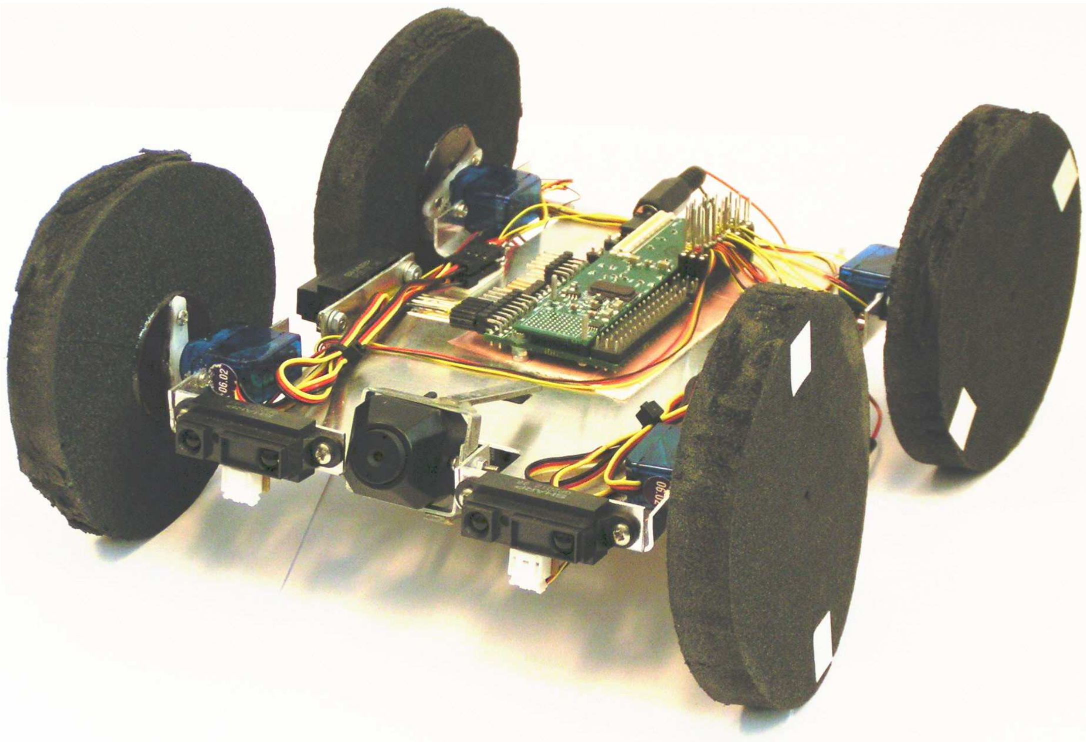
Path Finding Autonomous Car, Designed for Room Mapping



Group members of IAS10-1032d

Kristian Borup Pedersen
Thore Svejgaard Wienike

Fall 2006 - Spring 2007

Aalborg University
Institute of Electronic Systems
Department of Control Engineering

Institute of Electronic Systems

Department of Control Engineering

Aalborg University

TITLE:

Path Finding Autonomous Car, Designed for Room Mapping

PROJECT PERIOD:

IAS9-10,
September 4th, 2006 -
August 13th, 2007

PROJECT GROUP:

IAS10-1032d

PROJECT PARTICIPANTS:

Kristian Borup Pedersen
Thore Svejgaard Wienike

SUPERVISOR:

Anders la Cour-Harbo

COPIES: 4

NUMBER OF PAGES: 148

ATTACHMENT: 1 CD-ROM

ABSTRACT:

Abstract:
This project deals with the design and construction of an autonomous car able to video map an unknown room. The main issues are analysis of the optimal way to dimension and map a room, design and construction of the car, and designing a controller for the car.
The analysis is based on finding the best suited robot for mapping the room, and how to optimize the use of a spy cam in accordance to making a complete video map of the room.
The car is designed to weight less that 500 g and to be able to estimate the dimensions of the room. In the design of the car a tilting device for the camera has been made to get full use of the cameras vertical view angle.
Two controllers for the car have been designed. One model-based and one nonmodel-based. Both use the classic control theory, but two different design methods have been used. Root-Locus for the model-based and Ziegler-Nichols for the nonmodel-based controller. Their performances have been compared and the nonmodel-based controller had the best performance of the two.
The nonmodel-based controller can only be used for simulations though, while the model based controller can be used for implementation.

Preface

This report, dealing with autonomous robotics, is documentation of the work of group 1032d, doing a master thesis on Intelligent Autonomous System in control engineeringat Aalborg University. The report is written as a part of the Civil Engineer education (M.Sc.). The project work took place from September 4th 2006 to August 13th 2007, with Anders la Cour-Harbo as supervisor.

The report is divided in two parts; the main report and the appendix. The main report has six chapters: Chapter one describes the project and outline the problem formulation. Chapter two analyse our expectations to the project and the car, resulting in a specification of requirements and an accepttest specification. Chapter three discusses the hardware system and its components. Chapter four discusses the development of a controller. Chapter five and six ends the main report with a discussion and a conclusion of the project.

Citations throughout the report are indicated by author, year and optional page or chapter, e.g. [Servey & Beichner, 2000, page 194].

The enclosed CD contains the report in PDF, the Matlab source code, and datasheets included in the literature list.

Kristian Borup Pedersen

Thore Svejgaard Wienike

Contents

Chapter 1

Introduction

Contents

In everyday life, the use of robots is increasing. Both consumers and the industry call for this evolution, which also increase the demand for more advanced robots. To meet these demands, large investments, in developing new more advanced robots, are made [The Danish Ministry of Science, Technology and Developement, 2006].

The robots developed today are, by many, considered to have some sort of intelligence. For this to be true the robots will, at least, have to be autonomous and have cognitive capabilities.

This project is based on developing an autonomous robot. The main purpose is to make a video map of a room. The robot developed is a car, which is designed and constructed specifically for use in this project.

1.1 Motivation

What do we do if we want to see places we cannot, or will not, go to? Places that may be dangerous for humans to enter or places where other humans are causing a threat?

One thing to do is to send a robot with an on-board camera to film the place. This robot can be controlled remotely or maybe even better, drive around autonomously mapping the area in question.

The motivation for this project is based on thoughts such as these. More specifically an example scenario of such a situation is used to help define the perspective of the project.

Association for Unmanned Vehicle Systems International (AUVSI) has a competition in creating autonomous vehicles. And for a part of this competition three example missions are created. One, a hostage rescue, two, a nuclear power plant melt down, and three, a biological disaster. The competition is called International Aereal Robots Competiotion (IARC). The idea and rules of IARC will be used in this project.

1.2 Project Survey

This section is divided into two parts, the first describing IARC, and the second describing the main topics of the project in relation to IARC.

1.2.1 International Aerial Robotics Competition

IARC is a competition in designing and constructing autonomous vehicles that are able to sense the surrounding environment. It is divided into four levels of completion. The first two levels are carried out by en autonomous helicopter, carrying the car from this project. Level one and the first part of level two are illustrated in the left part of Figure 1.1. The helicopter has to fly from its start position to a cluster of buildings. This is a flight of approximately three kilometers. When the buildings are located it has to find a specific building marked with the competition logo.

The final part of level two to prime level three is shown in the right side in Figure 1.1. Once the building is found, the car has to be send into a room through a window in the building in order to commence its mapping procedure.

For the car to be eligible for the competition there are rules and boundaries

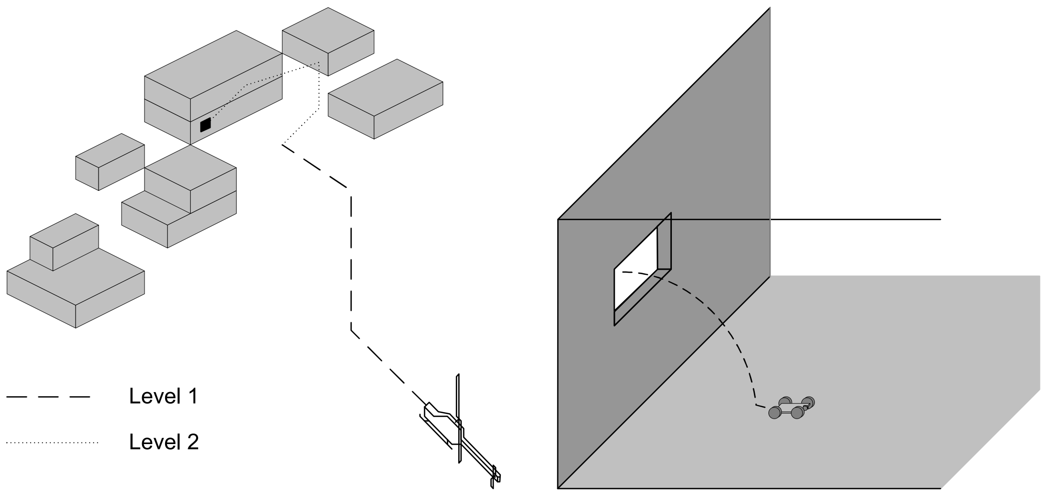


FIGURE 1.1: To complete level one, the autonomous helicopter has to fly from the ground station to the town. Within the town it has to find a specific, marked, house to complete level two. The last part of level two is sending the car into the house, through a window. Level three is completed when the entire room is mapped and either video or pictures of the entire room is send back to the ground station.

which it will have to comply with. It must

1. be fully autonomous.
2. not be independent of the helicopter to complete the competition.
3. be launched through a window from the ground or air under command of the helicopter.
4. fly or be carried the 3 km to the cluster of buildings.
5. be started before the helicopter is converted to automatic control.
6. be able to map the room in less than 15 minutes.

The third level has three different scenarios, the first mission example is "Hostage Rescue", the second mission example is "Nuclear Disaster", and the third and last is "Biological Emergency".[1] This is the level the car is participating in. In "Hostage Rescue" the car has to provide pictures of the hostages and their captors as fast as possible. The "Nuclear Disaster" mission is about a nuclear reactor that has exploded, and pictures of the position of gauges and switches of the main control panel is wanted, to see if there is an explanation on why the reactor exploded. "Biological Emergency" is about an excavation of a mausoleum, where the archaeologists got ill and died. To get control over the epidemic pictures of inscriptions on the walls inside the mausoleum is wanted.

1.2.2 Main Topics in Relation to IARC

The main topics of this project are path planning, hardware design and controller design. The controller design includes simulations of the controller and the behavior of the car in a room.

Path Planning

The path planing are part of what makes the robot autonomous. The path plan should describe the wanted behavior of the robot. The actual behavior will be based on the robot making a video map of a room in either of the three scenarios. That is, getting a clear view of the walls and whatever, and whoever are inside the room. To do this the robot will need to learn the size

[1] The competition rules can be found on the homepage: http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/200xCollegiateRules.html

and shape of the room and use these to determine a way to get a video map of the entire room.

The only thing known about the room is that it should be an office-like room, according to IARC. This means that, besides being of unknown size and shape, it can contain other obstacles as well. These can for instance be doors, furniture, and stairs leading upwards and/or downwards. Since it is an office-like room it is assumed that the floor is a plain even surface, and the walls are high enough for humans to walk upright. To have a useful room it has to have a certain size, therefore it is chosen that the size is not less than $2 \times 2\ m$. Furthermore most rooms are rectangular rooms or the like with 90° open and closed corners.

Hardware Design

One of the aspects in IARC is that the robot is a subvehicle. This means that it has to be transported by another autonomous vehicle, an aerial robot. This calls for considerations of the robot weight and size in addition to how it has to move around and how it shall make and transmit the video map to the other vehicle.

As shown in right side of Figure 1.1, the subvehicle is dropped through a window, which adds considerations of sustaining hitting the ground from the height of a window. We assume from previous work in the Unmaned Aerial Vehicle lab that the robot is dropped from a rail under the helicopter, and can impact the floor in different distances from the wall with the window, as shown in Figure 1.2.

Controller design

The path algorithms are part of what makes the robot autonomous, and the foundation of the robots movement. A path using feedback control will probably work well because it enables following a wall at a certain distance. This can be used when determining the size and shape of the room. Feed forward might be used to avoid obstacles due to its simple structure. Using a controller opens up for the possibility to see if it can compensate for the errors the feed forward algorithms introduces. When the controller is designed its performance will be simulated, to see how well it performs, and how it will react on different obstacles.

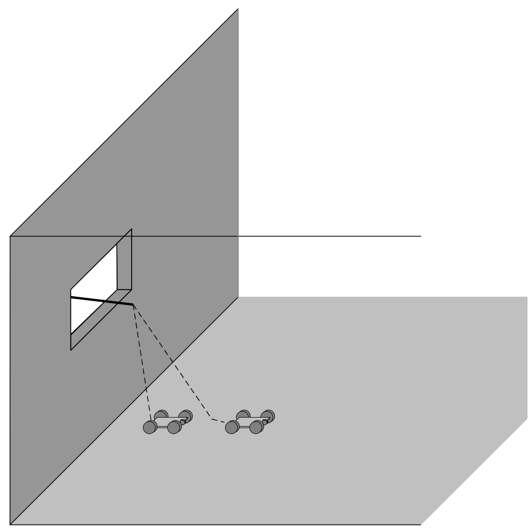


FIGURE 1.2: The robot is dropped from the helicopter via a rail through the window.

1.3 Project Delimitation

The limitations are split into two parts, the first concerning the robot, and the second concerning the room. There are two types of limitations in this project, the kind of robot considered, and of the room the robot is entering.

1.3.1 Robot Chosen

The chosen type of the subvehicle to complete level three in IARC is a car.The car is believed to be the best way to make a video map with. An alternative is a hovering robot that needs constant control to stay at the same point. The picture quality is assumed to be better from a car that stays at the same point, than from an aerial robot that relies more on precise mechanics, electronics and sensors to keep it stable at all time. A condition for using a car is that what needs to be filmed can be filmed from ground height (up-tilt).

1.3.2 Room limitations

The chosen scenario is "Hostage Rescue"[2].

According to the rules of IARC the car is to be send into a room through a

[2]The description of the scenarios can be found on the web page: http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/200xCollegiateRules.html

window. This means that the car will land in a random spot on the floor, pointing in a random direction, and that it is possible it turns upside down. The only thing known about the room is that it is in a house.

This has to include the possible obstacles that can appear. To be able to make a clear distinction between the obstacles that can occur in the room and the room itself, limitations of obstacles and assumptions for the room is made. The two issues mentioned, obstacles and assumption, are described in the following two sections.

Room Obstacles

Most furniture that occurs in a room has legs of a certain size. The two obstacles considered are doors and furniture legs.

The doors are considered to have two positions only, either closed or fully opened, in which case an open door can be considered as a gap in the wall and a closed door can be considered as part of the wall.

Furniture legs can be from tables, chairs, sofas etc., which means that there will always be four or more legs from each piece of furniture. Moreover, all legs are considered to be taller than the height of the car, so that the car can drive beneath the furniture.

Room Assumptions

The room is within a house and as such it can only assume certain dimensions. As mentioned earlier it is an office-like room it is assumed that the floor is a plain even surface, and the walls are high enough for humans to walk upright. To have a useful room it has to have a certain size, therefore it is chosen that the size is not less than 2×2 m. Furthermore most rooms are rectangular rooms or the like with 90° open and closed corners. We chose the following definitions for the test environment along with the sizes of doors and obstacles.

Shape: The room will have nothing but 90° corners. These can be both open and closed.

Walls: All walls are each between two and six meters long and the height at least 2.50 m, so that people are able to walk around upright.

Furniture legs: The dimensions of these are approximately 5×5 cm and the height is more than the height of the car.

Doors: In Denmark the standard door width is 82.5 *cm*, which therefore will be the width of the doors in the test environment.

In Figure 1.3 some examples of different rooms are given, all of which the car should be able to map.

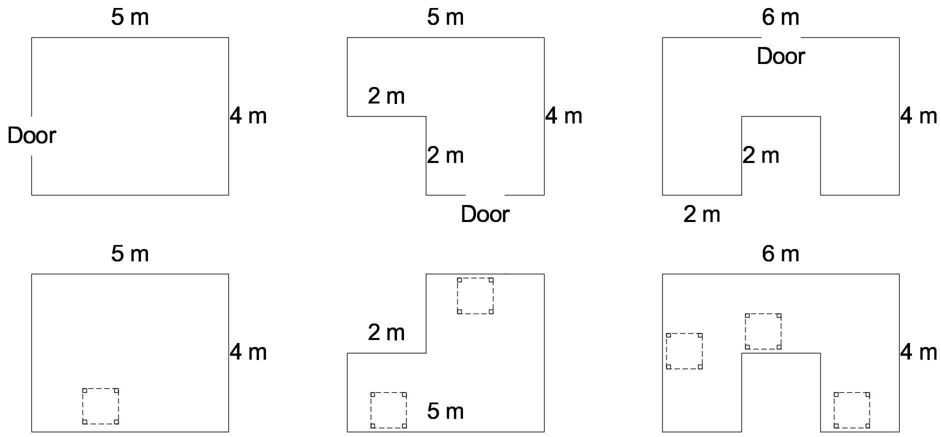


FIGURE 1.3: The mapping environments, with an example of the two types of obstacles considered, door and table.

1.4 Problem Formulation

Taking the three aspects in the previous section into consideration, the project's main problem can be expressed:

- *How can we design and construct an autonomous car able to send a video map of an arbitrary room to an autonomous helicopter, without the use of image recognition and with the extra aspects that the car has to be light enough for the helicopter to carry it and robust enough in its physical construction to survive the impact with the floor after being passed through an open window.*

Chapter 2

Analysis

Contents

The analysis starts with putting the project in perspective to IARC and the delimitations chosen, and from that it defines the expectations for the project.

Secondly the analysis is written with the perspective of the car and what it is expected to be able to do, and what behavior it is expected to show while within the room. In this there will be considerations on what the hardware and what the software should do and what the interfaces between the two are.

The requirement specification will be developed from the considerations concerning the car. To define whether a requirement is met all requirements have to be testable.

The project expectations will be reflected in the accept test specification that ends this chapter.

2.1 Project Expectations

The problem formulation states that a car has to be designed and constructed, to make a video map of an unknown room. But before we look at what the car should be able to do, we will look at why the choice of mapping vehicle is a car.

As this project uses IARC level three the type of subvehicle will have to be chosen. To do this three principles are analysed, a fixed subvehicle, which could be part of the main vehicle or separate, a ground based subvehicle, which has to be carried by the main vehicle to the room, and an aereal subvehicle, which may either follow the main vehicle or be carried by it to the room.

2.1.1 Fixed Position

The first thing we analyse is the fixed subvehicle. That it is fixed means that it is unable to move around and it will therefore either be at the end of a rail attached to the helicopter or be dropped in by the helicopter. In both cases it have to make the video map from the place it is inserted. This method is interesting as it is a simple solution, but as this analysis will show, does it have certain issues which can be difficult to overcome.

The essence of the competition is that the subvehicle has to autonomously map an unknown room. This means that it has to film everything within it, that is getting a view good enough to distinguish persons from one another, in reference to IARC level three, hostage rescue, described in section 1.2.1. Looking at a rectangular room and a subvehicle which is fixed to the position it landed when send in by the helicopter, as shown in the left side illustration in Figure 2.1, will the subvehicle be positioned up to 1.5 m from the wall. The 1.5 m is an estimation based on the idea of dropping the subvehicle from a rail attached to the helicopter.

Two questions rises to determine whether the method of using a fixed positioned subvehicle is usable,

- what is the range of the camera and is a zoom ability needed?
- and can it handle a room of unknown shape?

Considering the first question, it is likely that a zoom ability is needed. As shown in the right side illustration in Figure 2.1, and considering that the room has unknown width and length, the subvehicle might not be able to cover the entire room without the ability to zoom.

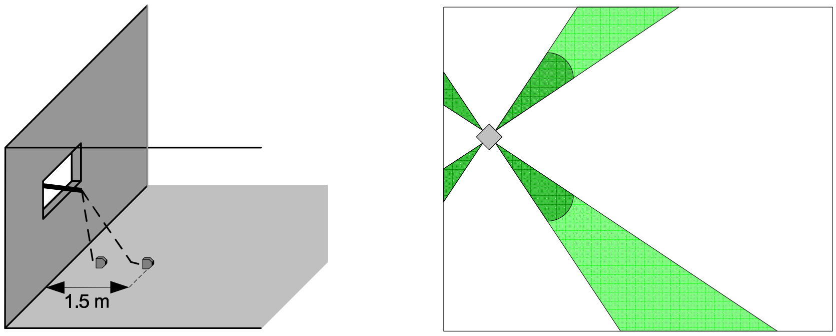


FIGURE 2.1: Left: Possible positions of the subvehicle that are unable to move around. Right: an example of a room where the filming device on the subvehicle need a zoom ability.

Considering the second question, it becomes clear after looking at two different room shapes that the subvehicle with fixed position cannot map rooms with one or more open corners. Two examples of room shapes it is unable to make a complete video map of is illustrated in Figure 2.2.

The room to the left in the figure shows two positions of the car. The one where the light gray cones originates from is a position where the car can make a complete video map, without having the ability to move around. The position where the darker gray cones originates from the car is a position where the car is unable to make a complete video map of the room.

Extending the room to have two open corners. The car is placed in the position where it, in the room with one open corner, was able to make a complete video map of the room. The room illustrated on the right in Figure 2.2 shows, by use of the dark gray cones, that the car in the that position is incapable of making a complete video map of the room, without having the ability to move. Taking another look at the illustration on the right, suggests that the car, in this shape of room, cannot have a fixed position and make a complete video map at the same time.

If other obstacles are considered, such as furniture legs, will the video map affected in the same way as it would in rooms with open corners. Depending on where the subvehicle is place as related to the object, it could be unable to cover what is behind the object.

