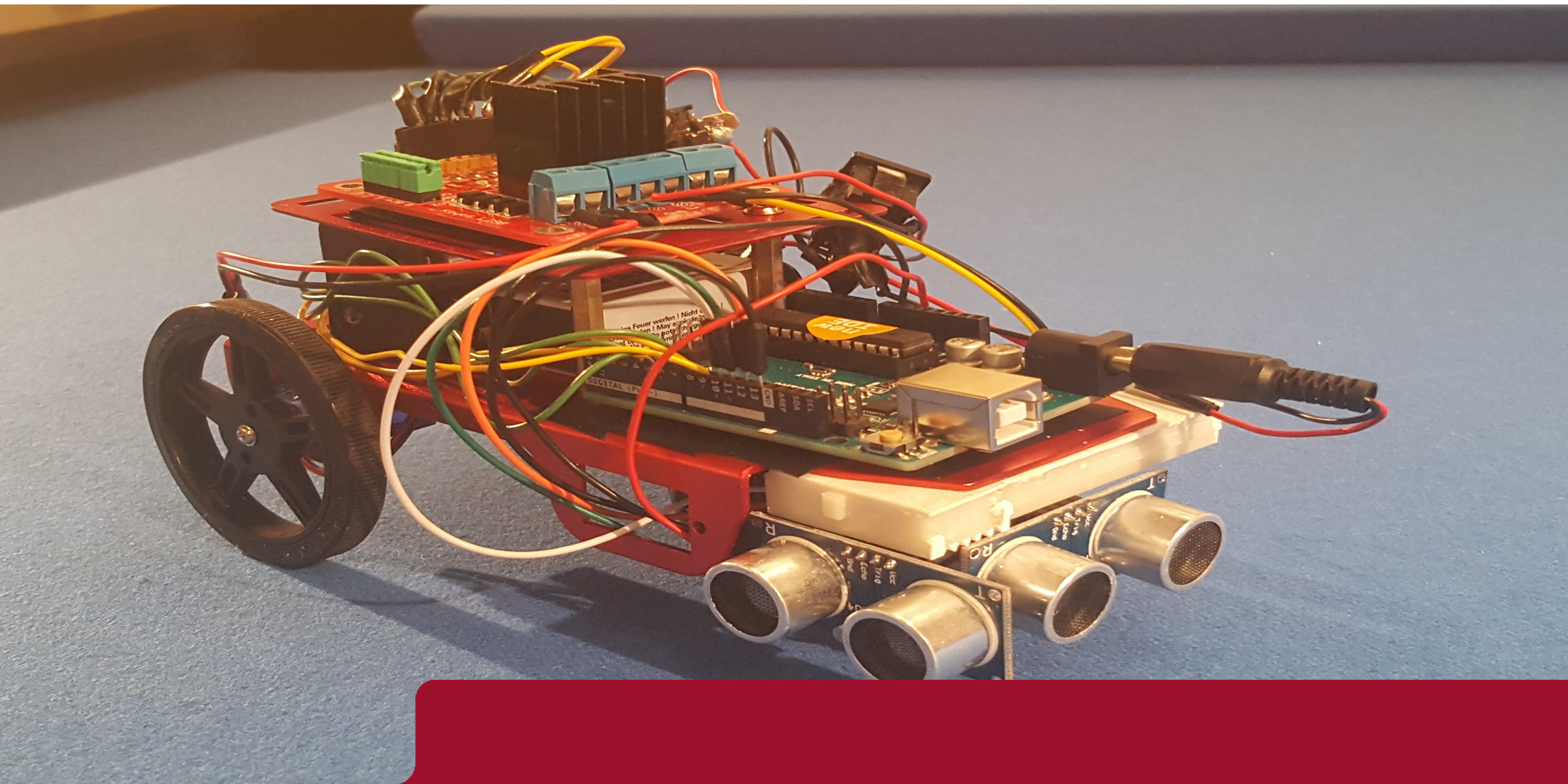
EXAMENSARBETE INOM TEKNIK,
GRUNDNIVÅ, 15 HP
STOCKHOLM, SVERIGE 2016

Self-driving car

JACOB EKESUND



KTH
SKOLAN FÖR INDUSTRIELL TEKNIK OCH MANAGEMENT

KTH Industrial Engineering and Management	**Bachelor's Thesis MMKB 2016:35 MDAB096** **Self-driving car** Jacob Ekesund	
Approved 2016-06-07	Examiner Martin Edin Grimheden	Supervisor Nihad Subasic

ABSTRACT

Imagine to be able to catch a little more sleep on your way to work or school, drive home after a night at the bar or watch a movie on a long road trip. All these things have never been possible before without someone else driving the car, until recent years. Autonomous cars or self-driving cars is being introduced to society more and more and will be the next big step in the progression of personal cars. There are a number of factors that decides how fast this new technology will be adopted. Safety, reliability, ethics and cost to name a few. This project will focus on the cost aspect of self-driving cars by examine if ultrasonic sensors can be used to develop a cheap self-driving car and thereby reach a broad customer base. To determine this a small scale prototype car was built and tested in a highway cruising situation and the results showed that the prototype was able to drive itself.

