_avg^2)/2 > v_half)

    % velocity achived after t_v_half
    v_jerk_beg = v_in + (j_avg*t_v_half^2)/2;

    % velocity achived after 2*t_v_half
    v_jerk_end = v_jerk_beg + a_v_half*t_v_half -
        (j_avg*t_v_half^2)/2;

    % distance the car travels during the timedelay
    s_del = v_in*t_del;

    s_jerk_beg = v_in*t_v_half + (j_avg*t_v_half^3)/6;

    s_jerk_end = v_jerk_beg*t_v_half + (a_v_half*t_v_half^2)/2 -
        (j_avg*t_v_half^3)/6;

    s_brake = s_del + s_jerk_beg + s_jerk_end

% If the car does not half the velocity during t_a_avg the car
will decelerate continuously with a_avg for a time t_cont.
Then the acceleration will go to zero with rate of change j_avg

else
    t_cont = -(v_in + j_avg*t_a_avg^2)/a_avg
```

```

v_jerk_beg = v_in + (j_avg*t_a_avg^2)/2
v_cont = v_jerk_beg + a_avg*t_cont
v_jerk_end = v_cont + a_avg*t_a_avg - (j_avg*t_a_avg^2)/2

s_del = v_in*t_del;

s_jerk_beg = v_in*t_a_avg + (j_avg*t_a_avg^3)/6

s_cont = v_jerk_beg*t_cont + a_avg*t_cont^2/2;

s_jerk_end = v_cont*t_a_avg + (a_avg*t_a_avg^2)/2 -
(j_avg*t_a_avg^3)/6;

s_brake = s_del + s_jerk_beg + s_cont + s_jerk_end;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SYSTEM ACTIVE / SYSTEM DISABLED
%Decided on wether sytem is activated or not
if (v_in >= 1.8)
    safety = sm + 0.15;
else
    safety = sm;
end

s_stop = (s_brake + safety)*100;

if (s_stop>=Sensor1 || s_stop>=Sensor2 || s_stop>=Sensor3
    || s_stop>=Sensor4 || s_stop>=Sensor5 ||
    s_stop>=Sensor6) && (gear==reverse)
    critical = 1;
else
    critical = 0;
end
end

```

6.7.2 Simulink State Decider

```
function [brake, stopmode] = fcn(state, chosen_acc)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%APREHENDED COLLISION OR GOING TO COLLIDE
%Decide whether it's a critical condition or not
if state==1
    %Preload brakes
    brake = 0;
    stopmode = 0;

elseif state == 0
    brake = 0;
    stopmode = 0;

elseif state == 2
    brake = chosen_acc;
    stopmode = 0;

elseif state == 3
    brake = 0;
    stopmode = 1;

elseif state == 4
    stopmode = 0;
    brake=0;
elseif state == 5
    stopmode = 0;
    brake=0;
else
    brake = 0;
    stopmode = 0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