2.1.2 A Car

As the subvehicle with fixed position proved to be unable to make a video map of rooms which are not rectangular, it has been chosen that this project

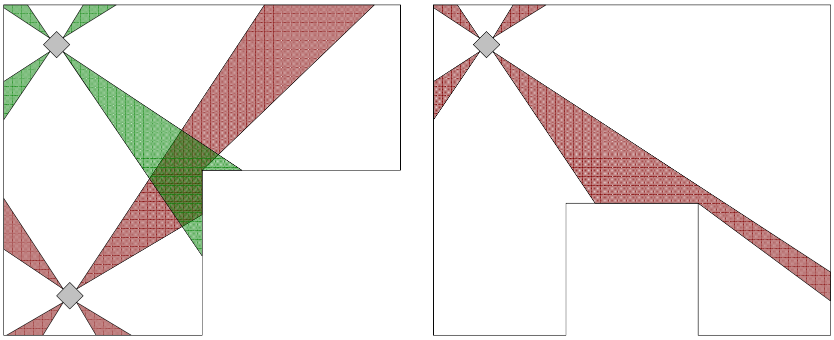


FIGURE 2.2: Whether the fixed subvehicle can make a complete video map of the room on the left is dependent on its position. In the room on the right the cannot in any case, as long as it has a fixed position, make a complete video map of the room. The light gray cones defines a position where the subvehicle can map the room, and the darker gray cones indicates that the room cannot be mapped from that position

evolves around designing and constructing a car, as the problem formulation states.

In table 2.1 are the advantages and disadvantages of choosing a car instead of an aereal subvehicle stated.

	Advantages	Disadvantages
1	Simpler Movement Control	Robust Physical Construction Required
2	Simpler Design	Filming From Floor Height
3	Is Easier Transported	
4	Only distance Sensors Needed	
5	Can Sustain Impacts with the Surrounding Environment	

TABLE 2.1: The advantages and disadvantages when choosing to make a ground based instead of an aereal subvehicle.

- **Advantages:**

Simpler Movement Control. To move the subvehicle from one position to another is simpler for a car, as there are lesser control and fewer variables to consider. And when reaching the new position the car is easier to stabilise as it is on the ground it can be idle, while the aereal subvehicle has to have its controllers active all the time.

Simpler Design. As the movement control is simpler for the car, the design will be simpler, as less components and less control will be needed.

Is Easier Transported. Both a car and another aerial subvehicle would have to be carried by the main aereal vehicle. But considering a helicopter for the subvehicle, it would be easier to mount a car underneath the main helicopter.

Only Distance Sensors Needed. A car will only need distance sensors to orientate, whereas an aereal subvehicle would need for instance a gyro.

Can sustain Impacts with the Surrounding Environment. As the car has to be able to sustain being dropped into the room by the main helicopter, it will have some physical robustness. A helicopter does not have the same physical robustness, and will crash if it hits walls or objects, while within the room.

- **Disadvantages:**

Robust Physical Construction Required. The car will have to be placed on the floor in the room, which for this project means that it is send through the window and dropped to the floor. This means that the car's physical systems will have to survive hitting the floor.

Filming Form Floor Height. The car driving on the floor, which means that the filming will be done from floor height. This gives the car a disadvantage when compared to the aereal subvehicle. Whereas the filming can reach the same visual hight, the filming from the ground will do this with a larger angle and therefore has to be further away than the areal subvehicle. This will give a less clear recording at the visual height.

The advantages and disadvantages will be further discussed in section 2.2, concerning the expectations of the car.

Having decided that the subvehicle will be a car and before analysing precisely what abilities the car should have, an analysis of its wanted behavior is conducted.

2.1.3 Considerations on Car Movements

The car has to drive around making a video map of the entire room. For the car to know when it has mapped the entire room, it has to learn the size and shape of the room.

To get the room dimensions two methods, based on different capabilities, are considered. The difference in the two methods is the range of which the car is expected to measure the distance. In the long range measuring method the car is expected to be able to measure the distance to all walls in the room. In the short range measuring method the car is expected to have a limited distance censoring range and to depend on finding an following the walls.

But before considering how the car should move around the room, it will have to find out which way is up.

Determining Forward Direction

during its fall the car can flip and end up turning upside down. So for the car to drive forwards, filming in the intended direction, it has to determine which side is up and which is down.

Long Range Measuring

This method uses long range distance measuring. Where long range is defined as being able to measure the distance to all walls, in line of sight, independently of where in the room the car is placed. This does not necessarily require the car to have distance measuring devices pointing in all direction, as it might as well could have the ability to rotate and thereby get all angles.

If the room has one or more open corners the car may not have all the walls within line of sight. To compensate for this the car will have to make more than one measuring point, as illustrated on the left most room illustrations in Figure 2.3.

There are no limit defined for the length of the walls which means that the required range of the measuring devices is unknown and can therefore not be used as a design parameter or a testable requirement.

Short Range Measuring

The other way to find the room dimensions, when taking a limited measuring range into account, is to find a wall and drive alongside the walls until all dimensions are found. Doing this will require that the car can keep a constant

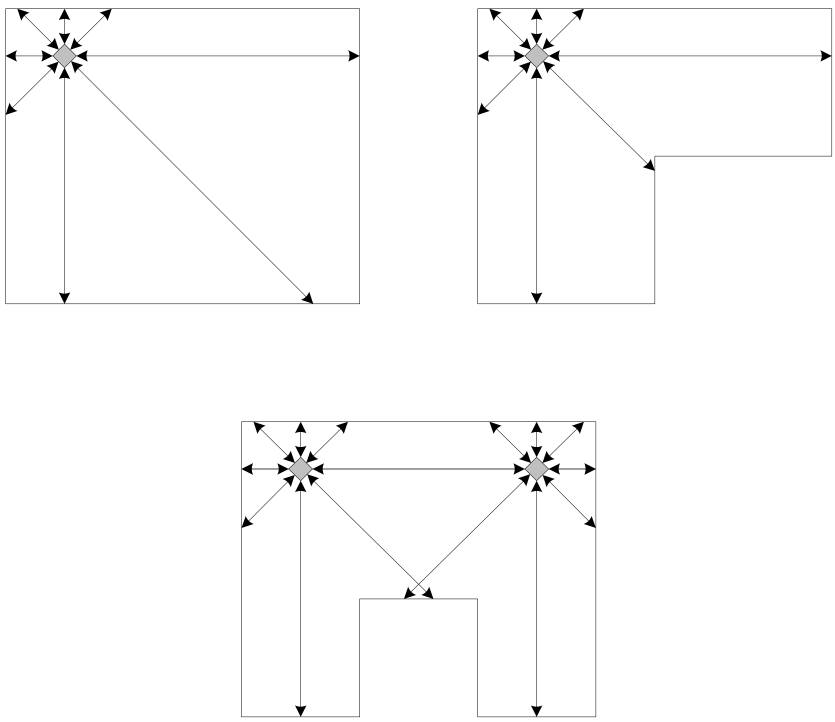


FIGURE 2.3: The measuring of the size and shape of the room with a car that can measure the distance to all walls within line of sight.

distance to the wall it is driving alongside. This method is illustrated in Figure 2.4.

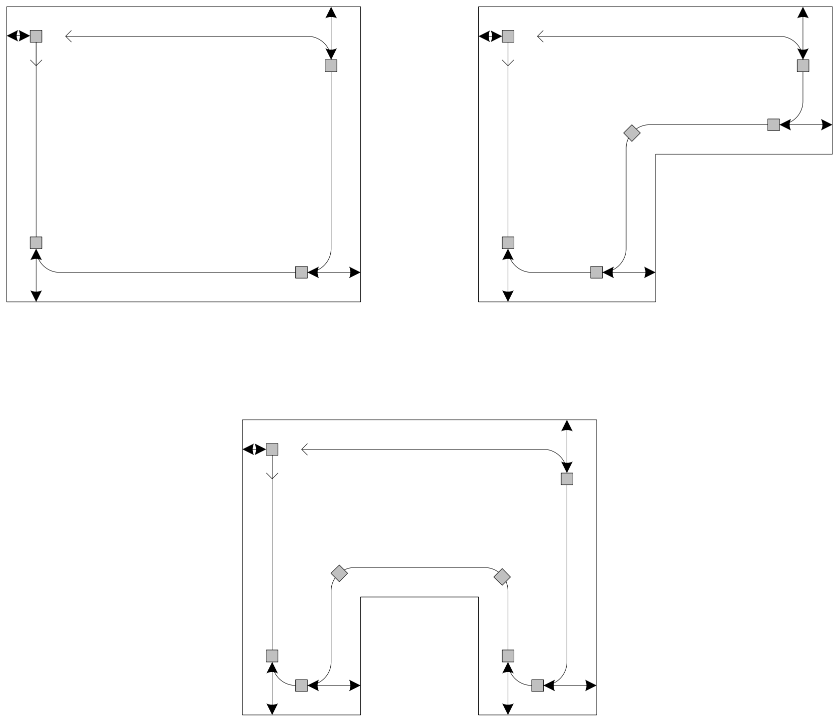


FIGURE 2.4: the car can find the room dimensions by driving alongside the walls.

In handling an unknown size room, the short range measuring method may take longer than the long range measuring method, but this method ensures that all wall lengths are covered. The precision of this method is also deemed higher, as a deviation in angle in the long range measuring method may result in wrong dimension calculations.

2.1.4 Making a Video Map

The next thing to consider is how well the short range measuring method is when considering that the car has to make a video map. Figure 2.5 shows the covered area in light gray, while the darker gray indicates the range and width of the filming.

In it self does the short range measuring method not cover the walls, and thereby nor persons standing against the wall, very well. and the center, or inner parts, of the rooms are not covered either.

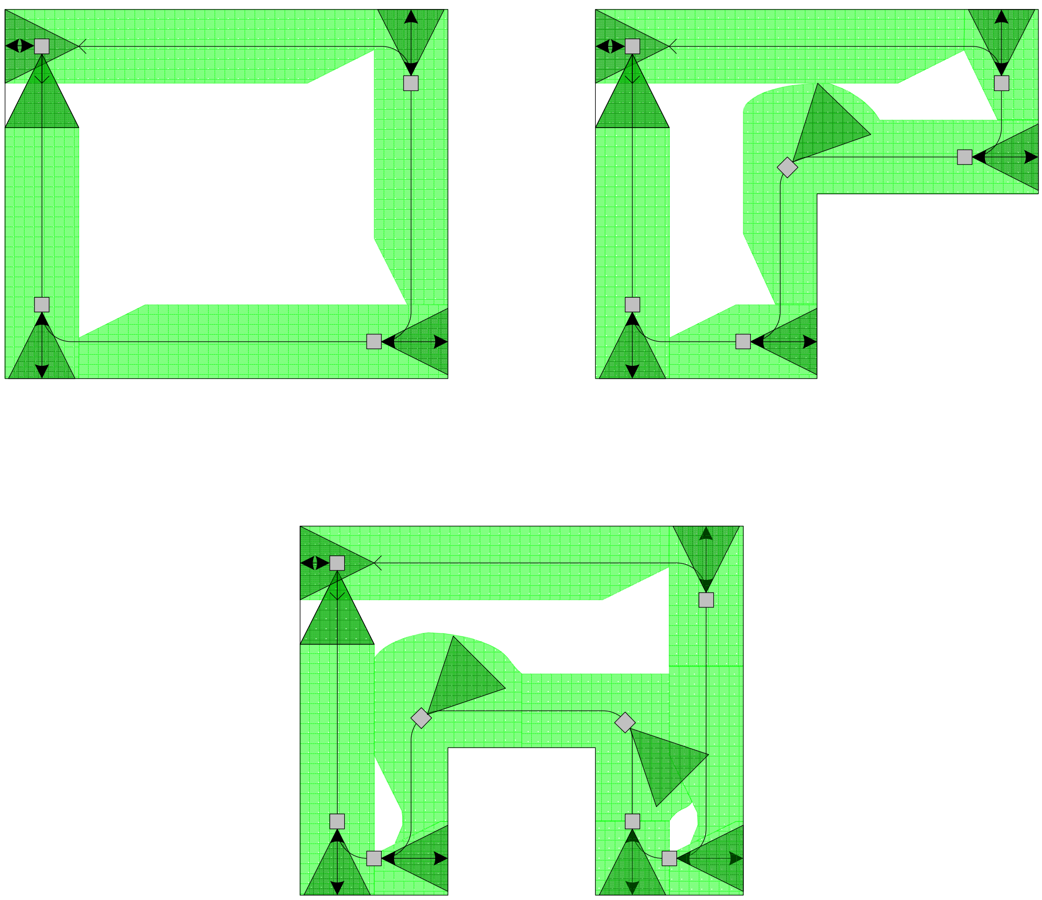


FIGURE 2.5: While the car is finding the dimensions of the room it also makes a video map of part of the room. Here shown for three different room shapes.

To analyse exactly what is covered a few definitions are needed.

Wall Coverage. The covering of the walls is in itself of no greater significance, but the covering people who is positioned up against the walls is. Therefore a definition of how the walls are best covered by the filming is defined. To get the best view of a wall the filming direction will have to be perpendicular to the wall.

Minimum Visual Height. The visual height is the vertical distance from the floor to the highest point that can be filmed. This upper limit is found with basis in the average height of a person, or a male, as the average height of males are more than the average height of females [Danes Grow Taller and Taller, 2003]. The height used is estimated from the height of the danish people [Danes Grow Taller and Taller, 2003]. In year 2000 the average height of a danish male was about 180 *cm*. There is furthermore a tendency, which indicates that the height is increasing over the years. When taking this tendency, and the fact that it is average heights, into consideration the minimum filming height is estimated to be 2 *m*. At this height the faces of the people in the room can be seen.

Covering Ranges. These are the maximum filming range, which is before the picture gets blurred due to the resolution, and the minimum filming range, which is defined as the horizontal distance from the filming device to where the visual height meats the Minimum Visual Height.

These three factors defines the coverage volume and are illustrated in Figure 2.6

Continuing with the room shapes in Figure 2.5, we look at how the coverage, during the dimensioning of the room, can be improved. There are two factors in mapping the room, covering the walls and covering the inside of the room.

Making Use of Sweeps

While driving alongside the walls it can be difficult to cover the walls, as the car will be too close to the wall it is driving alongside to get a clear view at the height wanted. It seems easier to get a view of the insides of the room if the car stops from time to time and makes a sweep. That is a rotation of 180° and back. This sweep should be performed a number of times depending on the car's coverage volume. This behavior is illustrated in Figure 2.7.

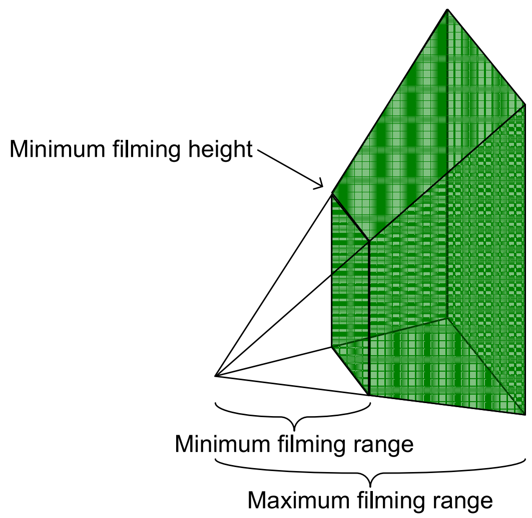


FIGURE 2.6: The coverage volume.

The percentage wise coverage of these sweeps will depend on the room size. So to cover the center of larger rooms and the walls an addition to this method have to be used.

Making Use of Rotation Points

Even with the use of sweeps does the car not cover the walls. Therefore an additional method is developed to do this. This method is based on making 360° rotations at a distance where the car can get an acceptable view of the walls. This position shall therefore be between the minimum filming range and the maximum filming range.

This method is developed to get an acceptable coverage of the walls, but as shown in Figure 2.8, can this method also be used to cover parts of the room interior as well.

As the distance measuring range will probably be lesser than the minimum filming range the car will have to drive to the rotation points "blindly." A way to compensate for this may be to have the car drive back to follow the walls every time it has made a rotation point.

Though this method also covers parts of the room interior, it does not cover it all and therefore does not ensure the complete video map. Therefore the method is expanded to have more rotation points added after the first set, which were covering the walls. The second set of rotation point should cover the interior of the room in question. How many rotation points that will be need is dependent on the size and shape of the room.

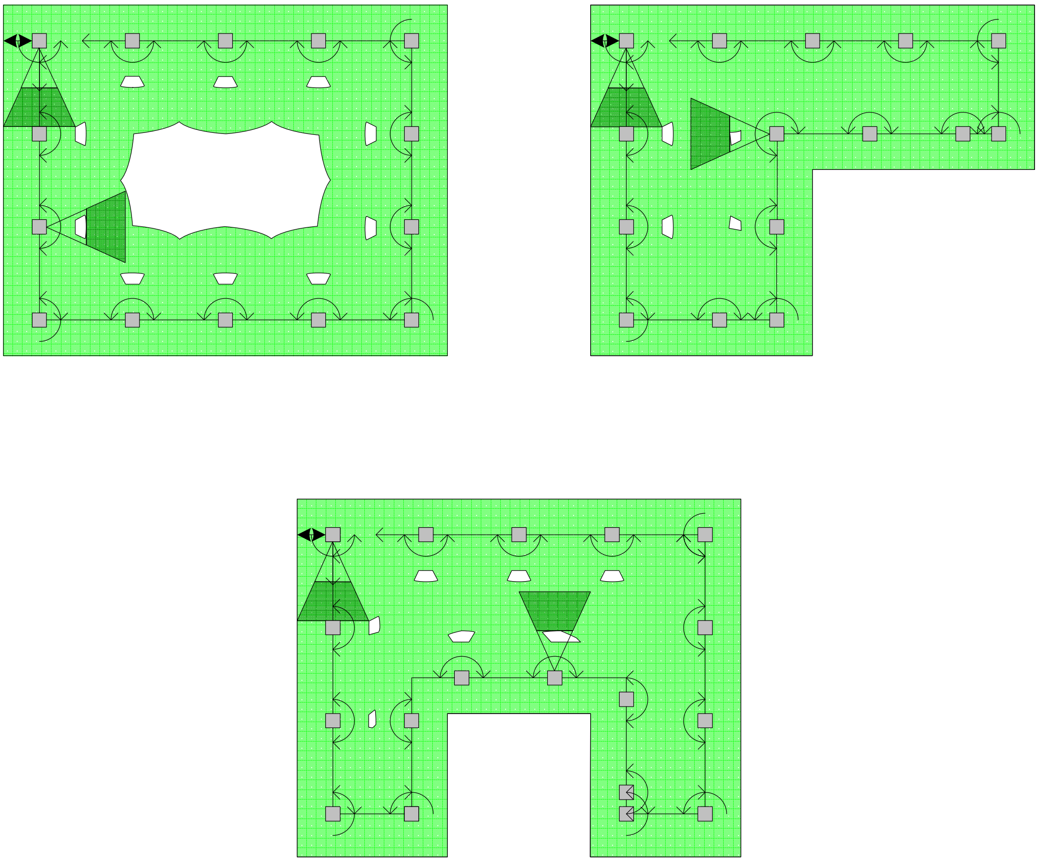


FIGURE 2.7: Covering the room while finding its dimensions can be improved by adding a number of sweeps depending on the car's coverage volume.

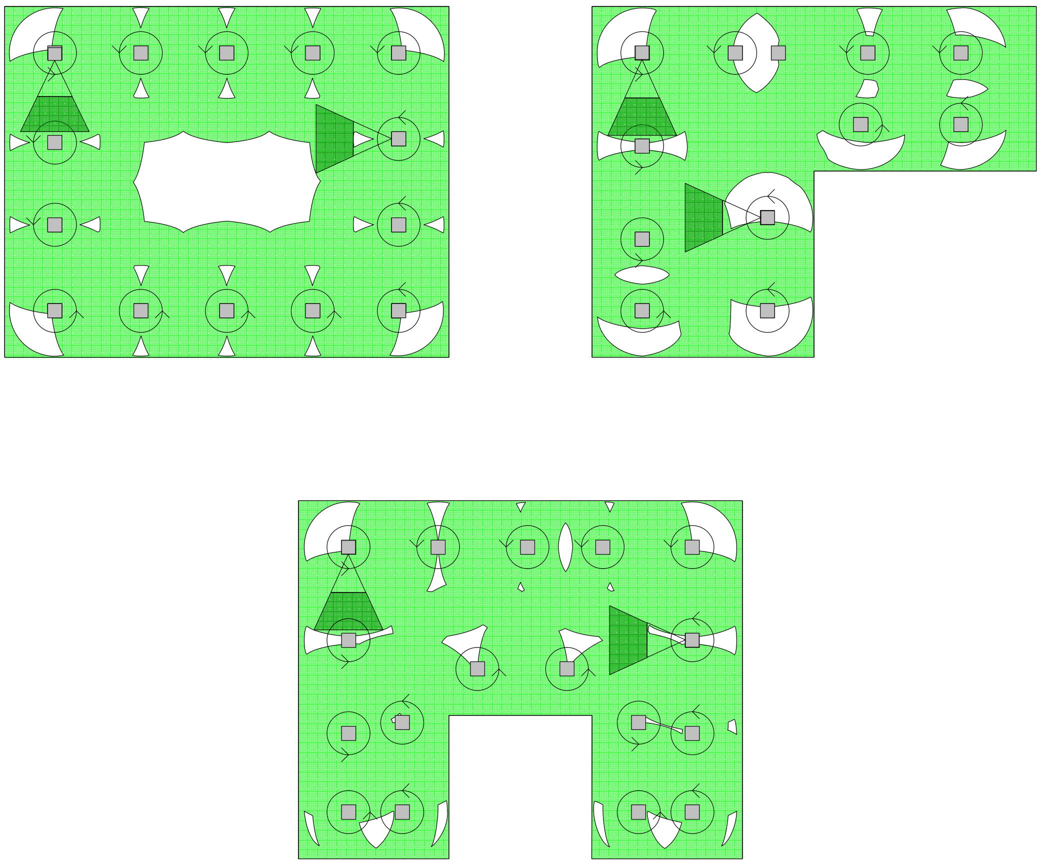


FIGURE 2.8: Making 360° rotations at certain spots in the room will ensure that the walls are covered.