Kandidatarbete MMKB 2016:35 MDAB096		
Self-driving car		
Jacob Ekesund		
Godkänt 2016-06-07	Examinator Martin Edin Grimheden	Handledare Nihad Subasic

SAMMANFATTNING

Självkörande bil

Tänk dig att kunna sova några extra minuter på väg till jobbet eller skolan, köra hem efter en natt i baren eller titta på film under en lång bilresa. Dessa saker har tidigare bara varit möjligt genom att ha en annan person som kör bilen, tills nu. Självstyrande bilar håller på att introduceras till samhället mer och mer och kommer vara det nästa stora steg som bilindustrin kommer ta. Det finns flera faktorer som bestämmer hur snabbt den här teknologin kommer adopteras. Säkerhet, pålitlighet, etik och kostnad för att nämna några. Det här projektet kommer att fokusera på kostnadsaspekten gällande självstyrande bilar, genom att undersöka om ultraljudssensorer kan användas vid utvecklandet av en självstyrande bil och på så sätt kunna hålla nere kostnaderna och nå en bredare kundbas. För att fastställa detta byggdes en småskalig prototypbil och testades i en simulation av motorvägskörning. Resultatet av testerna visar att bilen kunde köra sig själv med endast information från ultraljudssenorerna.

PREFACE

I would like to thank Nihad Subasic and the lab assistants for their counseling throughout the project. I would also like to thank Ludvig Bjärkeback and Viking Björk Friström for their feedback on the early stages of this thesis.

Jacob Ekesund

Stockholm, May, 2016

CONTENTS

NOMENCLATURE

Symbols

Symbol	Description
$r(t)$	Reference signal
$y(t)$	Output signal
u(t)	Input signal
t	Time (μs)
d	Distance (cm)
V_{sound}	Velocity (m/s)
K_p	Proportional constant
K_I	Integral constant
K_D	Derivative constant
κ	Boltzmann constant
R	Gas constant
T	Temperature (K)
$e(t)$	Signal error

Abbreviations

PID	Proportional Integral Derivative
PWM	Pulse Width Modulation
DC	Direct current

1 Introduction

1.1 Background

A self-driving car uses input from a variety of sources as GPS, laser rangefinders, radar and cameras. Software uses the information from these inputs and creates a 3-dimensional environment around the car and calculates in which way the car should maneuver [Google, 2016]. There are a number of factors that decides how fast this new technology will be adopted. Safety, reliability, ethics and cost to name a few. The vast majority of these cars are either not yet available to the public or costs too much for the average consumer.

1.2 Purpose

The purpose of this project is to research how suitable ultrasonic sensors are for use in a control system that allows a car to autonomously cruise on a highway. In addition, discuss if it can be a cheaper alternative to the multiple different sensors that are used in self-driving cars today. To help determine this, a small scale prototype car will be built and thereafter examine to answer the research questions:

Can a control system, that only receives sonar input, be developed in such a way that the car can follow a road with minimum deviation and in a smooth motion?

Is there a way to avoid obstacles in a safe way with only the input from the ultrasonic sensors?

The prototype will be equipped with two ultrasonic sensors. One at the front to detect obstacles and one on the right side to detect the distance to a crash barrier that most highways have on the side of the road.

1.3 Scope

The prototype will be designed to only drive in a simulated highway situation where there is a crash barrier at the side of the road present. This means the car will not have to be able to take particular sharp corners. This project main focus lies in the cars ability to follow a road by keeping itself at a fixed distance from the crash barrier. The lack of sensors at the left side limits the car when avoiding obstacles because it will not know if it can turn left to drive around an obstacle. Therefore the car will just stop if it come across a stationary obstacle or if it's a moving object like a slower moving car it will match its speed to the car in front.

The project will have a time limit of one semester and a budget of 1000 SEK.

1.4 Method

The prototype is built from a 3-wheeled robot base, see figure 1. The two back wheels is each connected to a separate DC-motor and the car can be steered by providing different power to the motors. To know its distance to the crash barrier the car has an ultrasonic sensor of model HC-SRF04 mounted at its side. It is also equipped with an ultrasonic sensor, of the same model, at the front to detect obstacles. The sensor is connected to an Arduino Uno that uses the information from the sensor and controls the power output to each motor side. To feed the motors with the desired output power the Arduino is connected to a dual full bridge driver of model Keyes L298 which allows for controlling two different motors by sending PWM-signals to it.

The car have two different control systems. One PID-controller that controls the cars distance to the crash barrier so it stays at a predetermined set point. The other control system is a PI-controller that controls the cars speed so it never comes within a preset minimum distance to a moving or stationary obstacle.

To answer previously asked research questions a test track with a crash barrier or fence will be built and the car will first drive on the test track with no obstacles to determine how well the control system for the steering works. Then it will drive the course with a still obstacle placed on the track and be examined if it can stop in time to not crash. The last test will have a moving obstacle to see how well the car adjusts its speed to be the same as the moving obstacle.

Figure 1. The chassis the prototype is built on.

1.5 Ultrasonic sensors in cars

Ultrasonic sensors have become a common thing for modern cars to be equipped with. They are mostly used for park assistance which gives the driver a warning when driving to close to an object. Volvo is currently working on a big project called Drive Me. The goal of the project is to release 100 self-driving cars to the public in 2017. The self-driving cars will be designed using several different types of sensors that will act as backup for each other if some were to break. The different types of sensors are multiple cameras all around the car, radar at the cars four corners, infrared sensors, GPS and at last it is equipped with twelve ultrasonic sensors all around the car. These ultrasonic sensors are used to detect other vehicles, bicyclists and pedestrians around the car. This is also done by the infrared sensors and the radar. By having multiple different sensors measuring the same things they can verify each other measurements but they also act as independent systems and if one go down the others can still provide sufficient measurements [Volvo Car Group, 2015].