6.8 Simulink Model

Below are overviews of the different layers in the Simulink model

6.8.1 Simulink Overview

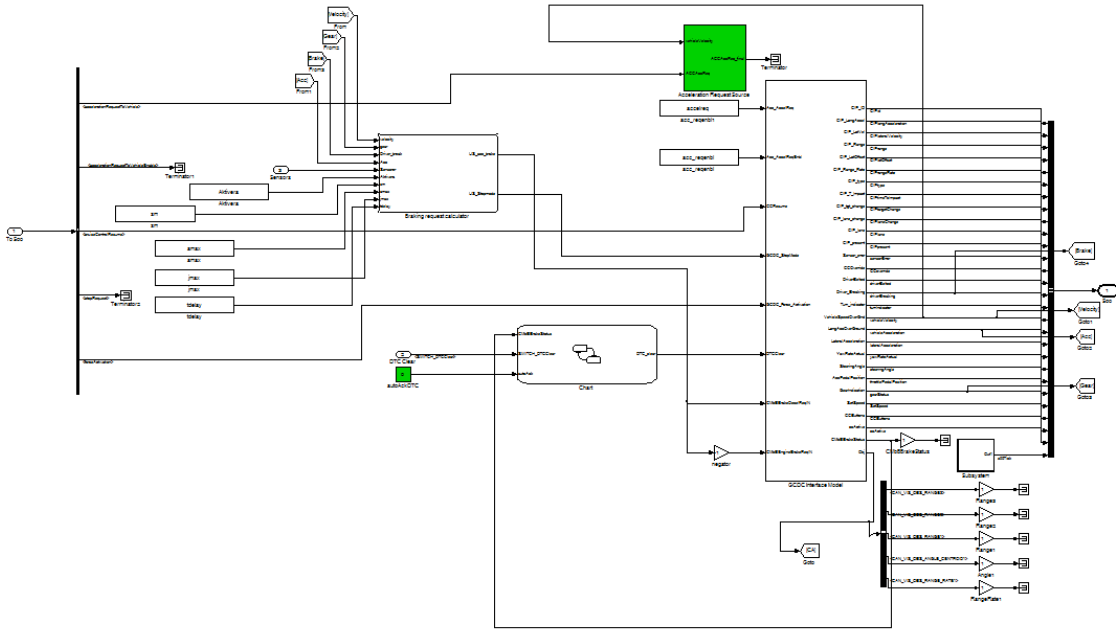


Figure 38: Overview of the Simulink model

6.8.2 Inner Simulink model

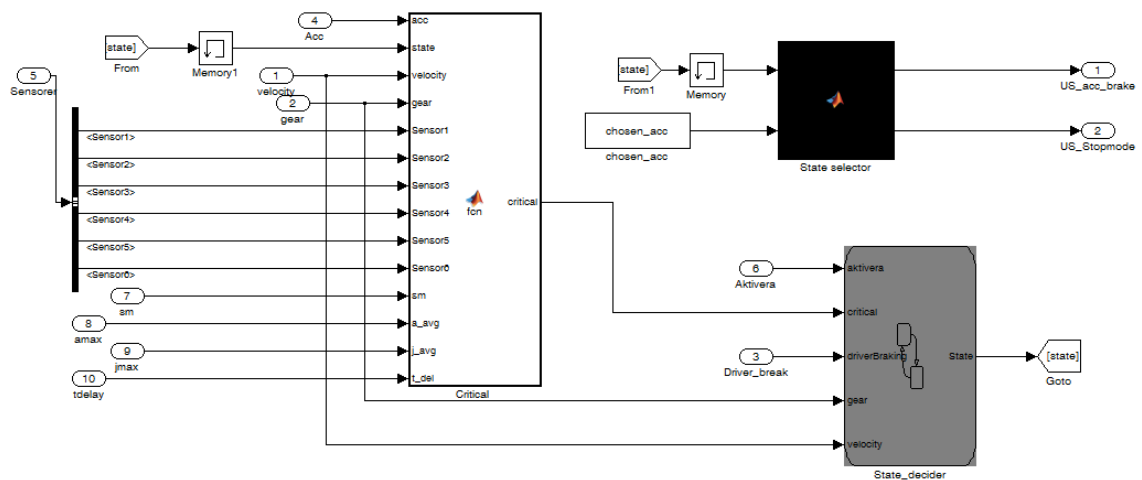


Figure 39: Inner Simulink model

6.8.3 Simulink UDP Block

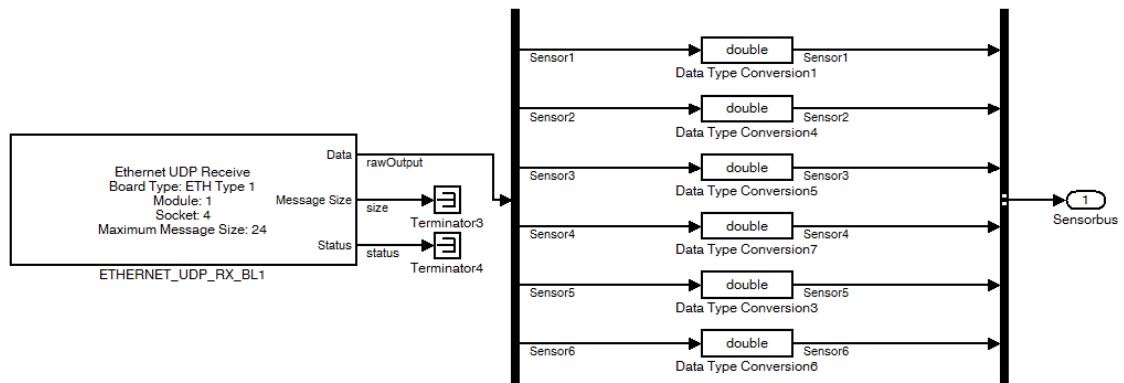


Figure 40: Simulink UDP block