The Mapping Methods Combined

Two methods have been analysed, the first was to film while finding the dimensions of the room by using sweeps inwards in the room, and the other was to find some points, when the room dimensions are known, in which the car shall rotate to get a clear view of the walls. These two methods combined is expected to give a complete video map of the room. Taking a closer look at the intersection of the methods, illustrated in Figure 2.9, reveals that the "rotation points" method contains most of the the "sweeps" method. It is therefore estimated that only one of the methods is needed, and the chosen method is "rotating points," as this covers the largest area, the walls, and can be used cover all of the interior of the rooms with its second set of rotation points.

This means that the car will first follow the walls all the way around the room to get the dimensions, during this it films the area alongside the walls. Secondly it will make rotation points to cover the walls. If there is still some parts of the room that are uncovered the car will thirdly make a another set of rotation points to cover this. This behavior is illustrated for three room shapes in Figure 2.10.

2.1.5 Obstacles

So far this analysis has looked at mapping a room with no open doors or obstacles, like furniture. Assuming that the car can handle encountering these two sorts of exceptions. Therefore it has been considered whether the handling of these exceptions compromises the making of the video map.

Open Doors: To avoid having the handling of open doors compromise the complete video map, it is expected that the car should return to its original drive path once an open door is detected. The expectation for this behavior is shown in Figure 2.11.

Objects: It is estimated that objects placed in the drive path or in the rotation points are the most troublesome, as these requires the car to deviate from its original driving pattern. But in spite of the deviations that might be introduced, the car should, due to its use of filming while rotating, still be able to make the complete video map. The expected behavior and how it affects the video map is shown in Figure 2.11.

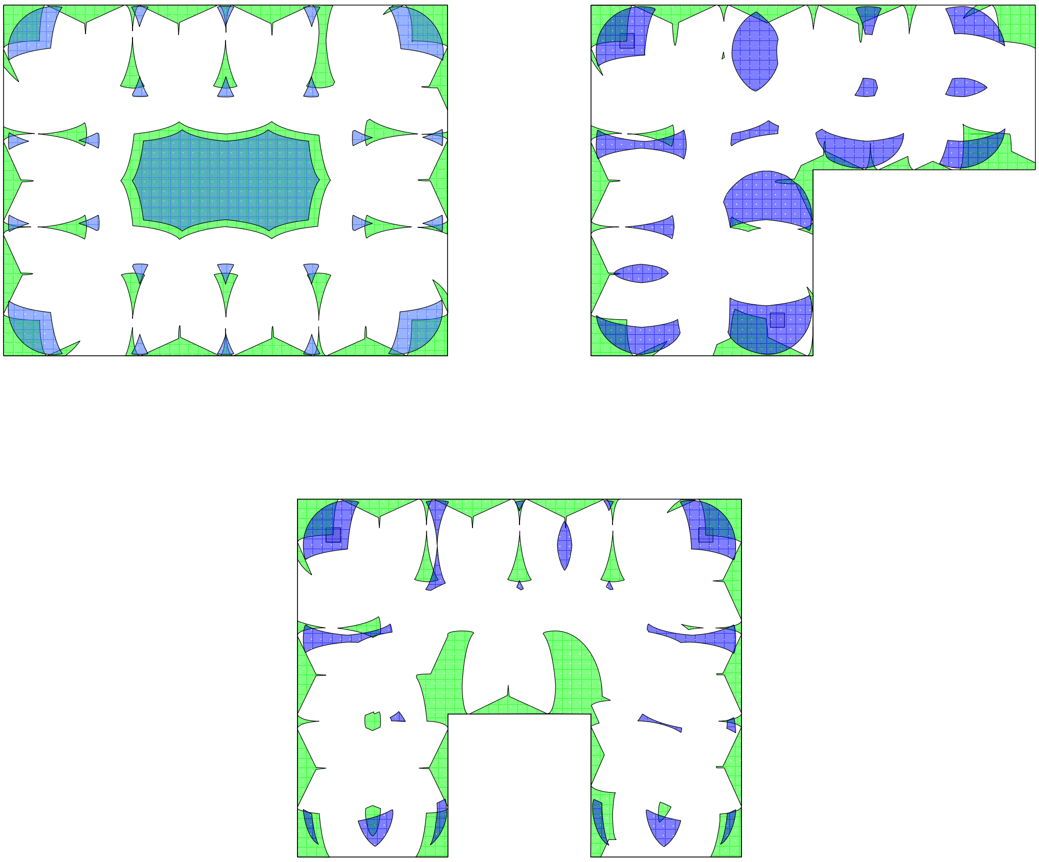


FIGURE 2.9: The white areas are the ones covered by both the "rotating points" and the "sweeps" methods. The light gray shows where the " sweeps" method is lacking and the darker gray shows where the "rotation point" method is lacking.

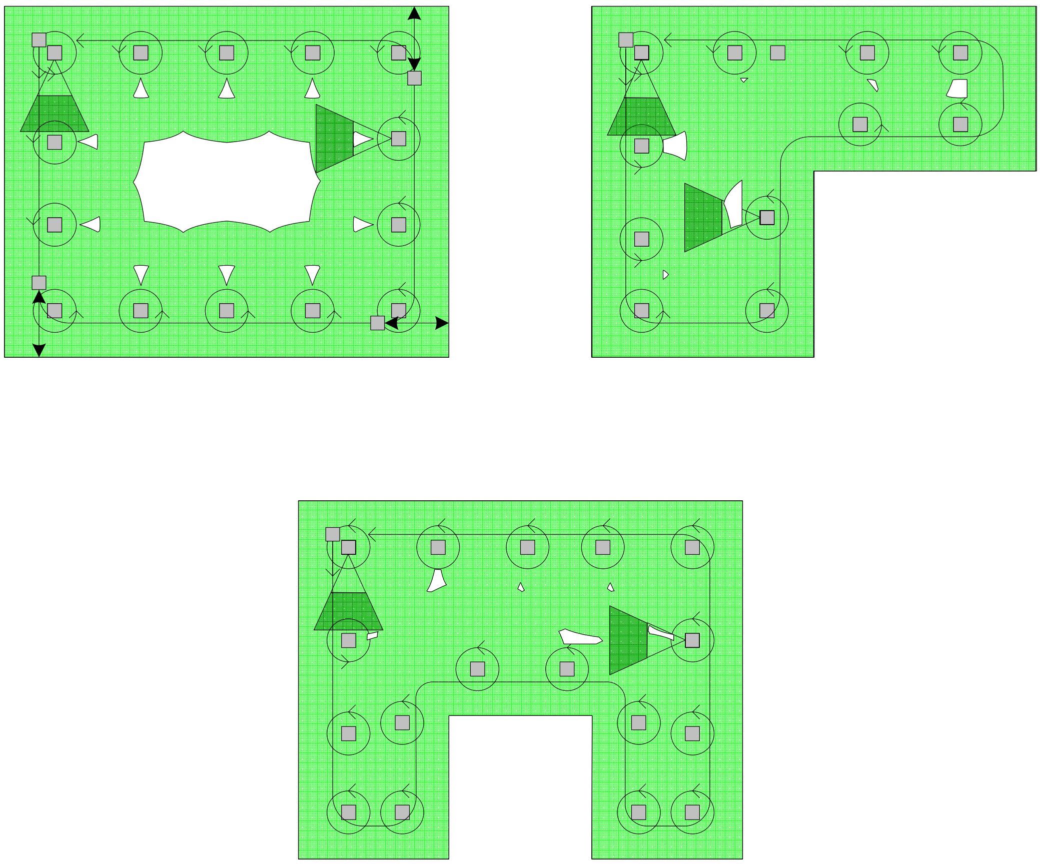


FIGURE 2.10: Using the three steps described in the text is expected make a complete video map of the room.

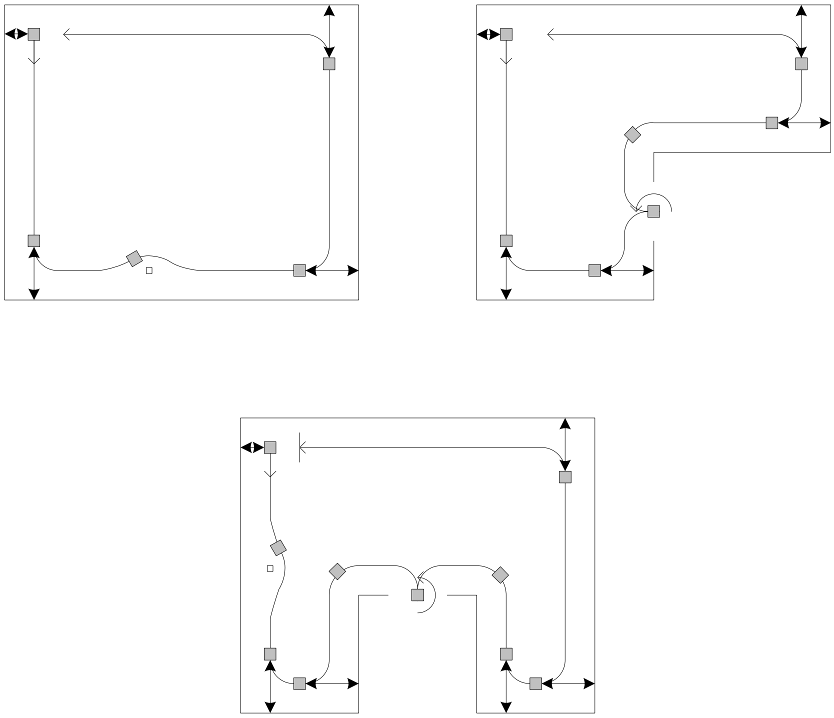


FIGURE 2.11: The handling of open doors and objects should not compromise making a complete video map of the room.

2.2 Car Analysis

In section 2.1, the wanted behavior of the car was analysed, with respect to making a complete video map of an arbitrary room in accordance to IARC. This section breaks this issue into smaller, more specific, issues, with the aspect of what capabilities the car should possess to have it show the wanted behavior.

2.2.1 Robustness

As stated in 1.4, the problem formulation, the car needs to survive being send through a window and dropped to the floor. The issue here is, whether the car can sustain landing on the floor. For the car to be able to sustain this, calculations on the energy affecting the car are made.

It is expected that the car is dropped vertically, which means that the car only accelerates vertically. Upon hitting the floor the kinetic energy, expressed in equation 2.1 [Servey & Beichner, 2000, page 194], affecting the car is dependent on its own mass and the velocity it has reached right before hitting the floor. This is the energy affecting the car suspension system. Figure 2.12 shows the car and the factors affecting the fall and hit.

$$E_{kin} = 0.5 \cdot M \cdot v_f^2 \tag{2.1}$$

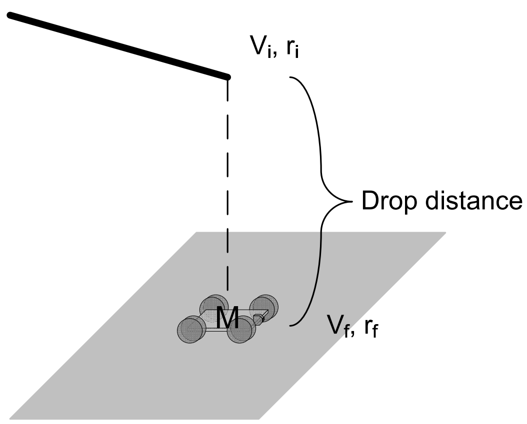


FIGURE 2.12: The car hits the floor after a one meter fall.

To find the kinetic energy the mass, M, of the car has to be known. The requirement for the maximum weight of the car is estimated from what the autonomous helicopter is assumed to be able to carry in addition to its own weight. We estimate the maximum allowed weight to be 0.5 kg. That is $M = 0.5\ kg$.

The velocity right before the car hits the ground, v_f, is found by use of equation 2.2 [Servey & Beichner, 2000, page 37], where r_i and r_f are the initial and final position of the car in reference to where the car is dropped from with the direction pointing downwards, v_i is the initial velocity, which is 0, and a is equivalent to the free-fall acceleration [Servey & Beichner, 2000, page 40], that is $a = 9.80\ ^m/_{s^2}$.

$$v_f^2 = v_i^2 + 2 \cdot a(r_f - r_i) \tag{2.2}$$

Substituting v_f^2 in the kinetic energy equation with 2.2 yields an expanded expression of the kinetic energy in equation 2.3, where all factors are known.

$$\begin{aligned} E_{kin} &= 0.5 \cdot M \cdot \left[v_i^2 + 2 \cdot a(r_f - r_i)\right] \\ E_{kin} &= 0.5 \cdot 0.5\ kg \cdot [0\ ^m/_s + 2 \cdot 9.80\ ^m/_{s^2}(1\ m - 0\ m)] = 4.9\ J \end{aligned} \tag{2.3}$$

This means that the car will have to be able to absorb 4.9 J from landing on the floor.

Another way to describe the effect of the impact when the car hits the floor, is to find the g-force affection. G-force is the deacceleration the car experiences when it from the velocity it has right before it hits the floor to it is brought to a total halt right after hitting the floor. In this, it is expected that the car will not bounce back, and thereby have the wheel lifted from the floor, once it hit the floor. The car's velocity right before it hits the floor is v_f from equation 2.2, $v_f = 4.4\ ^m/_s$. The time of the impact has not been tested, and as such is the collision time chosen to be 0.01 s, which is considered faster, than what the collision will actually be. As a result of the chosen collision time, will the deacceleration and resulting g-force affection can be found from equation 2.5.

$$deacceleration = \frac{v_f}{collision\ time} = \frac{4.4\ ^m/_s}{0.01\ s} = 440\ ^m/_{s^2} \tag{2.4}$$

$$\text{g-force} = \frac{deacceleration}{g} = \frac{440\ ^m/_{s^2}}{9.82\ ^m/_{s^2}} = 45\ g$$

2.2.2 Video Coverage

As mentioned in section 2.1, the car has a coverage volume. To ensure the car makes a video map of everything. This volume is dependent on the filming device used. There are two ranges which are of importance concerning the coverage volume, a minimum and a maximum filming range. The minimum filming range is when the visual height of the filming reaches 2 m, in accordance to 2.1.4, and the maximum filming range is where the image gets too blurred to distinguish the filmed persons' characteristics.

To ensure that there is a gain when making the sweeps and to decrease the number of rotations points will the maximum filming range have to be at least two times the minimum filming range. Choosing this relation is based on having a distance equal to the maximum filming range between each rotation point. The result of choosing this relation is illustrated in Figure 2.13.

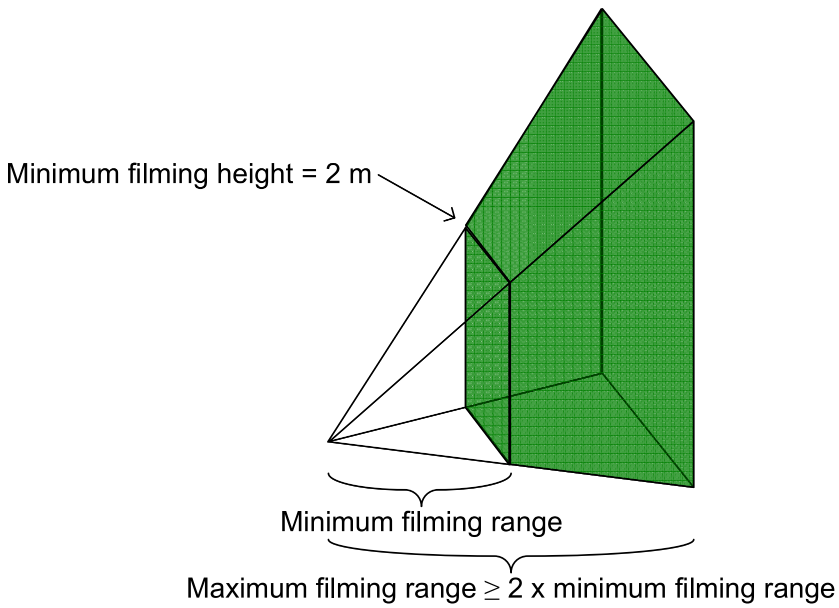


FIGURE 2.13: The limit of the short range is where the visual height reaches 2 m. Furthermore does the long range have to be more than twice the short range.

2.2.3 Navigation

The car needs the ability to navigate through the room without the use of image recognition. This means it will have to depend on measuring the distances to the walls in the room and when driving to a rotation point it will drive "blindly" through the room.

To avoid this to aspects have to be considered, which side of the car is up? And what should be the distance measuring range?

Floor Detection

As answer to the first question, which is related to the issue of which direction is forwards mentioned in 2.1.3, the distance to the floor from the bottom of the car has to be measured, and a clear distinction has to be possible.

Distance Measuring Range

To answer the second question, will the IARC scenario, hostage rescue, have come into consideration. Just as the minimum visual height of the filming is defined from the height of people, and the fact that the car has to film perpendicular to the walls, so does the car has drive a distance to the walls where it avoid people standing up against them.

The minimum distance the car should drive from the wall is therefore defined from the length of the peoples feet, which in metric units is 0.3048 m [Servey & Beichner, 2000, page 12], which is a conversion from the English unit ft. Taking into consideration that people have different size feet and that they probably will not be standing with their heels to the wall, changes the requirement for the minimum distance the car shall have to the walls, while driving alongside them, to 0.4 m.

2.2.4 Movement

Looking at the general concept of a car, it has some basic abilities when it comes to movement. Cars are able to drive forwards, backwards, and turn to either side. These four abilities is likewise considered for use in this project.

Drive Forwards

The forwards speed is only important to the extend that the car can make a video map of the room within 15 minutes, which is a limit set by IARC.

The precision of the forwards movement is of greater importance then, as a deviation from the intended direction propagates throughout the mapping. For instance, if the car deviates $5°$ from the intended direction, then in the smallest defined room, $2 \times 2\ m$, the car will have deviated approximately $30\ cm$, if the only error considered is when driving forwards. The issue with the deviation is not whether the car can find the dimensions of the room, as the car, still only considering the forwards driving, will have driven $361\ cm$ instead of $360\ cm$. The issue is whether the first rotation point, to film the walls, will be placed in a point where the visual height is too low to see the faces of the persons standing up against the walls.

To find a realistic maximum deviation allowed, the height of people is once again considered. In 2.1.4 the visual height is defined to be $2\ m$ derived from the tendencies of increasing heights of the Danish people [Danes Grow Taller and Taller, 2003]. Taking a closer look at this tendency and considering the range of ages where the largest increase appear. Table 2.2 shows the height average of Danish males in 1987, 1994, and 2000.

Unit	1987	1994	2000
[cm]	178	181	182

TABLE 2.2: The average height of Danish males. All values are estimates from a diagram in [Danes Grow Taller and Taller, 2003].

The tendency is not linear, but considering the values from 1994 and 2000 is estimated to give an acceptable estimate of the actual average height at present. An increase of one centimeter over six years result in an average high at present of $1.17\ cm \approx 1\ cm$. This means the average height in year 2007 is approximately $183\ cm$.

Considering the $2\ m$ minimum visual height, will the maximum allowed deviation from this be $200\ cm - 183\ cm = 17\ cm$. The exact requirement for how much the car can deviate from a rotation point on the floor is dependent on the filming angle. This means that the requirement is the result of 2.5.

$$maximum\ displacement\ [cm] = 17\ cm \cdot tan(filming\ angle) \tag{2.5}$$

Considering the example used earlier, with a square room with $2\ m$ walls, the maximum deviation angle, from the intended drive direction, is $2.7°$. increasing lengths of walls decreases the maximum allowed deviation angle.

Drive Backwards

This is the reverse of driving forwards and the backwards movement therefore has the same requirements for direction stability as the forwards motion has.

Turning

The ability to turn will make the car able to follow a non-linear path within the room. Usually the turn movement is an arch of a circle. That is, if the car needs to turn 90° it has turning radius defined by the sharpness of the turn and uses its forwards propulsion to make the turn. The turning motion will be used while the car is finding the dimensions of the room according to the considerations on the car movements in 2.1.3.

Another way for the car to change its driving direction is by rotating around its own axis. This ability will the car have to use at the rotation points.

Having the two ways of changing directions, we look at what deviations the car should be allowed to display while performing these actions. Both turning and rotating can have displacement and angle error, as illustrated in Figure 2.14.

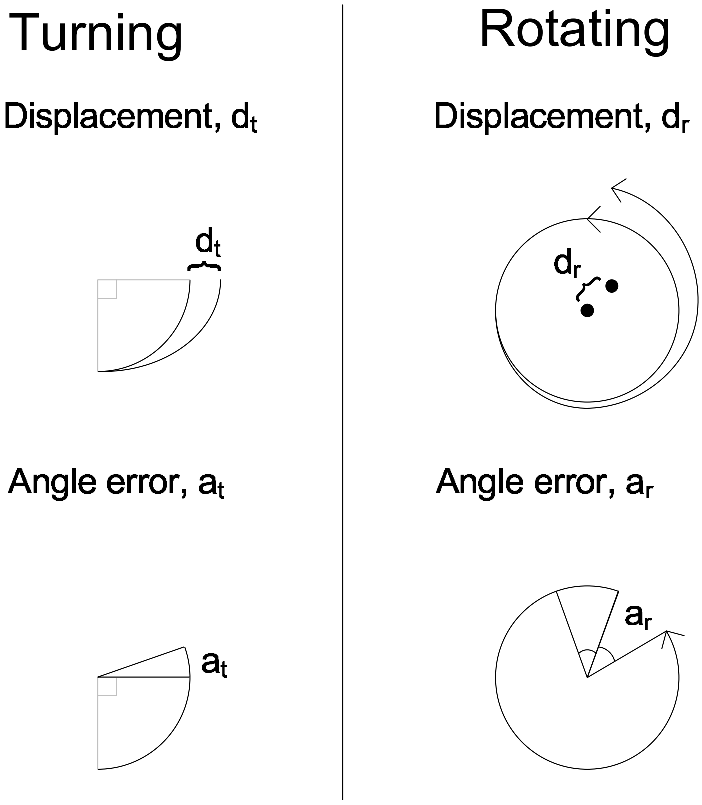


FIGURE 2.14: There are two errors that can occur for both turning and rotating.