2 THEORY

2.1 Measure distance with an ultrasonic sensor

An ultrasonic sensor is a cheap way of measuring distance. It works by sending out sound waves with a frequency higher than 20 000 Hz. This is the limit of what a human ear can hear which means ultrasonic sensors appear soundless to humans. Hence the name *ultrasonic* sensor. The sound wave then bounce back of the object in front of the sensor and can get picked up by the receiver [Teach Engineering, 2016]. See figure 2.

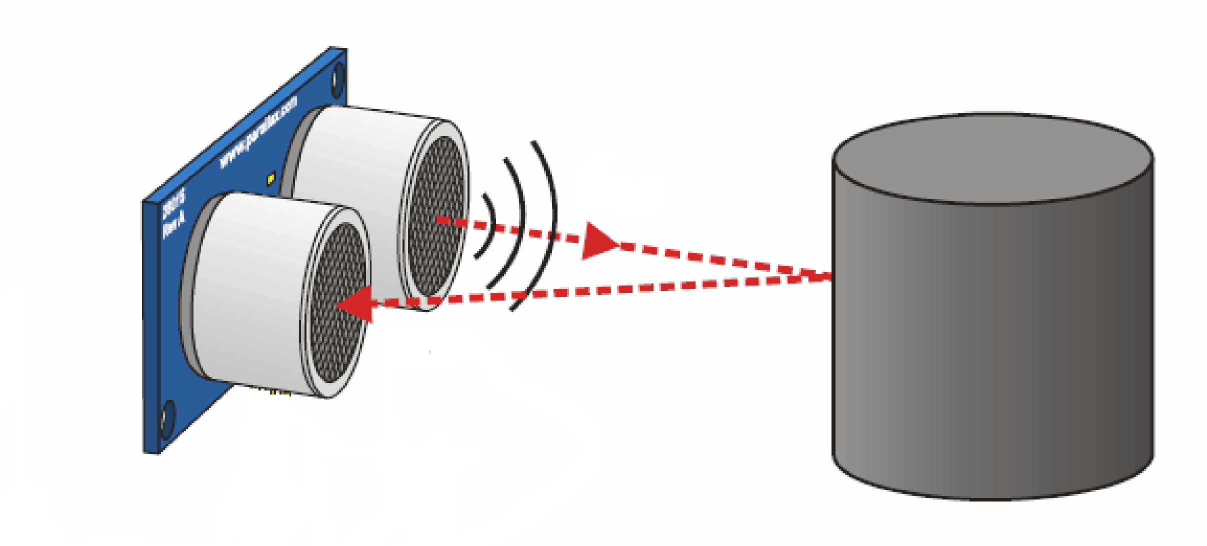


Figure 2. Visualization of how ultrasonic sensors work [Gantt, 2015].

The time it takes for the sound to bounce back is measured and used when calculating the distance. By knowing the speed of sound, the distance traveled in the measured time can be calculated using this simple formula:

$$d = V_{sound} \cdot t\ , \tag{1}$$

where d is the distance, t is the time and V_{sound} is the speed of sound which depends on the temperature. If we treat air as an ideal gas the speed of sound can be calculate with:

$$V_{sound} = \sqrt{\kappa \cdot R \cdot T}\ , \tag{2}$$

where κ and R are constants and T is the temperature in Kelvin [Young and Freedman, 2014]. The time measured by the ultrasonic sensor is the time it takes for the sound to travel back and forth which will have to be accounted for when calculating the distance by only using half of the measured time. The function becomes

$$d = V_{sound} \cdot \frac{t}{2}. \quad (3)$$

2.2 PID controller

A PID (Proportional-Integral-Derivative) controller consists of three components. All three components are functions of the signal error $e(t)$, which is the difference between desired output signal $r(t)$ and actual output signal $y (t)$. The signal error can be written as:

$$e(t) = r(t) - y(t). \quad (4)$$

The proportional part is as the name suggest proportional to the signal error $e(t)$ and can be calculated as:

$$P = K_P \cdot e(t), \quad (5)$$

where K_P is the proportional constant. The proportional part usually have the most impact on the output signal and it dictates how fast the system is. One problem with only using a proportional controller is when the error approaches zero the output signal also becomes zero and you get something called steady state error. A good way to avoid this problem is to implement an integral part that is proportional to the accumulated error over time. This means that the steady state error will disappear because when the signal error approaches zero previously errors will still effect the input signal. The integral part can be written as:

$$I = K_I \cdot \int_0^t e(\tau)d\tau, \quad (6)$$

where K_I is the integral constant. The system can also be made faster by increasing the value of K_I, however with too big values on both K_P and K_I the system might become unstable or have a big overshoot. To dampen these effects a derivative part can be added. The derivative part is proportional to the derivative of the error. This means that if the error changes quickly the derivative part will have a bigger effect on the input signal, thus lower the overshoot. The derivative part can be described as:

$$D = K_D \cdot \frac{d}{dt}e(t), \quad (7)$$

where K_D is the derivative constant. (5), (6) and (7) are combined to get the function for a PID controller which is

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau)d\tau + K_D \cdot \frac{d}{dt}e(t), \quad (8)$$

where $u(t)$ is the input signal [Glad and Ljung, 2014].

To determine the signal error a PID controller is connected in a feedback loop, where the output signal $y\ (t)$ is measured and sent back to the controller. Shown in figure 3.