Turns. A displacement after a turn means, as illustrated in Figure 2.14, that

the car is not at the intended position and has therefore not made an arch with a constant radius. The maximum displacement tolerated is related to the considerations in 2.2.4 on how much the car may deviate from its position at the first grid point. In a four corner room will this mean that the maximum displacement after a turn is approximately four centimeters, but only if the car has not deviation while driving forwards.
As with the displacement, does the angle error also relate to the error introduced while driving forwards. The forwards maximum deviation angle is 2.7°, at no angel error during turns. If the car displays an angle error while turning is this subtracted from the maximum deviation angle while driving forwards.

Rotations. Considering the maximum allowed displacement when rotating, it is assumed that the car is in the intended position. Therefore will the maximum displacement be the horisontal distance that corresponds to 17 cm.
To determine the maximum rotation error allowed a 360° rotation is considered, as it is in the rotation points the car will make the most use of rotations. The basic idea with the rotations is that the filming shall cover all 360°. This means that the maximum rotation angle error will be the horisontal filming angle.

The extend of the errors is illustrated in Figure 2.15.

2.2.5 Power Supply

The requirement for the power supply is given by IARC, as the rules states that no wired communication of power may take place between the autonomous helicopter and the car. This means that the car shall have its own power supply.

2.3 Specification of Requirements

2.3.1 Robustness

- The maximum allowed weight of the car is 0.5 kg.
- The car needs to be able to sustain hitting the floor after a fall of 1 m. The energy required for the car suspension to absorb is 4.9 J. The force of this fall will maximum be 45 g.

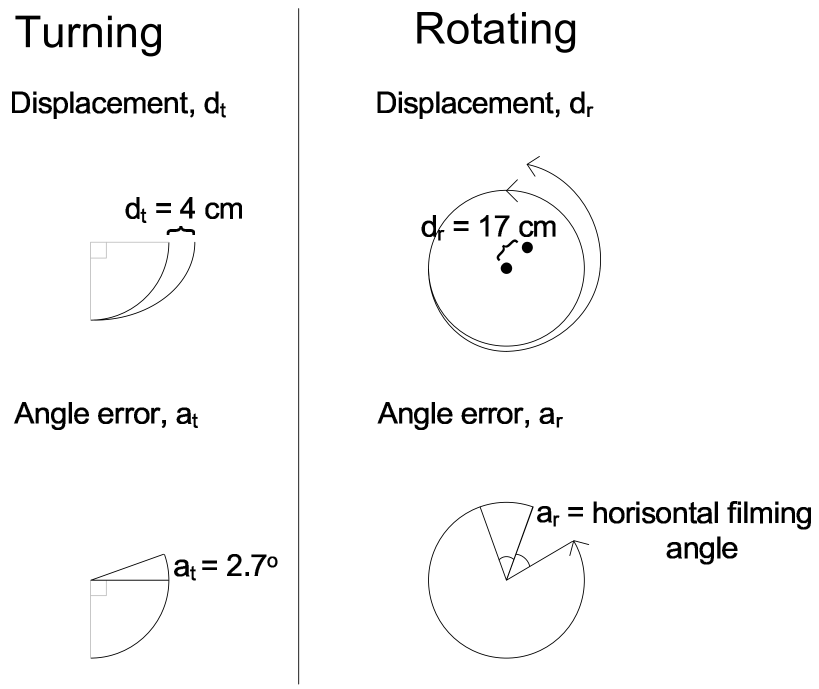


FIGURE 2.15: Maximum allowed displacement and angle errors while rotating and while turning.

2.3.2 Video Coverage

- Visual height at minimum filming distance: 2 m.
- maximum filming distance $\geq 2\times$ minimum filming distance.

2.3.3 Distance Measuring

- A clear distinction has to be possible to determine what side of the car is facing upwards.
- The car needs to have a minimum distance to the walls of 0.4 m. This means, the car need to have a maximum distance measuring range of more than 0.4 m.

2.3.4 Movement

The car has to be able to follow a linear path while driving alongside the walls. When encountering a wall or an obstacle, the car has to switch to a non-linear path.

Forwards. The maximum angle of which the car is allowed to deviate from the linear path is 2.7°.

Backwards. Same as for the forwards motion.

Turning. Maximum

- displacement: 4 *cm*.
- deviation angel: 2.7°.

Rotating. Maximum

- displacement: Horizontal distance that corresponds to 17 *cm* vertical distance. Depending on the vertical filming angle.
- deviation angel: The horizontal filming angle.

The deviations and displacements are based on a 2×2 *m* room. The maximum allowed errors for larger rooms is smaller.

2.3.5 Power Supply

- The car has to have its own power supply.

2.4 Accepttest Specification

The main goal of the project is to have the car make a video map of an unknown room. In section 2.1, Project Expectations, have the desired abilities of the car been analysed and from this analysis a wanted behavior has been found. To determine whether the car shows the right behavior in both practical tests and simulations, parameters for acceptance is set out.

The parameters reflects the theoretically designed path the car should follow to make a video map of the entire room. The actual behavior of the car is to be based on autonomous decisions and actions. This means the car is expected to decide the pattern best suited the situations by itself, though still in accordance to the wanted behavior.

Before the accepttest is specified a normal case scenario with no obstacles are outlined.

2.4.1 Normal Case Scenario

This scenario takes the car from the start point, where it has to be activated and carried by the autonomous helicopter, to the point where the car signals the helicopter that it has made a complete video map of the room.

IARC Level 1 and 2 briefly

1. Activation of the car (This is the only phase where human interaction is allowed).

2. Being carried by the helicopter to the town (This is step one of IARC, as described in 1.2.1)

3. Continuously carried by the helicopter to the room to be investigated (This is step two of IARC).

4. Lead through a window by a rail and dropped inside the room (IARC step two continued).

IARC Level 3

5. Landing on the floor, surviving the impact.

6. Find out whether it has the bottom sensor pointing upwards or downwards.

7. Drive forwards to find a wall.

8. Drive alongside the first wall found (distance measuring initiated).

9. Getting the dimensions of the room by driving alongside the walls (measuring wall lengths and counting corners).

10. Determining how many and where the first set of rotations points shall be (These are to film perpendicular to the walls).

11. Determining what parts of the room are unmapped and deciding where the second set of rotation points should be (These have to cover the rest of the room).

12. Signal the helicopter that the mapping is done.

2.4.2 Acceptance Parameters

The acceptance parameters are divided into two main groups,

- the parameters concerning the car, for instance start position,
- and the parameters concerning the room, for instance room shape.

Car Parameters

Robustness. For the car to commence mapping, it first as to survive hitting the floor after a fall of approximately 1 m. In accordance to the specification of requirements, this means that the car has to be able to sustain a force affection of 45 g

What Side is Up. During the fall the car can flip in the air, and as such, the car can land upside-down. Therefore the car needs a way to tell whether it has flipped in the air, or rather it needs a way to tell whether its bottom is facing downwards. The result of having flipped, so that the bottom of the car is facing upwards, is that the wheels have to rotate the other way around for forwards propulsion.

Start Position and direction. The car will land within 1.5 m of the wall, pointing in a random horisontal direction. As such will the car have to be able to find a wall regardless of start position and direction. This require the car to be able to drive "blindly" until it reaches a wall, and be able to initiate the room dimensioning path alongside the wall detected.

Follow a Wall. The distance at which the car has to drive alongside the walls at is given in section section 2.3, and if the car is forced to deviate from the path, either due to a feedforward has introduced an offset or an obstacle is incountered, it has to use a feedback control to reach the reference distance of 40 cm.

Rotation Points. The car needs feedback control to drive alongside the walls, but when the room dimensions are found the car needs to drive away from the walls to make its rotation point according to the mapping method found best in section section 2.1.4. How many rotation points needed depends on the dimensions of the room.

Done Mapping. When the car has made the last rotation point it has to signal the autonomous helicopter that is has finished the video map of the room.

Room Parameters

Measure Wall Lengths. The wall lengths will be used to determine the size of the room. Furthermore will the lengths be used to determine where the rotation points should be placed to get a clear view of all the walls and of the room interior.

Identify Corners. For the car to know when it has dimensioned the entire room it needs to count the number of corners. Rooms with only 90° corners, as the rooms in this project is limited to, will have at least four closed corners, depending on how many open corners there is in the room. To ensure the room is dimensioned correctly, the car will likewise need to use the lengths to determine when it once again has reached the place it initiated its dimensioning.

Identify Objects. Within the room there can be different kinds of furniture, the shapes and the obstacles these imposes, has been described in section 1.3. These objects will the car have to be able to drive around without stopping. Objects in front of the car can furthermore be detected like walls, and the car will therefore have to be able to determine whether it has detected a wall or an obstacle.

Identify Open Doors. Besides furniture as obstacle, can the room have open doors. These impose an obstacle as, the car is only supposed to make a video map of the one room it is send into. This means doors has to be identified as doors and handled like they are part of the wall the car last was driving alongside.

2.4.3 The Test Specification

As with the acceptance parameters, the tests should be performed in correspondence to the normal case scenario and to the obstacle handling described under the room parameters. For each test a corresponding simulation has to be made. Each test specification, will have the following structure:

- What is to be tested.
- How is the simulation performed and what is the expected result.
- How is the test performed and what is the expected result.

Test 1: Surviving the Impact with the Floor

This is a test of the car's physical construction and its robustness to the force the car is exposed to when it makes impact with the floor.

Simulation: This will not be simulated, as this is a test of the robustness of the physical construction.

Test: The test should be performed by dropping the car from a height of one meter. The car suspension is expected to absorb the forces affecting the car.

Test 2: Detecting Which Side is Up

The car has to determine which side of it is facing upwards, as the rotation direction of the wheels has to be reversed if the car has the bottom facing upwards.

Simulation: The simulation for this test is not whether the car can detect which side is up, but whether it can reverse the motor rotation direction and still show the same behavior and follow the same path. It is expected that the simulations will show that the car follows the same path, as it is expected that the motors can have the car drive equally fast both forwards and backwards.

Test: To test whether it matters, which side is up the car is placed first with the right side upwards and afterwards the bottom facing upwards. Both tests are expected to show the same result.

Test 3: Drive Forwards and Stop when Encountering a Wall

This tests the cars capability to drive forwards and whether it can react to inputs from the front sensor.

Simulation: These should be done in a generated rectangular room, with different directions and random starting position. The simulated car is expected to drive forwards and stop when a wall is detected in front of it.

Test: The room which this test is to be conducted in is of less importance, as the car should just be placed at different distances to a wall and at different angles. Independent of the distance and the angle to the wall, the car is expected to drive forwards and stop when it detects the wall.

Test 4: Drive alongside a wall.

The main purpose of this test is to see whether the feedback controller functions properly when implemented on the car. Furthermore, this test will include the handling of closed and open corners.

Simulation: The simulation needed here is to give an indication of the controller performance, as such, the time it takes for the car to reach its reference distance, the distance it should be driving from the wall, when it has experienced an offset should be recorded and illustrated.

As for the handling of closed and open corners, the car needs to switch off its controller and make a 90° turn. Making this turn, the simulation has to display that the car will have a turn displacement of less than the maximum allowed offset for the controller, which is 18 cm.

Test: Firstly the controller should be tested without the corner handling. Secondly the corner handling should be implemented as well and the car shall use the controller to drive alongside a wall at the reference distance, detect a wall make the 90° turn and have the controller compensate for any offset introduced during the turn, while continuing alongside the next wall. This test should be repeated with an open corner.

Test 5: Finding the Room Dimensions

From the time, the car reached the first wall and to it is in that position again, it should have measured the room dimensions and counted the corners, both open and closed.

Simulation: The three rooms from section 2.1.4, should be generated and the simulated car should drive alongside the walls and stop when it reaches its starting point. The dimensions and numbers of corners should then be printed to the screen. The three specific room shapes from section 2.1.4 is chosen as these seem like resonable office-room, more open corners is unlikely for an office.

Test: The test is performed like the simulation, the car has to drive around in the rooms. It needs to show the same behavior as the simulation indicated.

Test 6: Determining Positions for and Making the Rotation Points

From the room dimensions the car has to be able to determine where to make its rotations to have filmed the entire room.

Simulation: When the room dimensions are found the simulation has to calculate where the rotation points should be and display these in the generated room. The car hen has to drive to all of these rotation points.

Test: As with the simulation, the car has to stop and calculate, or determine, where to place the rotation points. It then needs to display the same path as the simulated car does. This test requires a room which are similar to the room simulated.

Test 7: Signal the Helicopter

When the car has mapped the entire room is has to transmit a signal indicating it has finished ts task.

Simulation: A wireless connection will have to be simulated for this test. When this is done, a signal indicating that the car has finished making the video map is transmitted through the simulated wireless connection.

Test: Instead of the helicopter a computer is used to receive the signal from the car. For this test the car has to make a video map of a room without any obstacles and transmits the signal indicating that it has done the task, when it is done.

Test 8: Object Avoidance

The avoidance of objects in the drive path should not compromise the dimensioning of the room, and neither should it compromise making the rotation points. The car should be able to drive around the objects if any is in the drive path, and it should chose an alternative rotation point if any object is placed in one of the planed rotation points.

Simulation: To illustrate furniture, obstacles in the simulation shall be placed four at the time in the corners of a square. This will be used to indicate how the car will react to furniture, as these are defined in 1.3.

Test: The test shall be performed by adding furniture of the stated shape in the test room where the car made the complete video map according to the normal case scenario.

Test 9: Handling of Open Doors

The car will need to be able to detect whether the car is passing through an open door or has made a turn at an open corner. If an open door is detected, the car has to return to the drive direction it had before it made the turn through the door.

Simulation: The simulation should be performed as with the simulation to handle open corners. The controller should be used to have the car move alongside a wall and when an open corner is detected the 90° turn should be made. The difference in this test should be that the open corner is an open door and this should be detected either before the turn is completed or right after, so that the car can rotate and get back to the drive path alongside the wall after the door.

Test: The test should be performed in the same manner as the simulation, and as such should the behavior in the test be the same as in the simulation. An extra note is that, as the car needs to count corners, it has to make sure that it either does not count the door as an open corner or count down the numbers of open corners with one per door when it detects the door.

Chapter 3

Hadware Design

Contents

The main purpose of the hardware chapter is to design a car capable of making a video map of a office like room. To do this different types of hardware are needed, such as a camera, sensors to orientate in the room, and some kind of engines to move around in the room just to mention some of the hardware designed in this chapter.

3.1 Video Coverage

The main issue is to make a video map of a room, and therefore a camera is needed. To meet the requirements from the 2.3 Specification of Requirements, the right video camera has to be found, and the placement of the camera has to be decided.

3.1.1 Requirements

The requirements for the video camera are listed below.

- Visual height at minimum filming distance: 2 m.
- Maximum filming distance $\geq 2\times$ minimum filming distance.

3.1.2 Design Considerations

As the car has to weight less than 500 g, this parameter has to be considered when choosing hardware. Of the different types of cameras, wireless cameras are considered. The wireless cameras have the advantage that they can be light weight, and are designed to have a low power consumption. Another consideration is that the picture quality has to be good enough to distinguish human faces as described in 2.1.4 Making a Video Map. Before the camera is chosen, the place for the camera on the car will be found.

Camera Placement

There are a number of placements of the camera on the car, some of them are illustrated in Figure 3.1. As mentioned in the analysis, the camera should be pointing in the driving direction, which means that the camera pointing in other directions will not be considered.

To be able to film if flipped in the air, the camera must be centered in the height, such that the camera never touches the ground. This gives three possible placements of the camera on the car, in the front of the car to the left, center and right. As the wheel size is unknown it would be an advantage to place the camera in the center of the car to get a better view to both sides. The chosen camera placement is illustrated in Figure 3.2.

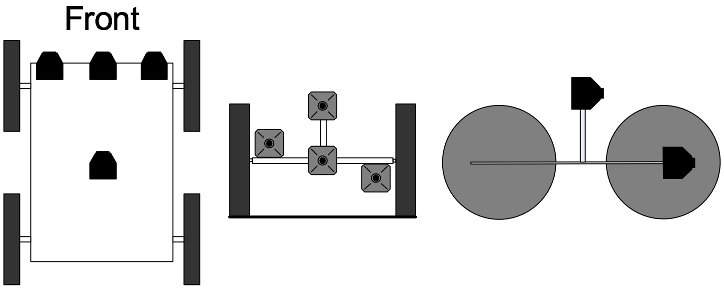


FIGURE 3.1: Different possible camera placements on the car.

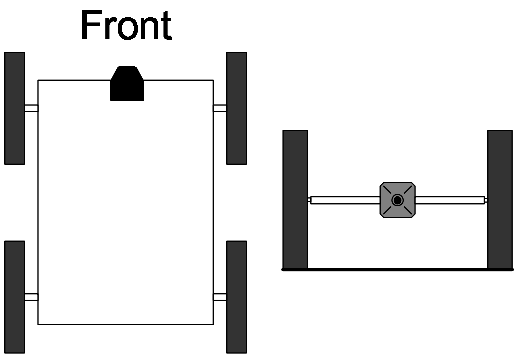


FIGURE 3.2: The chosen camera placement.

3.1.3 Video Camera Choice

The camera was found browsing the Internet for spy cameras, as it is assumed that these has the functionality, and the light weight, we need. The camera deemed best, judging from the specifications given, is the Wireless Camera GP-811T. The receiver for this camera connects to the USB-port in a computer and the camera is on line right after installing the software that comes with the camera on a cd.

To use the camera in an optimal way, modifications to its power supply cord was needed, as it had a connector almost the size of the camera itself. Also the view angles of the camera had to be found to define how far from the walls we need to be to film the faces of people in the room. Furthermore, a device was constructed to mount the camera on the car.

Transmission Frequency: ISM-2.400 $\approx$ 2.483 GHz

Modulation Type: FM

Bandwidth: 18 MHz

Undisturbed Transmission Range: 100 m

Supply Cord

The camera uses an eight voltage supply which is assured by a transformer, connected to the mains. To use it on the car, the supply cord was cut and connected to the cars power supply.

Mounting

To film the room the car has a mounted camera pointing forwards. For full use of the camera's scope, a tilting mechanism is designed to have the camera film as little of the floor as possible, measured at its operating distance. The tilt mechanism is shown in Figure 3.3 and its mount on the car is illustrated in Figure 3.4.

View Angles

The desired camera coverage is to be able to see approximately two meters up at a distance of two meters from the wall. This means the camera will need to have a vertical view angle of $\sim 45°$. For the chosen camera the vertical

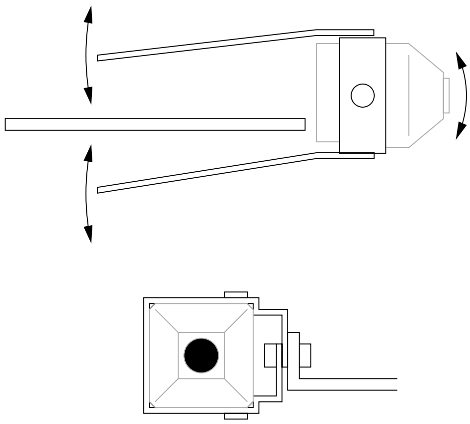


FIGURE 3.3: The tilt mechanism.

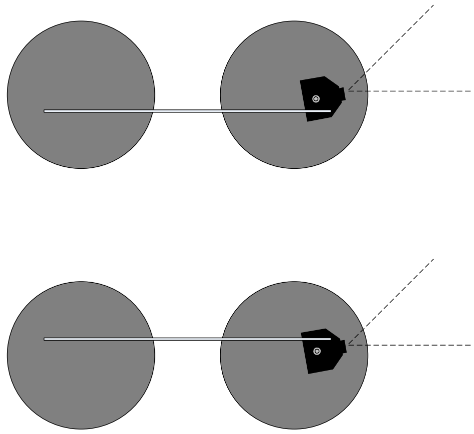


FIGURE 3.4: The mounted tilt mechanism.

view angle is $\sim 44°$, which, by itself, is acceptable. And when taking into consideration that the camera is elevated 5 or 5.7 cm depending on which side of the car face upwards, the actual view height is approximately 1.98 m at a two meters distance to the wall. The vertical and horizontal view angles is show in Figures 3.5 and 3.6 respectively.

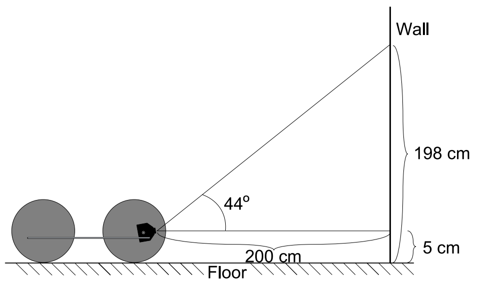


FIGURE 3.5: The vertical view angle of the camera.

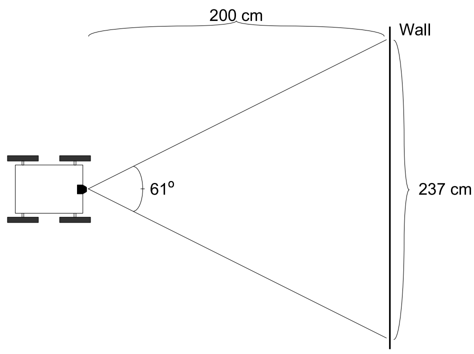


FIGURE 3.6: The horizontal view angle of the camera.

3.1.4 Conclusion

The best placement of the camera was found to be in the center of the front of the car, which will be used for the camera. The chosen GP-811T wireless camera reaches a view height of approximately 1.98 m on a distance of 2 m, which is two centimeters short of the requirement, but it is close enough to be acceptable. The camera has a maximum distance of more than 4 m, and the second requirement for the camera is then maintained.