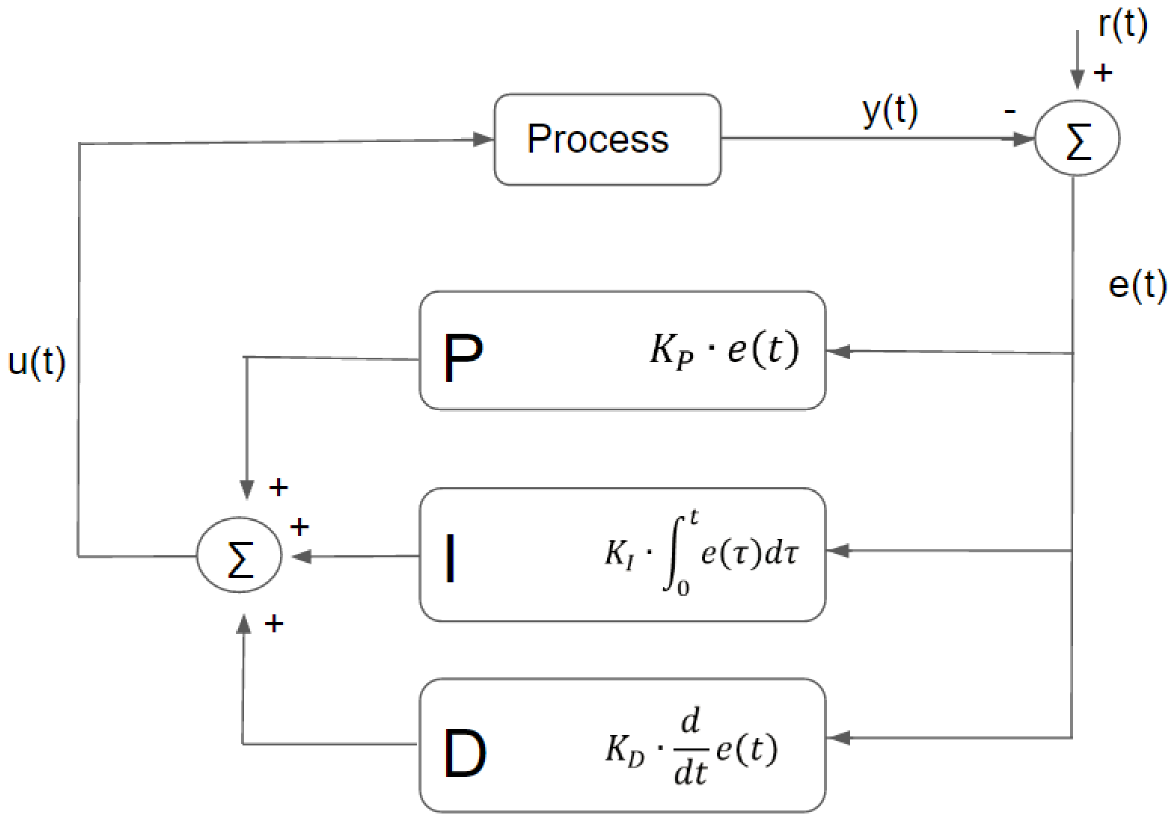


Figure 3. Feedback loop of control system, created with [Google Presentation, 2016].

2.3 DC-motor control

To control DC-motors an H-bridge can be used. An H-bridge work by having four switches that can be closed or open. Depending on which switches that are open and closed the direction the current flows through the motor changes. So by controlling which switches are closed and open the direction of the motor can be controlled. This is demonstrated in figure 4. When S1 and S3 are closed and the rest are opened the current travels from left to right through the motor and when S2 and S4 are closed and the rest are opened the current changes direction and travels from right to left through the motor [Johansson 2013].

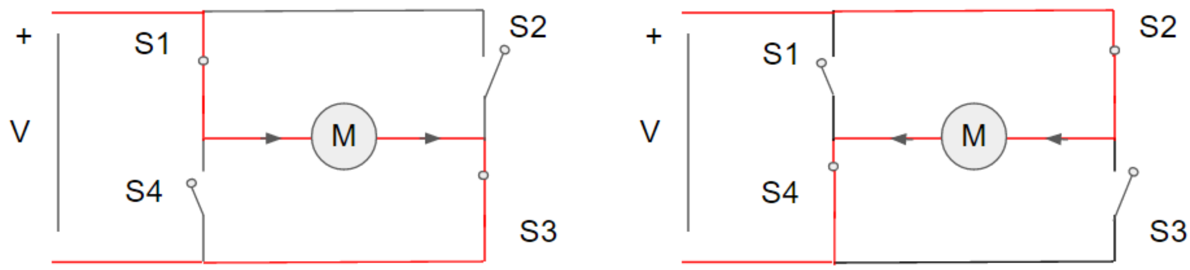


Figure 4. Demonstrate different current directions, created with [Google Presentation, 2016]

H-bridges are also very useful when controlling motor speeds. An H-bridge can transform a DC current to an AC current by switching the switches on and off. This creates square waves of current that drive the motor. What determines the speed of the motor is the amount of time the switches are on versus off. This relation can be described with duty cycle. Where 100 % duty cycle the switches are always on, 50 % duty cycle the switches are on half of the time and 0 % duty cycle the switches are always off. This square wave signal is called a PWM (Pulse Width Modulation) signal. In figure 5 examples of PWM signals for different duty cycles are shown. In this project a dual H-bridge have been used which is basically two H-bridges that are parallel connected to the same external power source but they are connected to different motors and different ports on the Arduino.

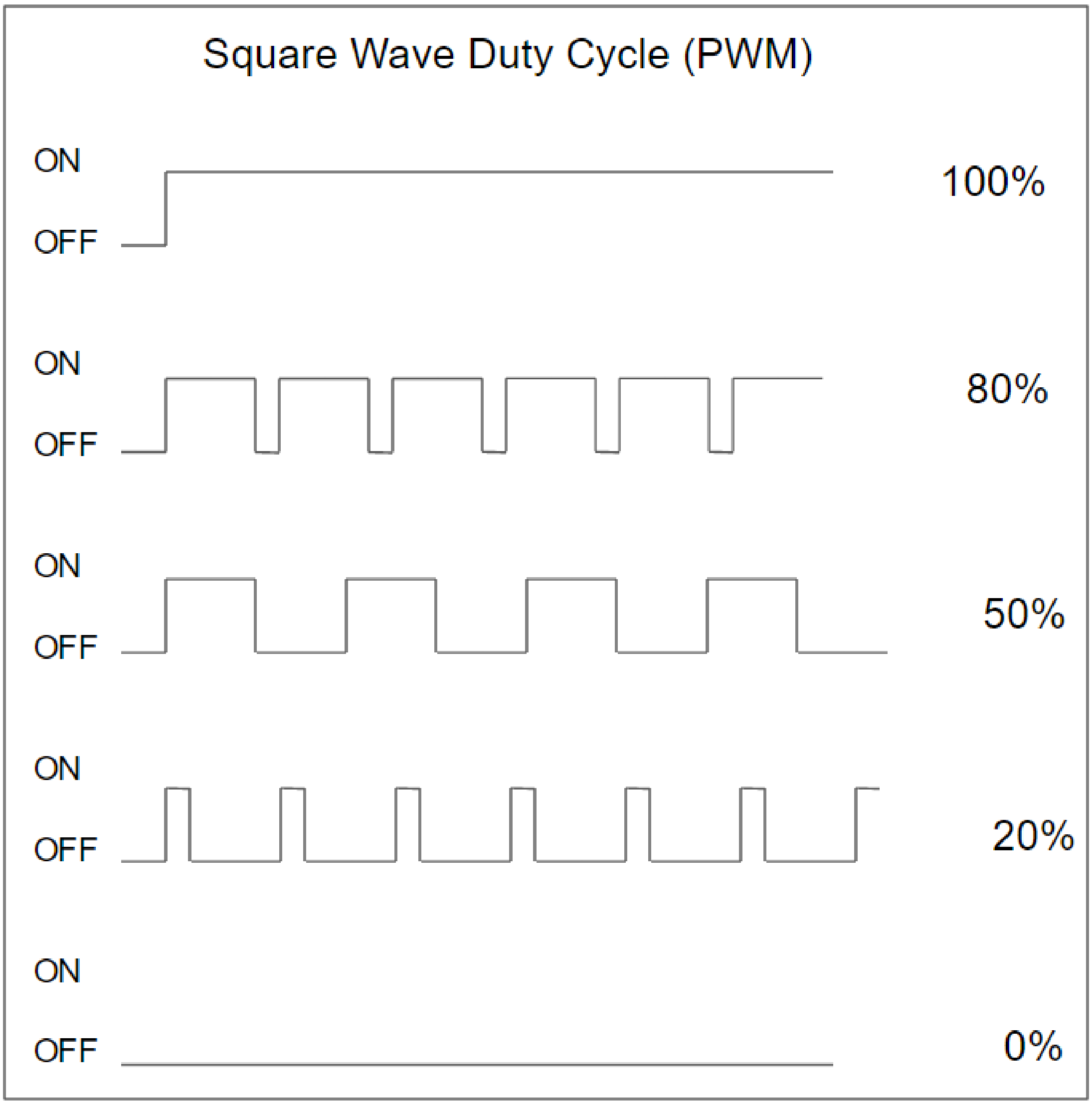


Figure 5. PWM signals for different duty cycles, created with [Google Presentation, 2016].

3 DEMONSTRATOR

3.1 Problem formulation

The demonstrator needs to accurately measure distances on its right side and in front of it. To accomplish this two ultrasonic sensors will be used. One mounted on the right side and the other mounted in the front. To account for noise and disturbance the software needs to filter out false readings from the sensors. The car must be able to adjust its speed and position depending on the sensor readings. Therefor the car will have a 3-wheel design whit two back wheels, each connected to a separate DC motor. The motors have to be controlled individually in order to steer the car. To determine how the car should maneuver a PID controller will be used to control the cars position relative to a wall/barrier on its right side. To control the cars speed a PI controller will be used to make sure the car never comes to close to an object in front of it. The two DC motors will be connected to a dual H-bridge which allows for individually controlling the motors using PWM signals. The dual bridge will be connected to an external power source to power the motors.

The main problems that needs to be solved are:

- Reading the distances from the sensor and filter out false readings.
- Develop a PID controller for position control and a PI controller for speed control.
- Develop software for controlling the motors.

3.2 Software

For reading the ultrasonic sensors an open source library [Eckel, 2015] have been used. The reasoning behind using this library is that it contains several useful methods. It also makes it easier to connect multiple sensors and keeps the sketch more structured and easy to follow. Both sensors are pinged by using a method within the library. This method returns the median of five measurements in microseconds. This median filter will get rid of any noise and disturbance which allows for more accurate readings. To calculate the distance a method that takes the measured time and returns the distance in centimeter is used. This method uses the measured time with (3) and the speed of sound is calculated for 20° C with (2), which is normal indoor conditions.