3.2 Sensors

The car has to orientate in the room, meaning that it has to find the size and shape of the room. To do this distance sensors will be needed. For determining which side of the car is pointing upwards a sensor to place at the bottom of the car facing downwards will be found.

3.2.1 Requirements

The requirements from 2.3 Specification of Requirements and 3.1 Video Coverage for navigation are listed below.

Distance: Maximum distance measuring range of more than 0.4 m.

Up/Down: The sensor output has to have a significant difference when pointing from the bottom of the car to the floor and when pointing upwards.

3.2.2 Design Considerations

There are a need of two different types of sensors, distance sensors which are often analogue, and a sensor to tell what side is up/down on the car, this could be a digital binary sensor.

Sensor Placement

The distance sensors can be placed different places, but there has to be one pointing in the driving direction to prevent collision with obstacles and walls. And since it is wanted to drive alongside the walls of the room, a sensor at one of the sides of the car is necessary. Because the car can flip in the air and thereby turn upside down, it is needed to have a sensor at each side of the car. This gives at least three distance sensors as shown in Figure 3.7.

To get a better view, more sensors can be added, e.g. more than one front and side sensor, and a sensor pointing backwards. But since the car can drive forwards and rotate it is assumed that the car is not driving backwards, and

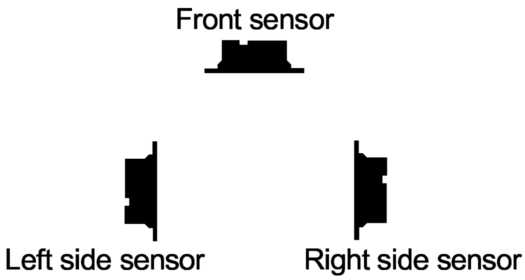


FIGURE 3.7: The car's sensor configuration.

therefore a sensor pointing backwards is not necessary. This is the reason that a sensor pointing in this direction is not used.

The front sensor is used to detect obstacles and new walls. To detect a new wall only requires a single sensor, since it is wide and easy to detect, but to detect an object requires more from the sensor since it can be small, and therefore the sensor have to have a wide view angle, that covers at least the width of the car at a given distance, to avoid collision with the object.

It is expected that the side sensors primarily are used to follow the walls in the room, and therefore only used as feedback to a controller, which means that a single sensor at each side is enough. To center the weight, and thereby stabilize the car, it would be an advantage to place the sensors in the middle of the length of the car at each side.

The wanted sensor coverage is illustrated in Figure 3.8. The outer circle is the wanted distance the sensors should be able to sense, the dotted lines are marking the width the sensors at least should cover at the given distance.

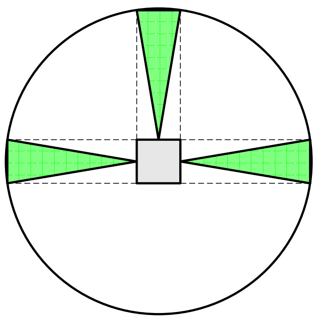


FIGURE 3.8: This is an illustration of the wanted view with the front and side sensors.

During its fall, the car may flip and thereby land upside down. Therefore it has to identify which side is up. To do this a sensor is placed facing downwards, initially, as illustrated in Figure 3.9.

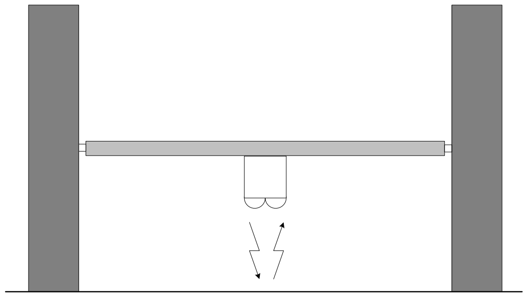


FIGURE 3.9: The sensor, used to detect the floor, is initially facing downward.

The idea is to measure the distance from the bottom of the car to the floor and thereby determine whether the car has flipped. In practice a threshold is defined and if the distance is greater, the car has flipped.

3.2.3 Distance Sensors Choice

The chosen sensor is a standard IR sensor, that is easy to obtain, light weight and protected by a plastic casing. The considered distance sensor is SHARP GP2D12. It has an internal distance measuring IC and a connector with supply, ground, and signal pins. It is easy mountable with two screws. Its output is an analogue signal of $0.6-4.7\ V$ which corresponds to the distance range $10-80\ cm$. The weight of the sensor is $3.6\ g$. At a distance of $40\ cm$ the sensor view width is about $4\ cm$.[GP2D12.pdf] The sensor is an IR sensor, and therefore light sensitive.

If two front sensors are used the required view angle is halved, and if some sensors that have a view angle that covers the width of the car this would be the optimum placement of front sensors. This also enables the camera to be placed in the center of the front of the car, which were found to be the best place for the camera in 3.1 Video Coverage. The found sensor setup is illustrated in Figure 3.10.

Figure 3.11 illustrates what the chosen sensors cover in respect to the requirements.

The range of $0.8\ m$ is considered sufficient to detect obstacles in front of the car, and to keep a constant distance to the wall that the car is driving alongside. Furthermore the requirement of a maximum distance measuring range greater than $0.4\ m$ has been respected.

To find the measuring noise, eight tests were conducted, and the test results are shown in Figure 3.12. This is a small sample case, but it is assumed that

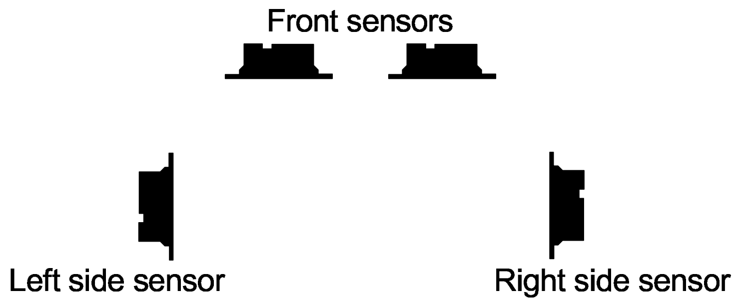


Figure 3.10: The placement of two front sensors and two side sensors, one at each side.

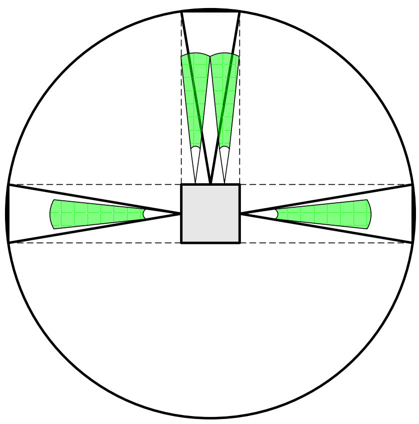


Figure 3.11: The colored areas shows the work area of the sensors.

it is enough to find the noise with. The line is the mean value of the eight tests, and the dots are the actual measuring points.

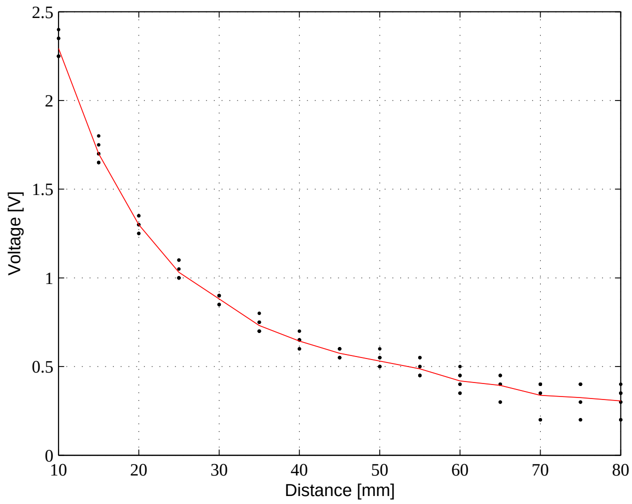


FIGURE 3.12: GP2D12 measuring results.

From the tests the measurement noise is calculated by dividing the standard deviation with the mean value, and to get it in percentage, this fraction is multiplied with 100. The result is shown in Table 3.1. At distances below 50 *cm* less than 5% measuring noise is maintained, with one exception at a distance of 35 *cm*. Within this range the maximum distance range requirement has been obtained. The graph in Figure 3.12 shows that distances up to 60 *cm* is recognizable despite of the high noise percentage, but at distances over 60 *cm* it is difficult to distinguish the different distances due to the small difference in output from the sensors and the high amount of noise.

Distance [cm]	10	15	20	25	30	35	40	45
Noise [%]	2.72	3.14	2.91	4.44	2.94	5.09	4.98	4.65
Distance [cm]	50	55	60	65	70	75	80	
Noise [%]	7.00	9.09	12.67	15.83	25.96	27.27	23.80	

TABLE 3.1: Measurement noise in percentage from the GP2D12 sensor test at different distances.

3.2.4 Up/Down Sensor Choice

Of the sensors available, two has been chosen to be examined further, SHARP GP2D15 and Optek OPB704. The GP2D15 sensors characteristics are shown in Figure 3.13 and Figure 3.14, and the characteristics of sensor OPB704 is shown in Figure 3.15.

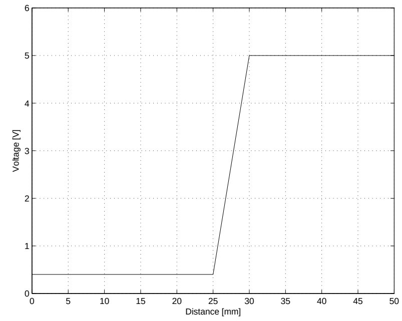


FIGURE 3.13: Characteristics of the GP2D15 sensor at low distances.

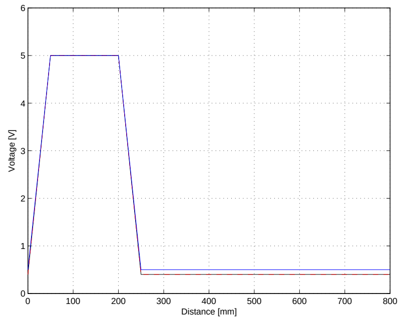


FIGURE 3.14: Full range of the GP2D15 sensor.

The GP2D15 is a digital sensor with high output when an object is closer than about 25 cm, and low otherwise. But at low distances under 2.5 cm it can not detect objects, and the output then is low. As this might be the distance from the bottom of the car to the floor, this sensor is not usable.

The OPB704 sensor is an analogue with high output when there is no objects, and lower the closer an object becomes. To be sure to distinguish the floor from the air, the sensor has to be placed such that it is 1.5 cm or closer to the floor when pointing downwards. This gives an output difference of about $4.8 - 3 = 1.8\ V$. The OPB704 sensor is chosen to be the up/down sensor.

OPB704 is an IR sensor, but it consists only of a LED and a photo transistor in a casing. To control the current through the LED, an external circuit will have to be designed. The distance interval of the OPB704 is $5 - 20\ mm$. The weight of the OPB704 is 1.4 g, plus any extra components needed.[OPB704-extra.pdf]

The needed circuit is designed in the following.

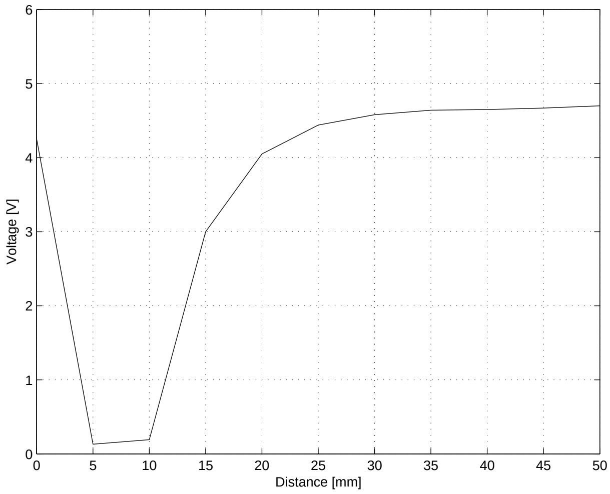


FIGURE 3.15: Characteristics of the OPB704 sensor.

OPB704 circuit

The circuit used[1], is illustrated in Figure 3.16. The basic idea is to have a low output when the sensor is detecting the floor and high otherwise. Therefore a resistor connects the output with Vcc.

Furthermore there is a need to control the current through the LED. The current wanted is 20 mA. A resistor is serial connected from Vcc to the anode on the LED. The voltage drop across the LED is 1.6 V. Resistor dimensioning:

$$R = \frac{(U_{Vcc} - U_{LED})}{I} = \frac{5\ V - 1.6\ V}{20\ mA} \approx 180\ \Omega$$

The emitter of the photo transistor and the cathode of the LED is both connected to ground.

3.2.5 Conclusion

The GP2D12 has an acceptable resolution, and the measurement range up to 60 cm is considered sufficient. Furthermore the noise is found to be lower than 5% when staying closer to the wall than 50 cm. This is enough to use this sensor as a distance sensor on the car.

[1]The circuit was found on the Internet, at http://www.roborugby.org/optical.html

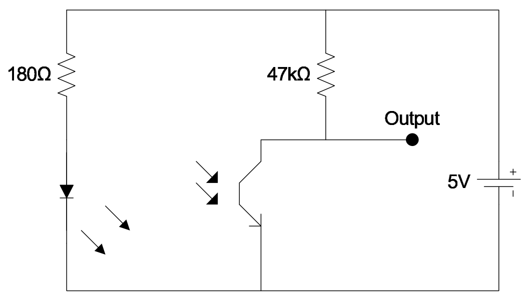


FIGURE 3.16: OPB704 circuit.

The tests of the two up/down sensors were conducted to find the better one and with the requirements stated in the beginning of this section, only one of the sensors are usable, the OPB704. Another fact to back up this choice is that the OPB704 is the most light weight of the two sensors. The total weight of the OPB704 and circuit is 2.4 g, which is acceptable. The OPB704 sensor is deemed usable.

3.3 Movement

To get the car moving, one or more motors are needed, and the propulsion is transferred to movement through the use of wheels.

3.3.1 Requirements

Forwards. The maximum angle of which the car is allowed to deviate from the linear path is 2.7°.

Backwards. Same as for the forwards motion.

Turning. Maximum

- displacement: 4 cm.
- deviation angel: 2.7°.

Rotating. Maximum

- displacement: Horizontal distance that corresponds to 17 cm vertical distance. Depending on the vertical filming angle.
- deviation angel: The horizontal filming angle.

3.3.2 Design Considerations

With the found placement of sensors and camera, a four wheel configuration is chosen. To find the right motors, the motor type and the motor placement has to be chosen. Then the size and material of the wheels has to be designed to protect the rest of the car. After these considerations the motor can be found.

Motor Type

To avoid use of fuel, and thereby the need of a fuel tank, electrical motors are used. In particular RC μ-servos are considered due to their small size and light weight. To be able to use this kind of motors, they have to be modified to turn around like other motors. This is done by cutting off the plastic that prevent the full rotation motion inside the motor. To find the number of motors, the placement of the motors are found.

Motor Placement

With the four wheel configuration and the sensors and camera on the car, two possibilities are considered. These two has a minimum of mechanics to make the car lighter, but it comes at the expense of robustness. The two considered motor placements are illustrated in Figure 3.17. The option on the left on Figure 3.17 has two motors, one at each side, that are connected to the wheels of the same side. The advantage of this configuration is that it is lighter than the other configuration, but it is also more vulnerable to mechanical failure when the car hits the ground from its fall through the window, due to the mechanics needed to connect the motors to the wheels. The other option is to have an engine for each wheel to avoid the use of gears and drive belts, which could result in mechanical failure as a consequence of the car hitting the floor at the end of its flight. The disadvantage of this configuration is that it weights more because of the two extra motors needed. Because of the better robustness the configuration with four motors is chosen, despite that it weights more. Both configurations are able to turn, by setting the left and right side drive opposite directions.

3.3.3 Chosen Wheels

The chosen wheels are mainly made of foam rubber. To connect the wheel to the servo a disc is glued on the center of the wheel, and the wing from the

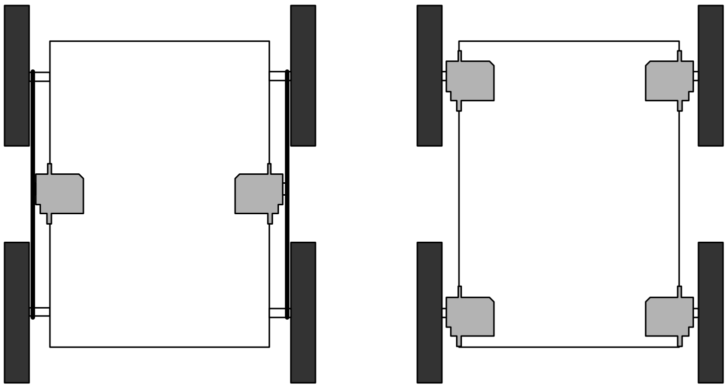


FIGURE 3.17: The two considered motor placement possibilities.

servo is glued on to the disc. The wheel can be seen in Figure 3.18.

The foam rubber is used because of the light weight, and that it is not going to carry a heavy weight, which means that the foam rubber is strong enough to carry the car. The diameter of the wheels are 9 cm.

The total weight of a wheel is measured to be 4 g.

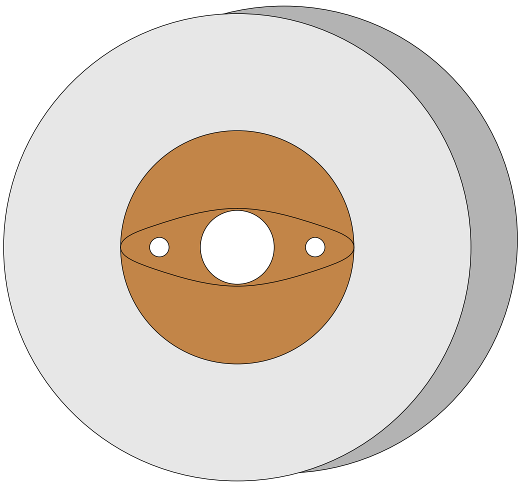


FIGURE 3.18: The constructed wheels.

3.3.4 Chosen Motors

The μ-servo candidates were found on the internet, from a danish store the university have used before. Two factors are considered in the choice of engine: Weight, and speed.

Weight, m: Addressing the weight factor, no maximum weight limit is specified. Instead, a specific type of motor is chosen. μ-servos have the advantage of low weight and enough torque to drive low weight vehicles and are even usable for small RC helicopters. Most μ-servos weigh less than 10 g.

Speed, v: Time is a factor, and as such is the speed of the car a factor as well. The speed of μ-servos is typically $58 - 111$ rpm.

Possibilities

Of the μ-servos studied, two were considered to be best suitable for the car. One, which has the lowest weight and one, which has the highest speed. In Table 3.2 the technical data for the two μ-servos is specified.

Data	Contraction	Hitec HS-50	Blue Bird BMS-303	Units
Weight	m	5	3.4	[g]
Speed	v	111	90.9	[rpm]
Torque	τ	0.0586	0.0777	[Nm]

TABLE 3.2: Best suitable engine choices for the car. The lightest and the fastest of the μ-servos studied.

Since the total weight of the car is unknown at this point, the maximum weight the motors have to get in motion is 0.5 kg. Both motors are assumed to be able to move the weight of the car, and with the chosen motor placement four motors are used, which means that each motor has to move $0.5/4 = 0.125$ kg. Despite of the slightly more weight and less torque the Hitec HS-50 is chosen because of its high speed.

A speed of 100 rpm is chosen, because it is still faster then the alternate μ-servo type, and gives a velocity of the car on 47 cm/s with the chosen wheel size. The μ-servos are tested in Appendix C.3. The test shows that this speed can be uphold with no load applied, but when the load of the cars weight is applied the car drives with 49 rpm, which corresponds to 23 cm/s. This is due to the low torque of the μ-servo.

3.3.5 Conclusion

As the four μ-servos are able to obtain the same velocity, it is assumed that it is possible to maintain a straight driving path, and the requirement of the maximum angle of which the car is allowed to deviate from the linear path is 2.7° is fulfilled, likewise with the turning and rotation requirements.

The HS-50 μ-servos were chosen because it was the fastest, and the test showed that it works as wanted for the purpose of this project. With this μ-servo the car can drive with a velocity of 23 cm/h, which is about the half of what was expected, but it is assumed to be sufficient for now. Therefore the conclusion is that the HS-50 μ-servo is used in this project as driving engines.

3.4 Micro Computer

To control the car a micro computer is needed.

3.4.1 Requirements

The requirements for the micro computer are listed below.

- PWM port to control the μ-servos
- ADC to get data from the analogue sensors
- Interface to a PC

3.4.2 Design Considerations

The micro computer has to have a small physical construction, and be able to mount on the car, without any loose parts that could fall off when the car hits the floor. The following micro computer were found on the internet.

3.4.3 Chosen Micro Computer

The chosen micro computer for this project is a Gumpack, which contains two micro computers, a gumstix motherboard and a robostix micro controller. This solution is small in size, and light weight, which is preffered. Furthermore this solution has the ports needed to interact with the motors and sensors, PWM and ADC respectively.

The two micro computers have the following specifications, found at [Gumstix inc., 2007].

Gumstix connex 200xm

The technical specifications of the gumstix are shown below.

Processor: Intel XScale PXA255

CPU speed: 200 MHz

Flash memory: 4 MB

Connections: 60 pin Hirose I/O connector, 92-pin bus header

This is an older version of the gumstix, the new version has 16 MB flash memory instead of 4 MB. The 60 pin Hirose I/O connector is used for basix-side expansion board , in this case to connect to the robostix. The 92-pin bus header is used for connex-side expansion boards like cfstix.

Robostix R341

The technical specifications of the robostix are shown below.

Processor: ATMega128 (an Atmel AVR processor)

CPU speed: 16 MHz

Flash memory: 128 KB

Connections: 60 pin Hirose I/O connector, I2C bus

Other functions: FFUART, ADC, PWM, timers, interrupts

The robostix can run programs as a stand alone micro controller, or be connected to the gumstix by the 60 pin Hirose I/O connector to maximize the programming capabilities.

To get data from the sensors, the ADC pins 0-4 at Port F on the robostix are used. The sensors are connected directly to this port, from where the they get their power supply.

The PWM output pins 1A and 1B at Port B are used to control the μ-servos. The μ-servos get their power from the power supply, and the signal wire is connected directly at the port on the robostix.

When the gumstix is mounted on the robostix with the 60 pin Hirose I/O connector, the FFUART pins can be used with a RS232 circuit to connect the gumstix with a PC. To activate this communication port, pin 1 is connected to pin 4, and pin 5 to pin 8 on the UART port on the robostix. The ports used are shown in Figure 3.19.

There are three timers/counters on the robostix. One timer is used to generate the PWM output, another is used to control the input rate from the sensors, and the last timer is then usable for the robostix program.

The only circuit that has to be designed is the RS232 circuit. Interrupts are not used at this point.

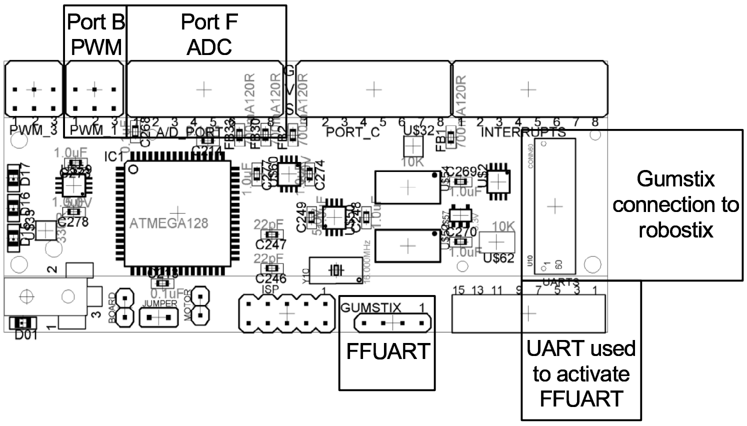


FIGURE 3.19: The ports used on the robostix.

MAX232A

As mentioned above the only external circuit needed is for communicating with a PC. To do this a MAX232A line driver from Maxim is chosen due to its availability.

RS232 Communication

The MAX232A circuit is constructed as the data sheet[MAX238.pdf] describes. Figure 3.20 shows the circuit diagram of MAX232A, with the used components. The circuit is constructed on its own Veroboard, and are connected to a PC with leads and a DB9 female connector, and the robostix with leads and a four pin connector.

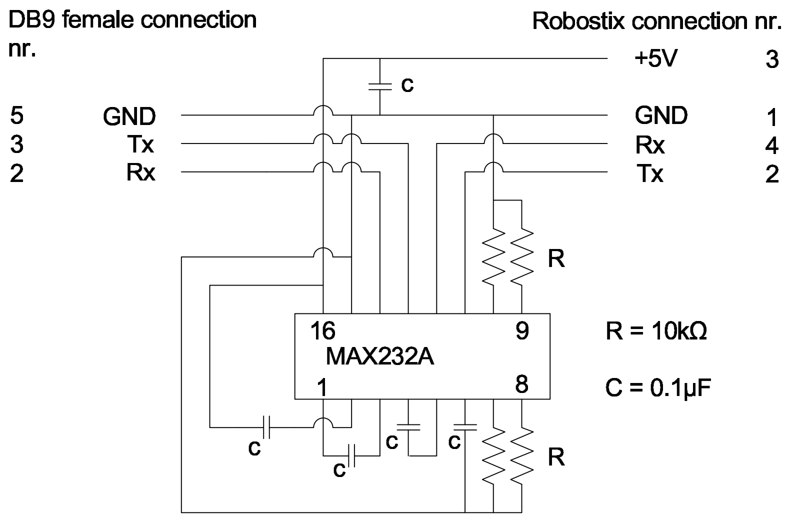


FIGURE 3.20: Diagram of MAX232A circuit.

3.4.4 Conclusion

The gum pack fulfills the requirements, and in combination with the RS232A circuit it can be used directly in this project. The PWM port (1A-B) is used to control the motors, the ADC port (0-4) is used to get data from the sensors, and the FFUART port is used to communicate with a PC through the RS232A circuit.

3.5 Power Supply

The circuit for the car's electrical power supply is illustrated in Figure 3.21. A test to find the current through each components, The Robostix, the motors, and the camera, was performed. The general test setup is shown in Figure 3.22 and the laboratory equipment used was:

	Manufacturer	Model	Lab.No.
Power Supply	HAMEG	HM7042-3	52755

3.5.1 Robostix, Gumstix, and Sensors

To find the maximum current through the car system the first test conducted was for the Robostix, which also functions as supply for the Gumstix and the sensors. The following five tests were conducted.

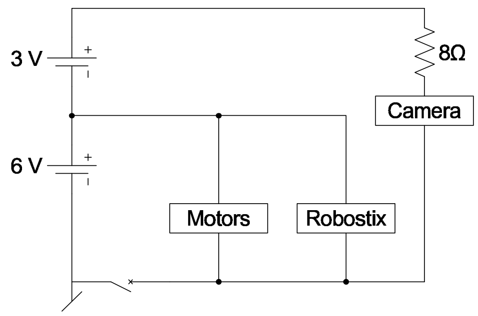


FIGURE 3.21: The Power supply circuit for the car's systems.

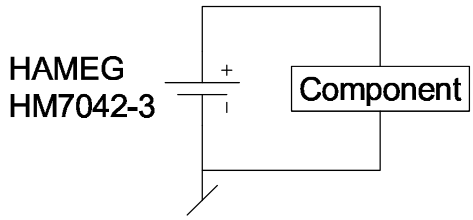


FIGURE 3.22: The general test setup for the tests of the components in the supply circuit.

When idle:	215 mA
With uploaded Robostix driver and idle:	263 mA
When data is send from a PC the Gumstix:	325 mA
When the Gumstix sends data to the Robostix	300 mA
When the both the Robosix and Gumstix are active	330 mA

3.5.2 Motors

Three test where performed on the μ-servos, when idle, when active with no load, and when active and driving on the floor (with load).

When idle:	43 mA
With no load:	400 mA
With load:	$1450 - 1550$ mA

3.5.3 Camera

As the camera is activated from the beginning of the mission and it only has one active state, and only one test was needed. The test of the active camera showed that the current through it is 125 mA. The camera needs to have 8 V and uses 125 mA, so a resistance of 8 Ω is inserted in series with the camera.

3.5.4 Discussion

The tests showed that the maximum total current through the three components is 2005 mA. The 6 V supply has to deliver this charge, while the less loaded 3 V supply has to deliver a charge of 125 mA.

The batteries used are Varta No.4003. Which are AAA batteries with a voltage supply of 1.5 V and a charge of 1200mAh.

This means that the 6 V supply can last for ${}^{1200\ mAh}/_{2005\ mA} = 0.6\ h \approx 36\ minutes$ and the 3 V supply can last for ${}^{1200\ mAh}/_{125\ mA} = 9.6\ h = 576\ minutes$.

With the circuit shown in Figure 3.21 we encountered a problem though. It seems, the batteries in the 6 V supply cannot keep the voltage when the motors are activated by the Robostix. The voltage drop to 2.2 V which is not enough for the Robostix to keep active. To accommodate this problem an extra 6 V voltage supply, consisting of four extra batteries of the same kind, are added to the circuit in parallel to the first 6 V supply. The circuit with a total of three voltage supplies is illustrated in Figure 3.23. The maximum charges are also included in this figure.

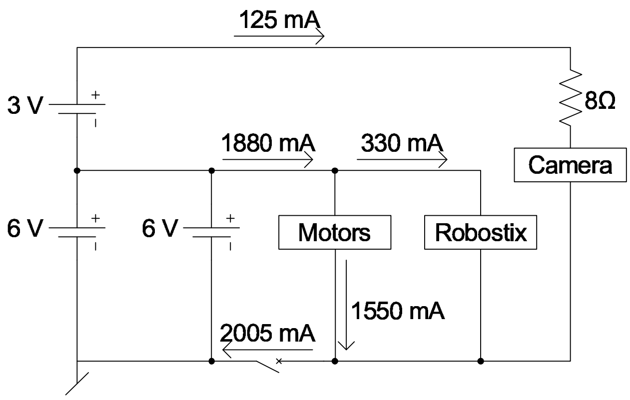


FIGURE 3.23: The final power supply circuit.

As with the previous circuit a test were performed on this circuit and the result was that it is able to keep the voltage and thereby keep the system active. The double 6 V supply can therefore keep the system active for 72 $minutes$.

3.5.5 Conclusion

The maximum system current load is 2005 mA which is supplied by use of 10 AAA batteries. The entire system can be kept active for 72 $minutes$.

3.6 Robustness

With all the hardware designed, the total weight of the car is measured to be 280 g. The weight is used to calculate the force the car is hitting the ground from one meters height with.

3.6.1 Requirements

There are the following requirements for the robustness of the car.

- The maximum allowed weight of the car is 0.5 kg.
- The car needs to be able to sustain hitting the floor after a fall of 1 m. The energy required for the car suspension to absorb is 4.9 J. The force of this fall will maximum be 45 g.

3.6.2 Design Considerations

The wheels were chosen to absorb the energy from the fall. The robustness was given a low priority, which means that light weight hardware was considered more important. This results in a car that can be thought of as a rigid body because it has no suspention, a hard aluminium foundation plate. The weakest link is assumed to be the screw that holds the wheels on the μ-servos.

3.6.3 Conclusion

As the robustness has been given a low priority, the test of dropping the car from one meters height has not been conducted. This means that it is uncertain if the car can sustain the hit with the floor from one meters height.

3.7 Discussion

The chosen camera halved its weight when the supply cord were cut, and connected to the power supply on the car through a resistance to limit the current to the camera, and lower the voltage to 8 V which the camera needs.

The two front sensors were placed on each side of the camera, giving a good view in front of the car to detect obstacles. Through practical tests the side sensors had to be moved because the wheels send shadows on the sensors so that they did not give the expected values.

Tests showed that the μ-servos did not have enough torque to keep a high velocity. A possible solution is to use the other μ-servos considered because they have about 20% higher torque, which might be enough to keep the velocity higher despite of the lower maximum velocity.

The micro computer has the needed ports for controlling μ-servos and enough input ports for the distance sensors and the up/down sensor. The CPU on the robostix is powerful enough to control the car by itself without the gumstix if necessary.

The power supply has a high weight compared to the other hardware, it might be possible to use a battery from a mobile phone or the like to get a light weight power supply. Despite the weight the power supply works as expected.

The robustness of the car got a low priority, and it is therefore unknown if the car is light enough and the wheels are able to absorb the energy to sustain a fall from a height of one meter.

3.8 Conclusion

The main tasks for the hardware is to make a video map of the room, drive alongside the walls to find the dimensions of the room, and to survive a impact with the floor where the car is dropped from the height of a window ($1\ m$).

The hardware has the capabilities of driving, which implies that it is possible to find the dimensions of a room. Furthermore with the chosen camera it is possible to make a video map of the room the car drives in. But as the impact test has not been conducted, it is uncertain if the car can sustain hitting the floor from one meters height. From the above statements the hardware is considered sufficient to make a video map of a room, where the dimensions of the room has to be found first.

Chapter 4

Controller

Contents

This chapter evolves around designing a controller for the autonomous car. The idea with the controller is to ensure that the car can follow a wall at the distance given in the specification of requirements, 2.3. In the design of the controller we have considered whether to use linear or non-linear control. As such, does this chapter contain discussions of different control methods and of whether the controller should be model based or not. Furthermore is the design method we use discussed and so is the specific requirements concerning the controller. Having designed and simulated the controller, it is discussed how to implement it in software, and whether the controller can function outside the boundaries of the requirements. For instance with a larger allowed offset.

4.1 Linear and Non-linear Control

In general terms, there are two groups of systems, the linear and the non-linear system. As such, control methods for both groups has been developed, or is under development. The methods for designing non-linear controllers are newer and are therefore a less proven field, whereas within the methods of designing linear controllers there are more thoroughly proven methods. It is therefore preferable to have a linear system, or to be able to approximate linear operation areas in the non-linear system.

4.1.1 Non-linear Systems

Before discarding the idea of making a single controller for the non-linear system, we look at one of the most common the possibilities within non-linear control.

Backstepping is a method where a virtual feedback is introduced to compensate for non-linearities which can amplify the disturbances. That is the odd-ordered, besides first order, terms in the feedback.

4.1.2 Linearise the Non-linear System

The typical method, when dealing with non-linear system is to linearise the system in one or more operating points, each with a defined operation area. To illustrate this we look at a first order differential equation, given in equation 4.1.

$$\frac{dx}{dt} = f(x, u) \tag{4.1}$$

$f(x, u)$ is a non-linear function and, as such, we choose an operating point and an operation area. The operation area is defined from where the function $f(x, u)$ is approximately linear around the operating point. If the system at any given input can move outside the defined operating area, more than one operation point is needed, which will require more than one linearisation.

An operating point has to be chosen and equation 4.1 can be expressed as:

$$\frac{d\overline{x}}{dt} = f(\overline{x}, \overline{u}) \tag{4.2}$$

The chosen operating point along with the deviation from it, Δx, is given in equation 4.3.

$$\frac{d}{dt}\left(\overline{x}+\Delta x\right)=f(\overline{x}+\Delta x,\overline{u}+\Delta u) \tag{4.3}$$

The right side of equation 4.3 is linearised using Taylor Expansion

$$f(\overline{x}+\Delta x,\overline{u}+\Delta u)=f(\overline{x},\overline{u})+\frac{df(\overline{x},\overline{u})}{dx}\cdot\Delta x+\frac{df(\overline{x},\overline{u})}{du}\cdot\Delta u \tag{4.4}$$

The higher orders of the Tailor Development is not expressed in equation 4.4, as Δx is rather small, which means that Δx^n, where $n>1$ is approximately 0. Inserting equation 4.4 into equation 4.3 and isolating $f(\overline{x},\overline{u})$.

$$\begin{aligned}\frac{d}{dt}\left(\overline{x}+\Delta x\right) &= f(\overline{x},\overline{u})+\frac{df(\overline{x},\overline{u})}{dx}\cdot\Delta x+\frac{df(\overline{x},\overline{u})}{du}\cdot\Delta u\Leftrightarrow\\ f(\overline{x},\overline{u}) &= \frac{d\overline{x}}{dt}+\frac{d\Delta x}{dt}-\left(\frac{df(\overline{x},\overline{u})}{dx}\cdot\Delta x+\frac{df(\overline{x},\overline{u})}{du}\cdot\Delta u\right)\end{aligned} \tag{4.5}$$

$f(\overline{x},\overline{u})$ can then be substituted in equation 4.2, and the linear approximation of the nonlinear system is presented.

$$\begin{aligned}\frac{d\overline{x}}{dt} &= \frac{d\overline{x}}{dt}+\frac{d\Delta x}{dt}-\left(\frac{df(\overline{x},\overline{u})}{dx}\cdot\Delta x+\frac{df(\overline{x},\overline{u})}{du}\cdot\Delta u\right)\Leftrightarrow\\ \frac{d\Delta x}{dt} &= \frac{df(\overline{x},\overline{u})}{dx}\cdot\Delta x+\frac{df(\overline{x},\overline{u})}{du}\cdot\Delta u\end{aligned} \tag{4.6}$$

As the non-linear system can be linearised and there are more proven methods to design linear controllers, we have chosen to make linear control of the autonomous car designed in this project.

4.2 Model Based Control

A model is used to mathematical describe a physical system. A dynamical linear model is a set of input, output and state variables related by differential equations. If the system is non linear, a number of work points can be made where the model is linear around the point. A dynamical linear model can be used to make a state space representation of the system, that has the following form for a continuous time-invariant system,

$$\dot{x}(t) = Ax(t) + B\mathbf{u}(t) \tag{4.7}$$

$$y(t) = Cx(t) + D\mathbf{u}(t) \tag{4.8}$$

where $x(t)$ is the state vector, $y(t)$ is the output vector, $\mathbf{u}(t)$ is the input or the control vector, A is the state matrix, B is the input matrix, C is the output matrix and D is the feedforward matrix. The structure of this approach is illustrated in Figure 4.1. In most cases feedforward is not used, which means that D only contains zeros and D is then discarded.

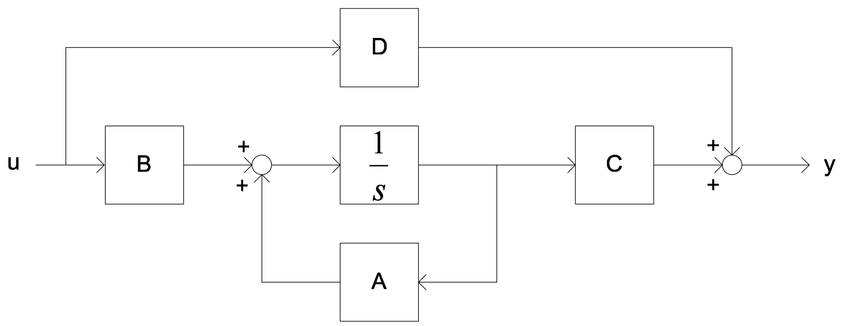


FIGURE 4.1: The structure of the state space representation.

Stability

Stability of a system can be studied from the eigenvalues of the state matrix A. The eigenvalues of A yields the poles in the systems transfer function, which can be used to analyze the systems stability. The eigenvalues of A are found through the characteristic polynomial,

$$det(\lambda I - A) = 0$$

where λ gives the poles and I is the identity matrix of the same dimensions as A. If the system has unstable poles, zeros can be found to cancel out those. The zeros must satisfy the following equation,

$$\begin{vmatrix} A - zI & B \\ C & D \end{vmatrix} = 0$$

where z gives the zeros. Furthermore to ensure stability the system has to be controllable and observable.

Controllability

A continuous time-invariant state-space model is controllable if and only if

$$Rank[B\ AB\ A^2B\ \ldots\ A^{n-1}B] = n$$

where Rank is the number of linearly independent rows in a matrix, and n is the number of state variables.

Observability

A continuous time-invariant state-space model is observable if and only if

$$\text{Rank}\begin{bmatrix} \text{C} \\ \text{CA} \\ \ldots \\ \ldots \\ \text{CA}^{\text{n-1}} \end{bmatrix} = \text{n}$$

Conclusion

It is an advantage to have model of the system that has to be controlled, because it describes the dynamics of the system, which opens for the possibility of fine tuning the controller specifically for the given system. Furthermore it is easy to analyze for stability by finding the poles of the state matrix. Both model based controllers and classical controllers can be used once the model is made.

4.3 Classic Control

In this section we look at the classic control for linear systems. More specifically we look at, and discuss, the PID controller and its uses for this project. The PID controller contains a proportional control, P, an integral control, I, and an derivative control, D. A standard feedback control setup is used, shown in Figure 4.2.

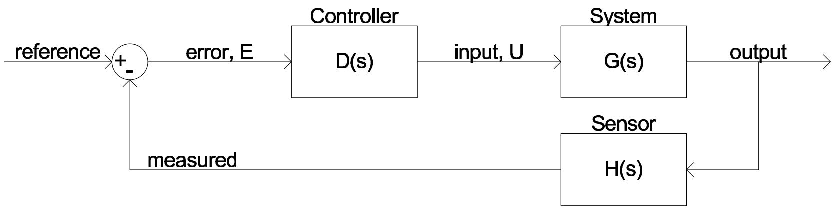


FIGURE 4.2: The standard feedback control loop, with a system, G, a controller, D, and a sensor, H.

4.3.1 The Proportional Control

The P controller is proportional to the system error, which gives the system a faster response time if the controller gain, k_p, is increased.

Using a P controller, will usually cause a steady-state error on the output, this will get smaller if the gain s increased. An increase in k_p will on the other hand give result in an increasing overshoot and can as such make a higher order system unstable. The expression for the proportional control is given, in both time and Laplace domain, in 4.9.

$$\begin{aligned} \text{time-domain: } & u(t) = k_p \cdot e(t) \\ \text{Laplace-domain: } & \frac{U(s)}{E(s)} = D(s) = k_p \end{aligned} \tag{4.9}$$

To compensate for the steady state error an integral control can be added.

4.3.2 The Integral Control

The integral control sums the error over time, which gives an accumulated offset this means that a system with an integral control only, will always experience overshoot and the integral control will not stop changing until the input is zero.

The integral control will eliminate the steady state error introduced by the proportional control and make the response time faster with an increase in the integral gain, k_i. But as with the k_p, will the overshoot increase with an increase in k_i and so will the time the system takes to settle. In 4.10 are the expressions for the integral controller in time and Laplace-domains given.

$$\text{time-domain: } u = k_i \cdot \int_{t_0}^{t} e(\tau) d\tau$$

$$\text{Laplace-domain: } \frac{U(s)}{E(s)} = D(s) = \frac{k_i}{s} \tag{4.10}$$

Both the proportional and the integral control gave an increase in overshoot if the gains were increased, the derivative control can be added to compensate for this.

4.3.3 The Derivative Control

The derivative control add the change in error over time. The change is multiplied with the derivative gain, k_d. By adding the derivative term to the controller, the change in controller output is slowed, which means that the overshoot is decreased and, as such, the stability is improved. Differentiating the error means that the noise is amplified, which leads to a high sensitivity to noise when adding the derivative control. The expression for the derivative control is given, in both time and Laplace domain, in 4.11.

$$\begin{aligned} \text{time-domain: } u &= k_d \cdot \frac{de(t)}{dt} \\ \text{Laplace-domain: } \frac{U(s)}{E(s)} &= D(s) = k_d \cdot s \end{aligned} \tag{4.11}$$

Depending on the requirements set up for the controller each term, P, I, and D, can be dimensioned.