The PID controller was design using multiple tests and by analyzing the results and having an understanding of how the different terms of a PID controller effect the performance. The work process of tuning the PID controller was to first tweak the P part and the I part to make the system faster. Then tune the D part to dampen the negative effects that was big overshoot and oscillation. One of the biggest problems faced in this project was that the car kept oscillate even when following a straight wall. After a great number of attempts to tune the PID controller to get rid of the oscillation with no luck, a new approach was taken. Instead of calculating the distance to the wall and using (4) to calculate the error, the error was calculated in microseconds. The preset distance was instead set to be a time duration. To convert previously preset distance to a time duration (1) can be rewriting to:

$$t = \frac{d}{V_{sound}}, \tag{9}$$

where d is the desired preset distance and t is the preset time duration to use when calculating the signal error. This new method of calculating the error results in an error that changes more dynamically over time instead of in one centimeter increments. This means the PID controller is more responsive and the oscillation have been reduced significant.

The PI controller that is used to control the cars speed is using the measured distance in centimeter to calculate the error. This is because through testing the controller was deemed accurate enough without switching to calculate the error with time duration. It was also determined that a D-part was unnecessary to implement because overshoot was not a problem.

The motors are controlled by first setting the direction of the motors and then sending PWM signals that determines the duty cycle of each motor. The PID controller and the PI controller have been implemented in such a way that first the PID controller takes the measurement from the side sensor calculates the error compared to a preset value and uses the error with (8) to get a steering value. This steering value is the power difference between the two motors. After this the PI controller takes the measurement from the front sensor and if the measurement is below a preset value the PI controller will start to regulate the power output. The PI controller uses the difference between the preset value and measured distance and uses that error with (5) and (6) to calculate the power variable. To calculate what PWM signals to send the program takes 255 which is 100 % duty cycle in PWM signals and subtract the power variable. This becomes the base PWM signal and power output. Then the steering variable is applied and two different PWM signals are created one that adds half of the steering variable to the base PWM signal and the other one subtracts half of the steering variable from the base PWM signal. The two PWM signals are the desired duty cycle for the motors and are sent to the H-bridge which drive the motors.

See figure 6 for a flowchart of the software.

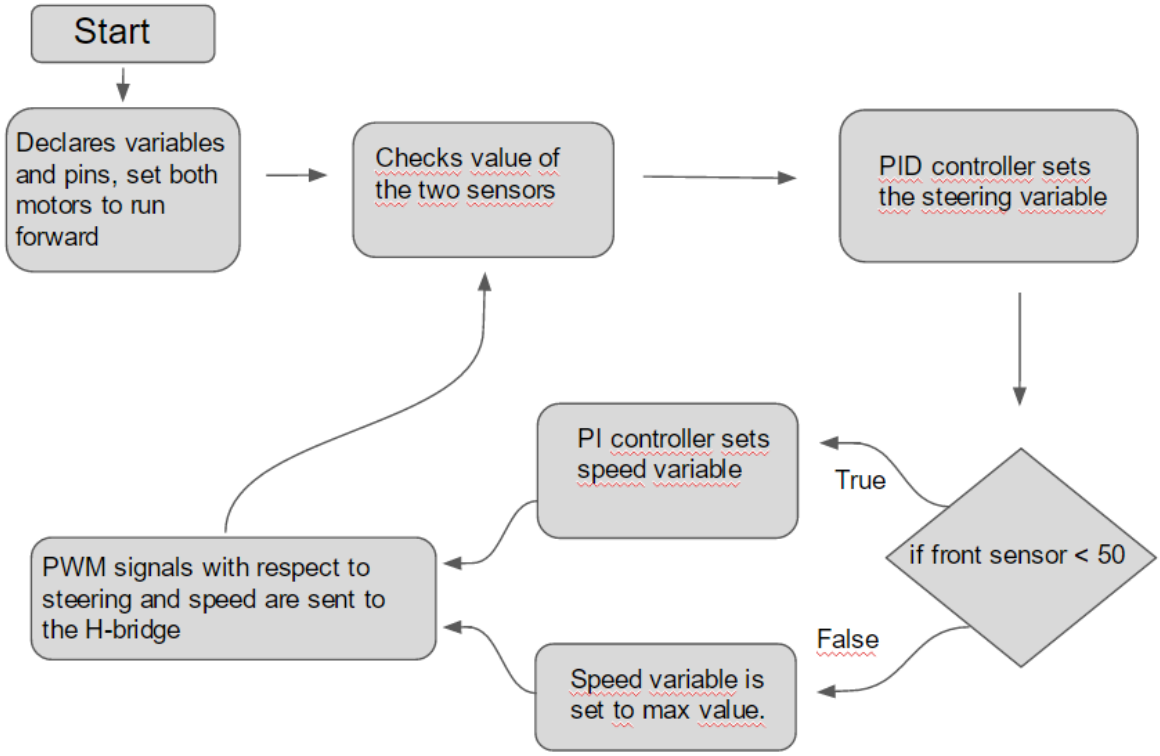


Figure 6. Flowchart of software [Google Presentation, 2016]

3.3 Electronics

3.3.1 Arduino Uno

To handle all the processing the car needs to do an Arduino Uno is being used. Arduino Uno is an open source microcontroller that has 32 KB flash memory, 2KB SRAM and 1 KB EEPROM. Its input voltage can range between 6-20 V and its operating voltage is 5 V [Arduino, 2010]. The Arduino is connected to both ultrasonic sensors and processes the information gained by the sensors via the control systems and then sends the calculated PWM signals to the H-bridge. To power the Arduino a 9v battery is connect to the Arduinos DC power jack.