4.3.4 PID Controller

the three terms are summed in the controller, as shown in Figure 4.3. The expression for the PID controller is given in 4.12, both in the time-domain and in the Laplace-domain.

$$\begin{aligned} \text{time-domain: } u(t) &= k_p \cdot e(t) + k_i \cdot \int_{t_0}^{t} e(\tau)d\tau + k_d \cdot \frac{de(t)}{dt} \\ \text{Laplace-domain: } \frac{U(s)}{E(s)} &= D(s) = k_p + \frac{k_i}{s} + k_d \cdot s \end{aligned} \tag{4.12}$$

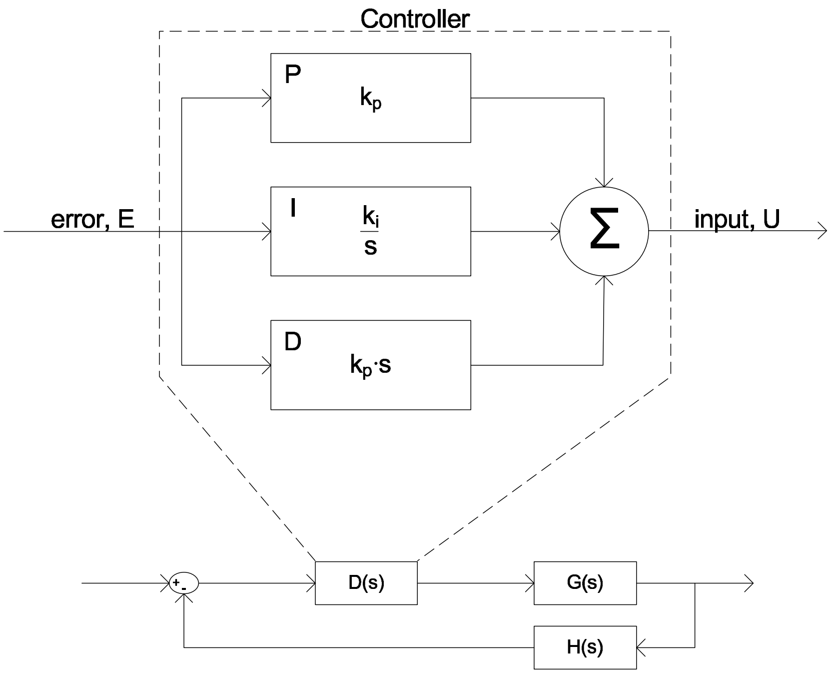


FIGURE 4.3: The three terms, P, I, and D, of the controller is summed for the controller output.

4.4 The System

To ease the simulation of the system, it has been build in Matlab Simulink. This furthermore has the advantage of being a graphical illustration of the system. But before the system is build in Simulink we make some general system considerations of the systems inputs and outputs and what the limitations of the system are.

4.4.1 Simulink Model of the Car's Propulsion

The main purpose of the controller is to make sure the car drives at least 40 cm from the wall it is driving alongside, as stated in 2.3, Specifications of Requirements. The 40 cm is the minimum reference distance for the controller, but to design the controller from the method described in 4.6, this requirement has to be specified along with other requirements needed for the controller design.

Reference Distance

To make sure this requirement is met, it has to be corrected to for the purpose of the controller. The sensors used has a measuring noise at 40 *cm* of ≈ 5 %, which corresponds to 2 *cm*. To make sure taht the car is 40 *cm* from the wall, the reference distance for the controller is set to42 *cm*.

Maximum Offset

The sensors used have a distance measuring range of $10 - 80$ *cm*. Taking the result of the tests of the sensors into considerations, the effective range of the sensors is $10 - 60$ *cm*. This means that the maximum distance the car can have to the wall, and still expect the controller to work is 60 *cm* and the minimum distance is 10 *cm*. In the design of the controller a symmetric maximum offset is chosen, and the asymmetric offset is considered again in the discussion of the controller. The maximum offset the controller should be able to handle is $60\ cm - 42\ cm = 18\ cm$.

Car behavior

The car ahs to be able to move to the reference distance, from the maximum offset, without detecting objects, which would not have been in the drive path, where the car driving at the reference distance. This behavior is illustrated in Figure 4.4.

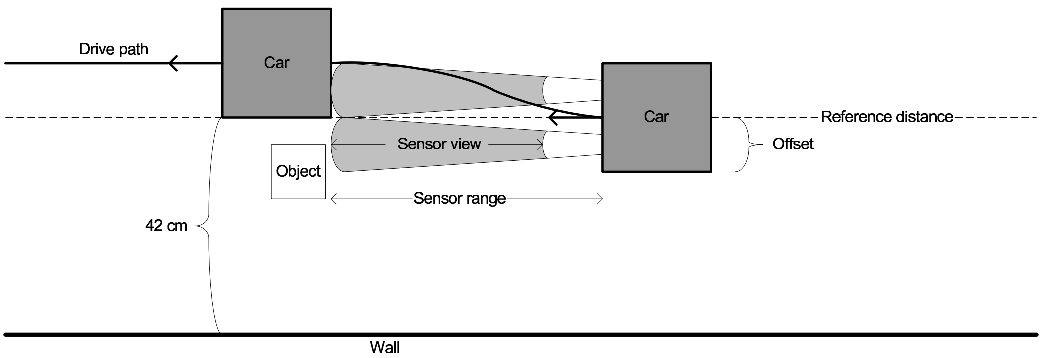


FIGURE 4.4: The car should use a turning motion to get to the reference distance, when it experiences an offset.

To get the car to display the wanted behavior, the propulsion in the direction perpendicular to the wall has to be controlled. This can be done by increasing the angle between the cars driving direction and the wall. the angle is not directly controllable though, unless the car stps to rotate and thereby controls the angle.

The controllable parameters are the average forward speed and the difference between the speeds of the left side wheels with respect to the right side wheels, or vice versa. If the averege forward speed is kept constant, the only controlable input will be the difference in wheel speeds. The car's speed in *rpm* is 49 *rpm*, chosen in 3.3, and the maximum speed is 55 *rpm*. This means that the maximum the wheels can differ from the average forward speed is 6 *rpm*, which is half the limit of the diffrence in wheel speeds, if both sides takes reference to the average forward speed. Keeping the average forward speed constant also means that the system has only one controlable input, the difference in speed of the wheels, in the model called "Turn."

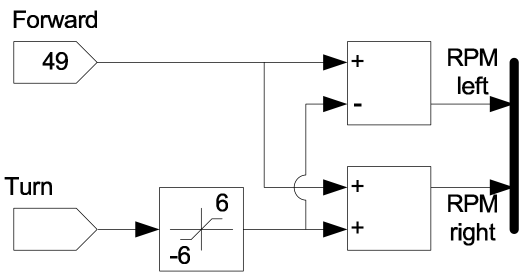


FIGURE 4.5: If the average forward speed is kept konstant the only controlable input is the difference in wheel speed, "Turn."

From the Two inputs in *rpm*, "Forward" and "Turn" can the average speed in $^m/_s$ and the angular velocity of the car e calculated. This is done in a Matlab function with two inputs and two outputs:

```
function r = car(u)

WheelRadius = 0.045;

RMP_Left = u(1);
RMP_Right = u(2);

WheelSpeedLeft = RPM_Left / 60 * 2 * pi * WheelRadius;
WheelSpeedRight = RPM_Right / 60 * 2 * pi * WheelRadius;

% The forwards speed of the car
Speed = (WheelSpeedLeft + WheelSpeedRight) / 2;

% The angular velocity of the wheels
AngularVelocityLeft = - (RPM_Left * 2 * pi) / 60;
AngularVelocityRight = - (RPM_Right * 2 * pi) / 60;

% The angular velocity of the car
AngularVelocity = AngularVelocityLeft + AngularVelocityRight;

r = [Speed; AngularVelocity];
```

How the two outputs, "Speed" and "AngularVelocity," relates to the propulsion of the car is shown in Figure 4.6.

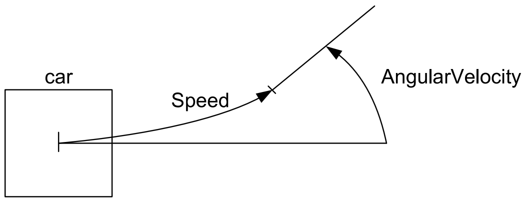


FIGURE 4.6: The realtion between the calculated speed and angular velocity of the car and the cars physical motion.

When the car has no angular velocity, that is the "Turn" input is zero, the only direction affected by the input is that parallel to the wall the car has to drive alongside. The position perpendicular to the wall is the same as its initial distance. The Simulink model is expanded to express this, as shown in Figure 4.7.

When the input "Turn" $\neq 0$ the angular velocity also affects the ouput. Besides the position along the axis parallel to the wall the position along the

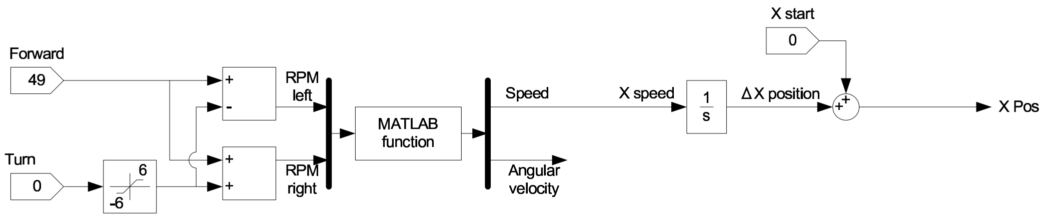


FIGURE 4.7: As "Turn" is zero, "Forward" only affects the output which describes the position on the axis parallel to the wall, "X Pos".

axis perpendicular to the wall is an output, "Y Pos." The value of "Turn" affects both outputs as shown in Figure 4.8. The Simulink model indicates that the propulsion can be controlled directly by using the "Turn" as control signal. The problem with this control is that at high controller gains it may not stabalise, as the changes in y-direction will be faster than the system can respond to. As low controller will however make slow the performance of the system. As such, another way to stabalise the system is sought.

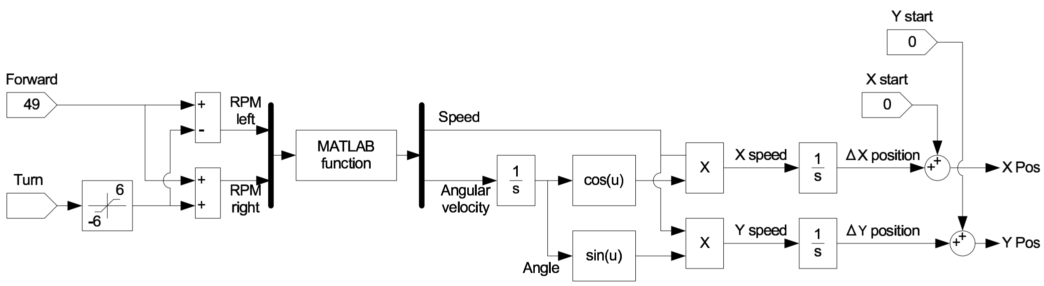


FIGURE 4.8: The complete Simulink model of the car, with four inputs and two outputs. The only controllable input is "Turn."

The idea was to control the y-position by regulating the angle the car has to the wall. This is not directly possible as mentioned earlier. Instead of having the angle as an input, the Simulink model can present it an output, as shown in Figure 4.8. This that given a reference angle will make it possible to have a feedback from the angle to the differens in wheel speed. This feedback is illustrated in Figure 4.9.

The "Y Pos" output was where the feedback where inted to come from, as the angle is not measurable and the "Y Pos" describes the perpendicular distance to the wall. The two outputs has a common characteristic in their wanted

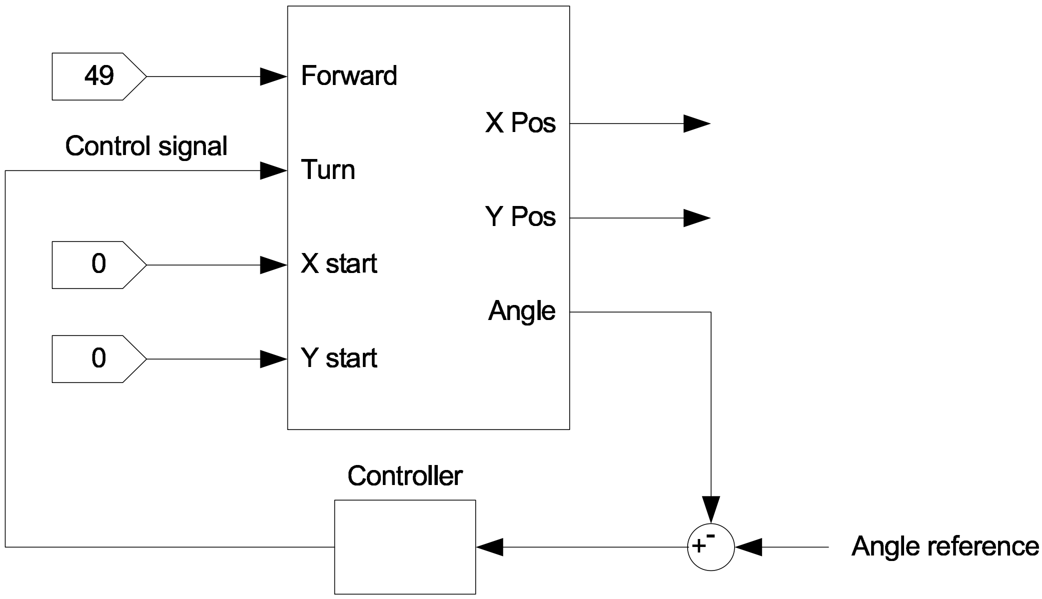


FIGURE 4.9: The control signal is generated by a feedback from the angle which is found within the Simulink model.

response to an offset. Both outputs needs to be increased or decreased to compensate for an offset, depending on whether the the offset brings the car closer to or further away fromt the wall.

As there is a controller between the output "Angle" and the input "Turn," the relation between the distance to the wall and the angle of the car is proportional. The magnitude of the proportional gain is proportioned from consideration on the maximum values of the angle and the distance. The angle is in radians, which means it will have a maximum of 1.5 rad, which coresponds to an angle of $\approx 90°$. The distance has a maximum of less than two meters, which is in the same range a the angle. As such, the proportional gain is chosen to be 1. The feedback is illustrated in Figure 4.10.

4.4.2 Simulink Model of the Car's Sensors

The "Y Pos" output from the car's propulsion model is the perpendicular distance to the wall, the real distance measured by the sensor is found as a function, $f(YPos, Angle)$. Therefore is a simulink model of teh sensor used fore control purposes made. In the model is the 5 % noise of the sensor added to the output, and as the sensor only is reliable within the range $10 - 60$ cm a rate limiter is added as well. The model and the basic idea with the function

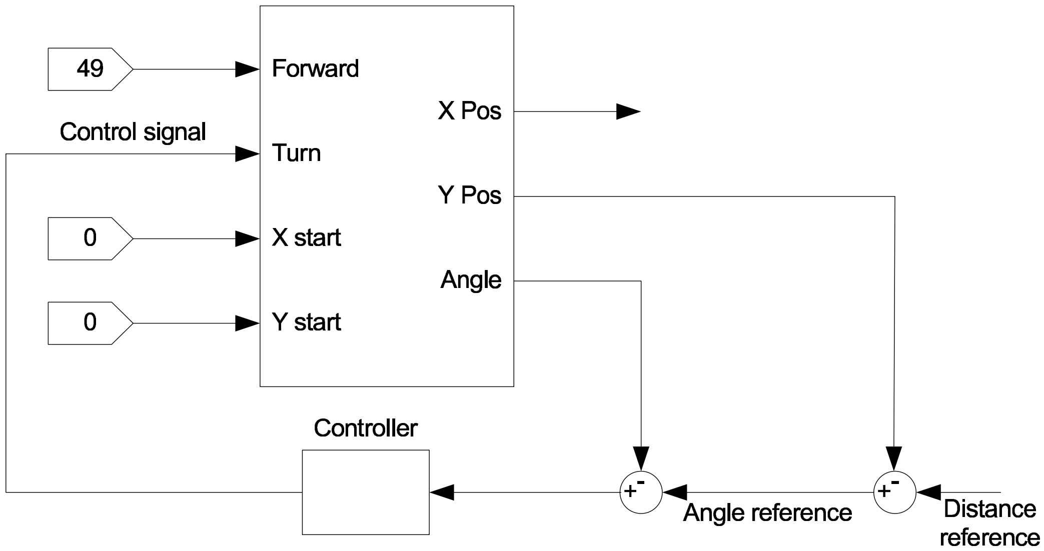


FIGURE 4.10: Two controllers are implemented in teh Simulink model, one for centrolling the input "Turn" and one giving a reference to the angle of the

f is illustrated in Figure 4.11.

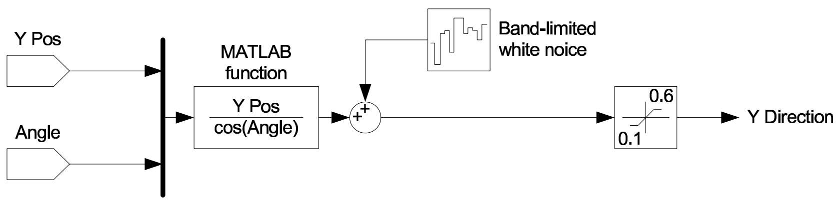


FIGURE 4.11: The real distance measured by the sensor is a function, $f(YPos, Angle)$. further more noise and the limits of the sensor is added to the model.

4.5 Modeling

To explore the opetunities of using model based control for the car, a part of the system is seperated and a model of that part is made. The part used is the Simulink model of the car propulsion, found in 4.4.1. This is a an unstable non-linear system. Therefore the system has to be linerised. The linearisation is done in Matlab with the command `linmod`, which has the A, B, C, and D matrix of the equivalent state space model as output. The state vector, x, and output vector, y, are the same: [X pos, Y pos, Angle]$^{\mathrm{T}}$. The input to the original Simulink model is the difference in *rpm* that affects the angular velocity of the car. The model uses an angle as input, as this would be more logic to control, as it in practice are easier to relate an angle than a difference in *rpm* to how much the car needs to turn. As the propulsion system is proportioned, is the conversion from *rpm* to an angle $1 : 1$, as any differece likely is proportional and therefor can be implemented in a controller. The state space model is expessed as:

$$\dot{\overline{x}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.47 \\ 0 & 0 & 0 \end{bmatrix} \overline{x} + \begin{bmatrix} 0 \\ 0 \\ 0.21 \end{bmatrix} \mathrm{u}$$

$$\overline{x} = [''Xpos''\ ''Ypos''\ ''Angle'']T \tag{4.13}$$

The A matrix reflects that only the angle affects the "Y pos"-state. The B matrix, which is the input matrix, only has an effect on the "Angle"-state, which corresponds to the fact that the control signal for the system is an angle.

$$\overline{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \overline{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \mathrm{u}$$

The C matrix is the Identity matrix, as the output and state vectors are equivalent. The D matrix contains only zeros, as there are no feedforwards from the input to any the outputs. The only output used to design the controller is "Y pos," which is the second row in the output vector. Therefore the second row of the C and D matrixes are isolated. This means that the system is a Single Input Single Output (SISO) system. The system transfer function is found by use of Matlab and is expressed in equation 4.14.

$$G(s) = \frac{0.1}{s^2} \tag{4.14}$$

4.6 Design Method

Two design methods are considered, root locus which is based on the transfer function of the feedback system, and Ziegler-Nichols tuning of PID regulators is based on the step response of the system that has to be controlled. The advantage of root locus is that it is possible to visually see the response of the system when zeros and poles are fitted. The disadvantage is that it needs a model. The advantage of Ziegler-Nichols method is that with a step response of the system the parameters for the PID controller is calculated. The disadvantage of this method is that the controller is allowed an overshoot. The theory used in the following two sections are based on [Franklin et al., 2002].

4.6.1 Ziegler-Nichols Tuning of PID Regulators

Ziegler and Nichols gave two methods for tuning the PID controller for a system that has a reaction curve when a step is applied as shown in Figure 4.12. The method considered is the tuning by decay ratio of 0.25 method. This means that the transient decays to a quarter of its value after one period of oscillation.