3.3.2 Ultrasonic sensors

As mentioned before, the car uses two ultrasonic sensors to keep track of its position relative obstacles. These sensors are of model HC-SR04. This model has a maximum range of 5000mm and a minimum range of 20mm with an error of margin of 3mm. The sensors are connected to and powered by the Arduino.

3.3.3 Motor controller

The two motors are individually controlled using a dual H-bridge of model Keyes L298. The H-bridge is connected to an external power source which consists of four AA battery in serial connection. The batteries provide a voltage output of 6V.

A detailed skis of the electrical connections can be seen in figure 7.

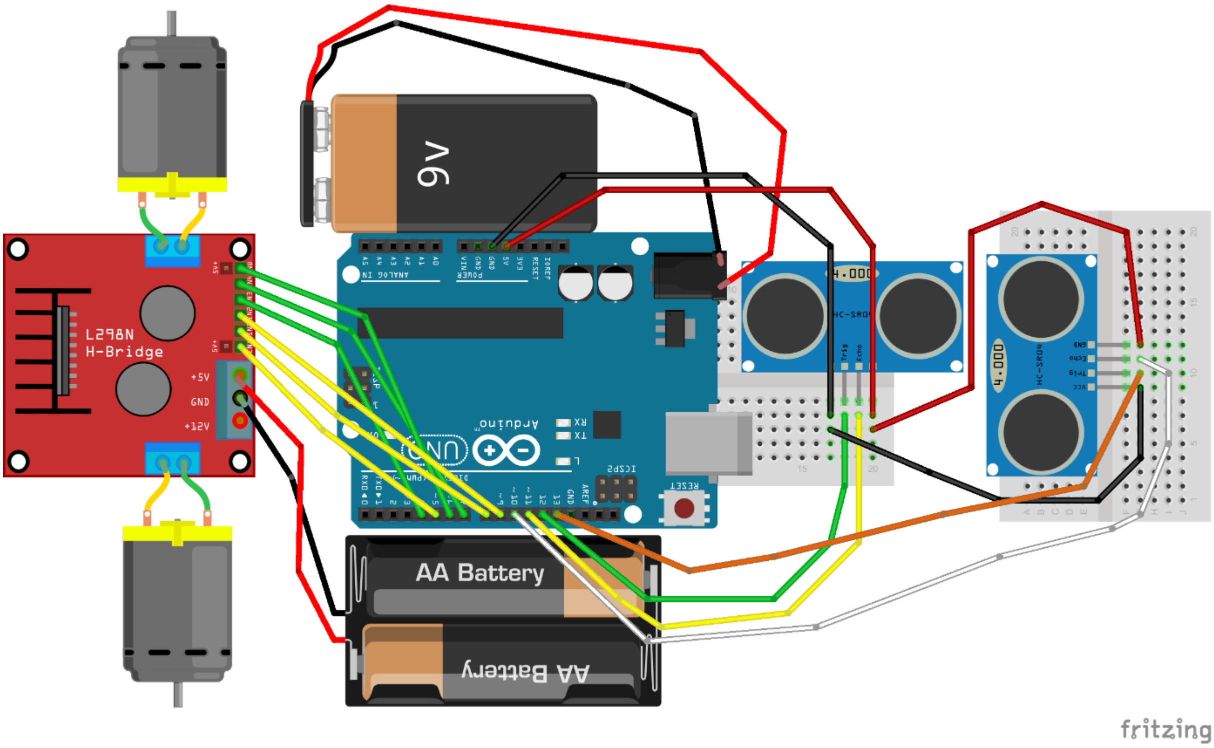


Figure 7. Electrical connection sketch, created with [Fritizing, 2016].

3.4 Hardware

The demonstrator uses a very simple design with very little moving parts. It consists of a chassis to which all the electronics are mounted on, one front wheel that can rotate freely 360° and two back wheels that each are connected to separate DC motors. The car can turn by simply giving one of the wheels more power than the other. To help the car turn with minimal effort the car have been designed to have its center of gravity as close to the back wheels as possible. This is because a solid body of mass tends to rotate around its center of gravity and by having the cars center of gravity close to the wheel base it turns more easily.

3.5 Results

To measure the results of the car a wall consisting of cardboard boxes and wooden boards was built. The wall curved both left and right and is supposed to simulate a crash barrier that can be found at the side of a highway. The test track is shown in figure 8. The car drove itself on this test track and was visually analyzed to determine how well it performed. To test how well the car stopped when it encountered a stationary obstacle a different test case was performed. The car drove itself in a straight line with an obstacle placed 2 meters in front of it. It was then measured from what distance of the obstacle the car stopped. The results are presented in table 1. The last test case was to test how well the car could adept its speed to a moving obstacle. This test was also visually analyzed.

Table 1. Results from the second test case.

Test	Distance to obstacle (cm)
1	24,0
2	24,5
3	23,5
4	24,0
5	23,5
6	24,0
7	24,0
8	24,5
9	24,0
10	24,5

The car is programmed to keep itself at 15 cm distance to the wall and stay at a minimum of 20 cm from objects in front of it. The constants in the PID controller and PI controller is shown in table 2. The reason for that the PI controller have a much larger proportional constant is than the PID controller is that the error is measured in cm for the PI controller and the error is measured in microseconds for the PID controller. This means the error in general will be much bigger in the PID controller, hence the smaller proportional constant. The constants for the control systems are shown in table 2.

Table 2. Constants of the PID controller and the PI controller.

PID controller	Constant value	PI controller	Constant value
K_p	$\frac{1}{7.3}$	K_{p2}	2
K_I	$\frac{1}{20}$	K_{I2}	1
K_D	2		

The results of the visually analyzed tests were that the car could keep itself within the set distance and had very little deviation through the left and right turns. It still does oscillate a little bit when driving on a straight track which is a result of the sensor having a margin of error that is 3mm. In the moving obstacle avoidance test the car successfully adjusted its speed to the moving target but would drift a few centimeters closer or further away.

Figure 8. Picture of the test track.

4 DISCUSSION AND CONCLUSIONS

4.1 Discussion

The purpose of this project was to investigate whether ultrasonic sensors can be a viable alternative to the multiple different sensors that are used in autonomous cars today. Two research questions were asked and a prototype car was built to answer these questions. The first research question was:

Can a control system, that only receives sonar input, be developed in such a way that the car can follow a road with minimum deviation and in a smooth motion?

There is not a clear yes or no to this question. Judging by the results you could say that the car manage to drive itself in a smooth manner and had very little deviation from the set point. But you could also give the argument that the car slightly oscillate and the oscillation is a result of the ultrasonic senor having a margin of error that is 3 mm. These tests are also performed on a much smaller scale and at much slower speeds so it is hard to tell what the outcome would be on a bigger scale where the speeds are much greater. One could argue that the oscillation would become a nonfactor because a margin of error of 3 mm would not have the same effect when the measured distances are several meters. On the other hand when scaling up the experiments and increasing the speeds the demand on the performance of the PID controller increases and you might find that the PID controller performs inadequate.

The second research questions asked was:

Is there a way to avoid obstacles in a safe way with only the input from the ultrasonic sensors? Limitations in time made that the implementation of more ultrasonic sensors were deprioritized. This resulted in that the car could not detect if there were any objects on its left side. Because of this, driving around an obstacle could not be made in a safe way. Therefor the focus of the project switched to just make the car avoid the obstacles by lowering its speed. The results show that the car could reliably stop in time for stationary objects. In the moving obstacle test the car drifted slightly closer and further away from the obstacle. However, this is not seen as a problem because the application of the car is to cruise on a highway and it will not matter if the car deviates a small distance from the preset distance as long as the preset distance is a lot greater than the deviation. Because of these successful tests, the argument could be made that the car in fact can avoid obstacle in a safe way but not drive around them. However, implementation of this on a full scale is problematic. Ultrasonic sensors have a limited range of the distances it can measure. The sensors used in this project has a max range of 5 meters which works fine on a small scale with low speeds, but in a full scale situation with much higher speeds the brake distance could be up to 100 meters depending on the car and road conditions. To be able to stop for a stationary object at those speed you need a sensor that can detect the obstacle from more than 100 meters. Ultrasonic sensors are not capable of that.

4.2 Conclusions

In conclusion ultrasonic sensor is an excellent and cheap way to measure distances and will definitely keep being incorporated in most self-driving cars, but will only be used when the distances to measure are short. Even though in theory it is possible to only use ultrasonic sensors for the cars steering system to reduce costs it is not justifiable to do so when human life is at risk. Autonomous car of the future will probably go the opposite direction by adding even more sensors to be able to compare the inputs from the different sensors and therefore get more accurate data.

5 RECOMMENDATIONS AND FUTURE WORK

5.1 Recommendations and future work

To move forward with this project there are several ways to go. You can add more ultrasonic sensors in order to give the car more information about its environment and develop software to make car drive around obstacles. You can also scale up the project by working with a bigger and faster RC-car and investigate how well the control system works for speeds up to 30 km/h. You can also expand on the idea that more kinds of sensors are better and add a laser rangefinder or a camera and examine how these new sensors can improve the safety and performance of the car.

REFERENCES

Arduino, “Arduino Uno”, https://www.arduino.cc/en/main/arduinoBoardUno , read 2016-05-23.

Eckel, *“NewPing Library For Arduino”, Tim Eckel,* http://playground.arduino.cc/Code/NewPing *, read 2016-05-08.*

Fritizing, http://fritzing.org/home/, created 2016.

Gantt, *“Triggering a servo using a HC-04 distance sensor and an Arduino”,* http://www.themakersworkbench.com/tutorial/triggering-servo-using-hc-sr04-distance-sensor-and-arduino *,* read 2016-05-23

Glad och Ljung, *“Reglerteknik”, Grundläggande teori, Torkel Glad och Lennart Ljung, 2014.*

Google, *“Google Self-Driving Car Project”,* https://www.google.com/selfdrivingcar/how/*, read 2016-05-08*

Google Presentation, https://www.google.com/slides/about/ , created 2016

Johansson, “*Elektroteknik”, Hans Johansson, 2013.*

Robotics in schools, http://www.roboticsinschools.com/pwm.html , read 2016-05-23

Teach Engineering *“LESSON: How Does an Ultrasonic Sensor Work?”,* https://www.teachengineering.org/lessons/view/umo_sensorswork_lesson06 *, read 2016-05-08*

Volvo Car Group, https://www.media.volvocars.com/global/en-gb/media/pressreleases/158276/volvo-cars-presents-a-unique-solution-for-integrating-self-driving-cars-into-real-traffic , read 2016-05-16

Wikipedia, ”*PID controller”,* https://en.wikipedia.org/wiki/PID_controller , read 2016-05-08

Young and Freedman, *“University Physics Thirteen Edition”, Hugh D. Young and Roger A. Freedman, 2014.*

TRITA MMKB 2016:35 MDAB096