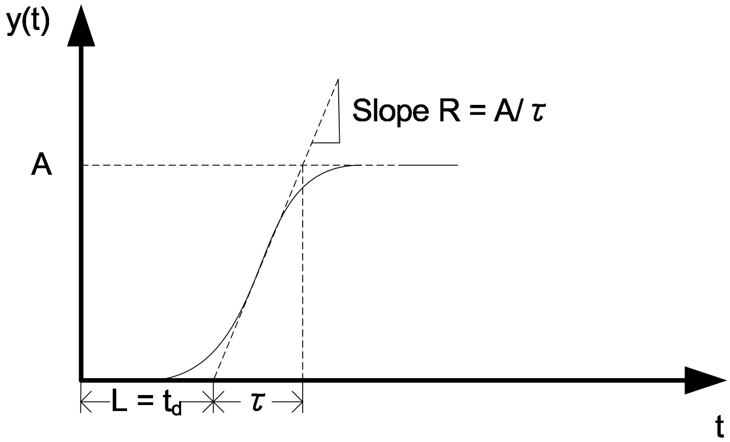


FIGURE 4.12: Reaction curve behavior needed for Ziegler-Nichols tuning by decay ratio of 0.25 method

The tangent with the highest slope of the curve is called the reaction rate and is defined as $R = A/\tau$, and if the tangent is followed down to the time axis the time delay $L = t_d$ is identified. The PID controller can be written in the following form,

$$D(s) = k_p \left(1 + \frac{1}{T_I s} + T_D s\right)$$

where Ziegler and Nichols suggests the following parameters

$$
\begin{aligned}
k_p &= \frac{1.2}{RL} \\
T_I &= 2L \\
T_D &= 0.5L \\
k_i &= \frac{k_p}{T_I} \\
k_d &= K_p T_D
\end{aligned}
\tag{4.15}
$$

4.6.2 Root Locus

Root locus is based on the closed loop transfer function of the feedback system shown in Figure 4.13 which can be described as in equation 4.16,

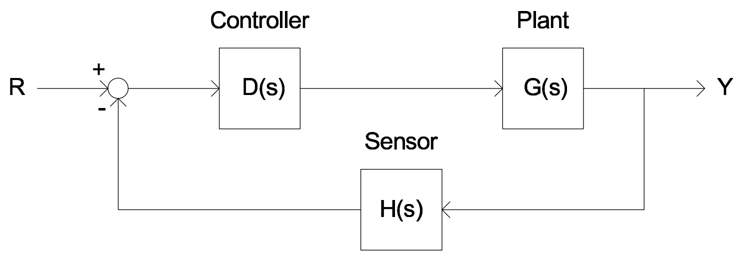


FIGURE 4.13: Block diagram of the ideal feedback system.

$$T(s) = \frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)H(s)} \tag{4.16}$$

where the characteristic equation for the poles can be described as in equation 4.17.

$$1 + D(s)G(s)H(s) = 0 \tag{4.17}$$

To be able to study the effects of altering a parameter, equation 4.17 is put in polynomial form, and the parameter of interest is chosen and named K. It is assumed that it is possible to define component polynomials $a(s)$ and $b(s)$ such that the characteristic polynomial can be described as $a(s)+Kb(s)$. The transfer function $L(s) = b(s)/a(s)$ is defined such that the characteristic equation can be written as

$$1 + KL(s) = 0 \tag{4.18}$$

To find the best values of K the locus of all possible roots of equation 4.18 are plotted as K varies from zero to infinity, and the resulting plot is then used to select the best values of K. The plot can also be used to analyze the effect of adding zeros and poles in $D(s)$ as compensation in the loop. This can be used both to select a specific parameter and designing the dynamic compensation. The graph of all possible roots of equation 4.18 relative to the parameter K is called the root locus.

The roots of $b(s) = 0$ are the zeros of $L(s)$ and the roots of $a(s) = 0$ are the poles of $L(s)$. The roots of the characteristic equation r_i are found from the factored form

$$a(s) + Kb(s) = (s - r_1)(s - r_2)\dots(s - r_n) \tag{4.19}$$

Equation 4.18 can be written in equivalent ways which has the same roots. These are called root-locus form and are shown in equation 4.20.

$$\begin{aligned} 1 + KL(s) &= 0, \\ 1 + K\frac{b(s)}{a(s)} &= 0, \\ a(s) + Kb(s) &= 0, \\ L(s) &= -\frac{1}{K} \end{aligned} \tag{4.20}$$

The root locus is the set of values of s for which equation 4.20 hold for some positive real value of k.

The SISO Design Tool in Matlab is able to draw the root locus, and makes it possible to fit poles and zeros for a single input single output system.

4.7 Requirements for the Controller

4.7.1 Rise Time

When the car experiences an offset it has to move toward the reference distance. It has to reach the reference distance in time to avoid detecting obstacles that would otherwise not have been detected by the front sensors. That is obstacles that are place in the drive path if the car was driving the 42 *cm* from the wall. The maximum measuring range chosen for the sensors is 60 *cm*, this means the car has to have reached the reference distance when

when it has driven 60 cm. The meaning of this is furthermore illustrated in Figure 4.14. The speed of the car is defined in 3.3, Movement, to be 23 ${}^{cm}/_{s}$. The rise time is derived from the distance the car maximum should use to reach the reference distance, and the speed of the car, in equation 4.21.

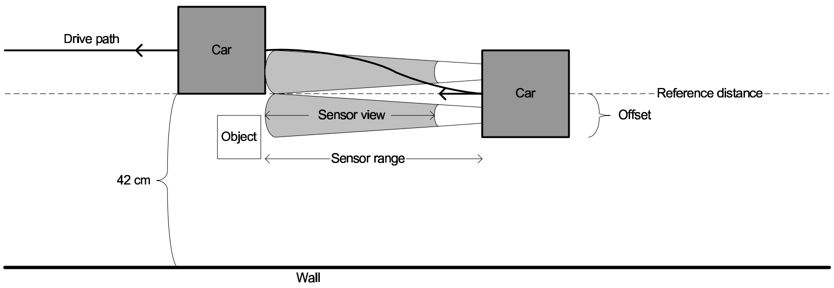


FIGURE 4.14: The wanted behavior of the car, with the controller implemented.

$$t_r = 0.8 \cdot \frac{60\ cm}{23\ {}^{cm}/_{s}} = 2.1\ s \tag{4.21}$$

4.7.2 Settling Time

The settling time, t_s, of the controller is defined to be when the car comes within two centimeter, which will assure that the requirement for minimum reference distance is met.

4.7.3 Maximum Overshoot

The presence of an overshoot, M_p, is unwanted. Even so, a maximum allowed overshoot is defined as 2 cm, which corresponds to the maximum deviation allowed when the controller has settled. The maximum allowed overshoot, in percentage, is derived in equation 4.22

$$M_p = 100 \cdot \frac{2\ cm}{18\ cm} = 11\ \% \tag{4.22}$$

From these facts the wanted behavior of the car with the controller is illustrated in Figure 4.15.

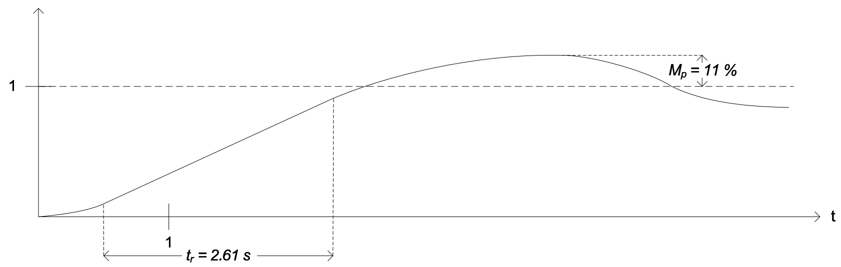


FIGURE 4.15: The wanted behavior of the controller.

4.7.4 Phase Margin

For a controller to be stable a minimum phase margin PM of 45° is needed.

4.8 Controller Design

4.8.1 Ziegler-Nichols Tuning of PID Regulators

As the system is unstable, a proportional controller is inserted to stabilize the system. For this controller the unity feedback is used. This combined system is added a step, to get a step response, and the result is shown in Figure 4.16. The amplitude of the step is $A = 1$, and from the tangent both $L = 1.21$ and $\tau = 5.26$ were found. Now R can be calculated, and then the controller parameters can be calculated too.

$$\begin{aligned}
R &= \frac{A}{\tau} = \frac{1}{5.26} = 0.19 \\
k_p &= \frac{1.2}{RL} = \frac{1.2}{0.19 \cdot 1.21} = 5.2 \\
T_I &= 2L = 2 \cdot 1.21 = 2.42 \\
T_D &= 0.5L = 0.5 \cdot 1.21 = 0.61
\end{aligned}$$

T_I and T_D are converted into integral gain k_i and derivative gain k_d respectively in equation 4.24.

$$\begin{aligned}
k_p &= 5.2 \qquad (4.23) \\
k_i &= \frac{k_p}{T_I} = \frac{5.2}{2.42} = 2.15 \\
k_d &= K_p T_D = 5.2 \cdot 0.61 = 3.17
\end{aligned}$$

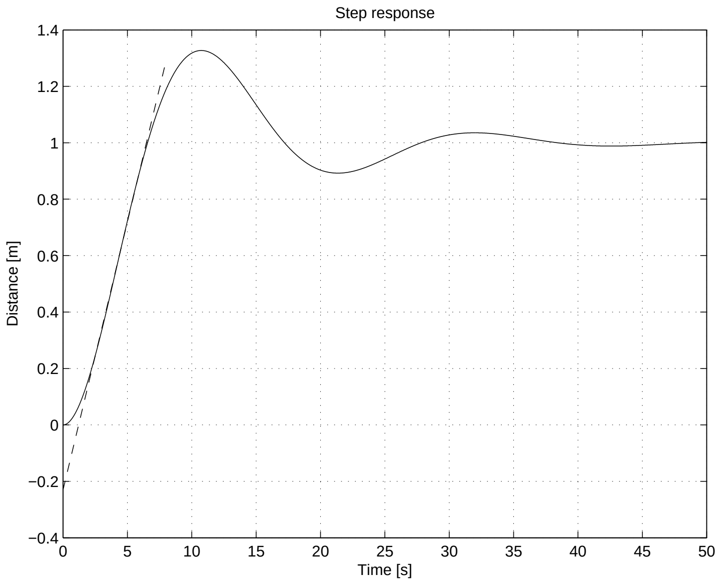


FIGURE 4.16: Step response of the system with a proportional controller and unity feedback.

A PID controller were applied to the system with the parameters from equation 4.24, and the result is shown in Figure 4.17.

The black curve in Figure 4.17 is the system response, and the broken curve is the Ziegler-Nichols PID controller. The PID controller reacts as expected, but falls under the reference after a small overshoot of about 5%. This fall is not optimum, therefore the controller is hand tuned. The result of the hand tuned Ziegler-Nichols PID controller is shown in Figure 4.18, and the parameters found are listed in equation 4.24.

$$\begin{aligned} k_p &= 5.2 \\ k_i &= 0.215 \\ k_d &= 3.17 \end{aligned} \tag{4.24}$$

The hand tuned Ziegler-Nichols controller has a bit slower rise time and a bit higher overshoot, but nothing to mention. The fall after the overshoot is neglected, which is considered more important than the change in the rise time and overshoot. This means that the hand tuned controller is used, and tested on the system with the real sensor measurements, meaning that the car does not drive perpendicular to the wall when driving to the reference

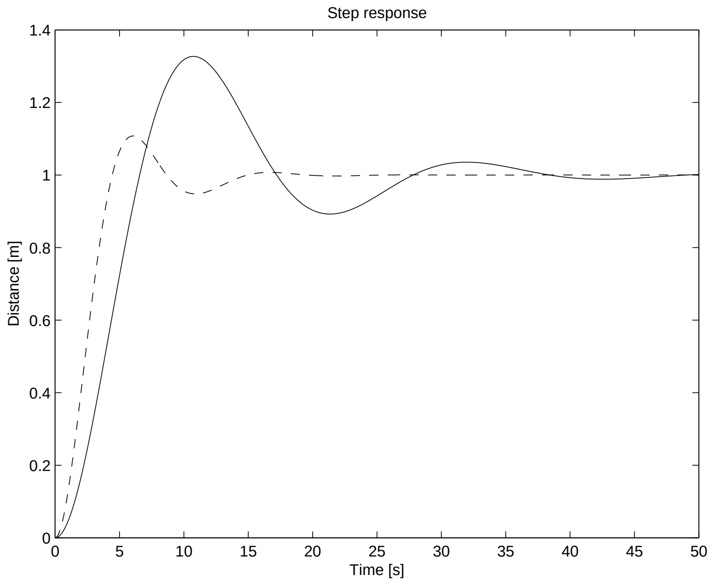


FIGURE 4.17: Step response with the Ziegler-Nichols PID controller.

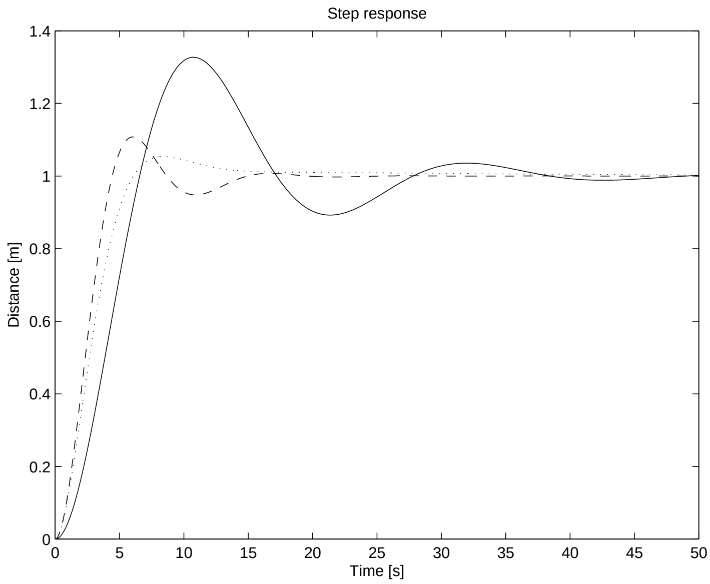


FIGURE 4.18: The step response of the system is the black curve, Ziegler-Nichols PID controller is the broken curve, and the hand tuned Ziegler-Nichols PID controller is the dotted curve.

distance to the wall. The result of this simulation is shown in Figure 4.19, and is how the controller is expected to perform in a practical test.

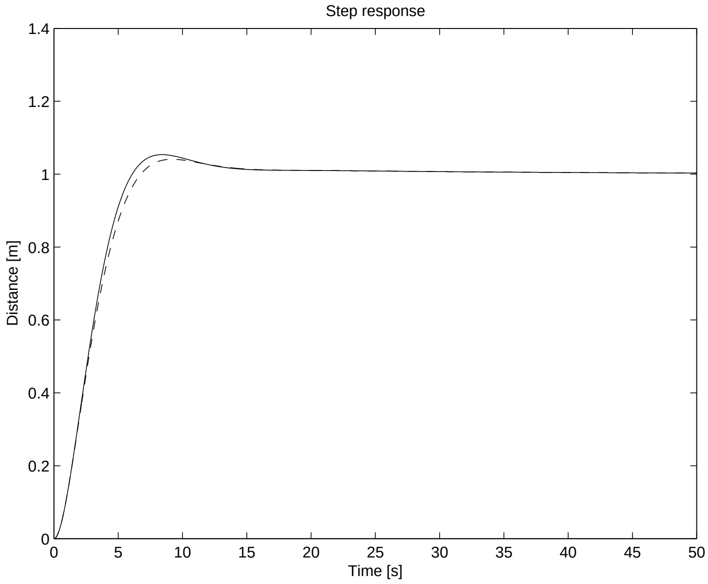


FIGURE 4.19: Step response of the system with sensor dynamics, where the black curve is the controller with the real distance to the wall, and the broken line is the controller with the expected sensor measurements.

Figure 4.19 shows that the hand tuned Ziegler-Nichols controller is expected to perform well in a practical test, but the step size has to be scaled to a distance the sensors are able to detect, which has an maximum of 18 *cm* from the reference of 42 *cm*. In other words as the sensors can not measure distances above 60 *cm*, this is the reference plus 18 *cm*.

4.8.2 Root Locus

The model from 4.5 Modeling is imported in the SISO Toolbox, and a zero is inserted which is illustrated in Figure 4.20. The compensation $C(s)$ is found with a infinity gain margin, and a phase margin of 76.8° which is above 45° and the system is thereby stable.

$$\begin{aligned} C(s) &= 3.5(1+3.5s) = 3.5 + 3.5s \Rightarrow \\ k_p &= 3.5 \end{aligned} \tag{4.25}$$

$$k_d = 3.5 \cdot 3.5 = 12.25$$

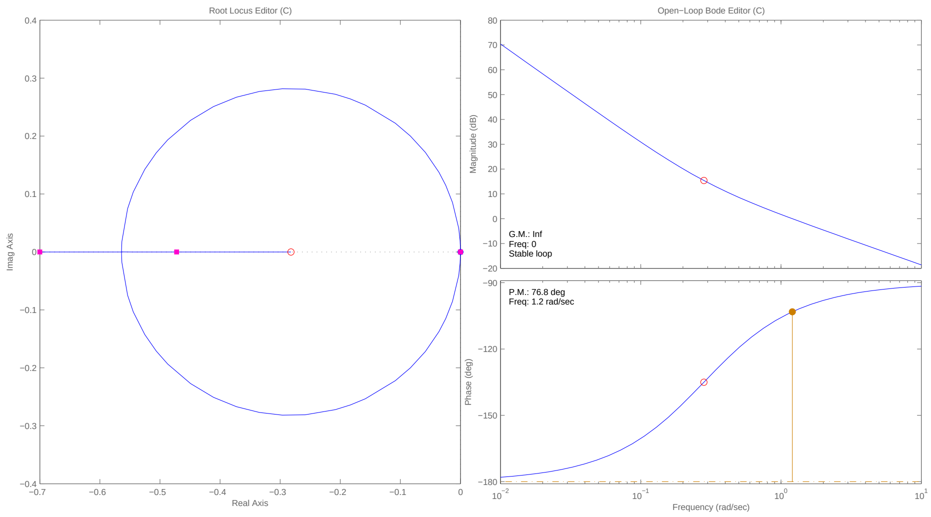


FIGURE 4.20: Root locus of the linearized car model.

Equation 4.25 shows that with root locus a PD controller is sufficient. The step response of the system with the controller from equation 4.25 is shown in Figure 4.21.

4.9 Discussion

The controller chapter had three primary objectives, the first was to get a reasonable model imported in Simulink, the secont was to test the models performance including for realizability, and the third was to test the found controllers for the model, and decide which was the best.

4.9.1 The Simulink Model

The found Simulink model was nonlinear, which meant that it had to be linearized. This was done with the Matlab function LINMOD, but in the linearization some dynamics were lost, that meant that the model was not accurate enough. Furthermore the model was unstable, therefore a non model based feedback were introduced to stabilize the model. These two factors influences on the model based controllers performance.

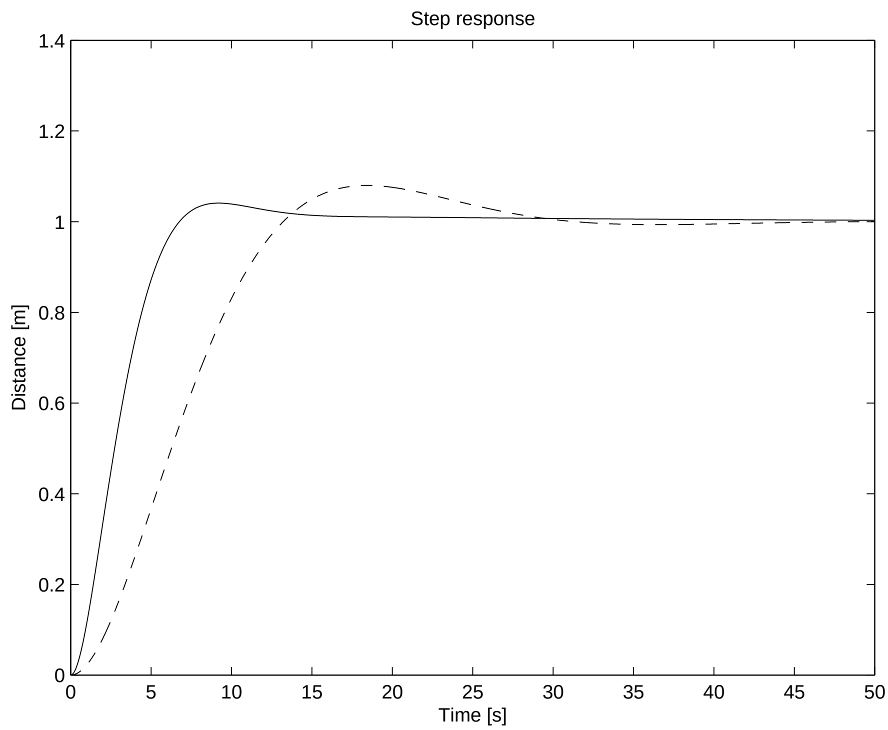


FIGURE 4.21: Step response of the system where the black curve is the Ziegler-Nichols controller, and the broken curve is the root locus controller.

4.9.2 Car Model Verification

A weakness of the chosen car model is that the angle θ is used in the feedback, as it is not observable with the sensors mounted on the car. This means that the controller used is not directly realizabl, but if another sensor is mounted on the side of the car, the angle could be calculated. Another possibility is to mount a compas on the car, but both solutions are based on extra hardware, which will increase the weight of the car, that is preffered as low as possible to make the car more robust to the surroundings. A third option is to use an observer that calculates the angle from the turn input to the system.

4.9.3 Controller Choise

The two controllers designed in this chapter, the root locus and the Ziegler-Nichols tuned PID controllers perfomances are compared in Figure 4.21.

As Figure 4.21 shows, the Ziegler-Nichols hand tuned PID controller performs best, which is because it does not use a linearized model with the loss of system dynamics. If the controller is wanted to be implemented on the real system, the controller has to be able to handle the real sensor data, which can be described as in Figure 4.22.

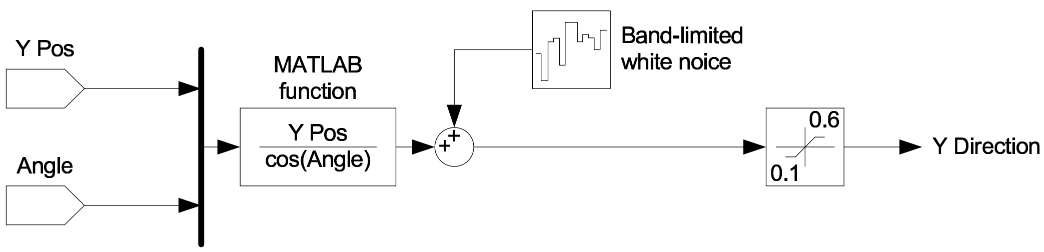


FIGURE 4.22: Modification of the sensor model.

4.10 Conclusion

As mentioned in the 4.9 Discussion the Ziegler-Nichols hand tuned controller performs best, and are now compared to the controller requirements.

4.10.1 Rise Time

The required rise time from equation (4.21) were found to be $t_r = 2.1\ s$. Figure 4.21 shows that the system has a rise time of $t_r = 4.25\ s$, which violates the requirement.

4.10.2 Maximum Overshoot

The overshoot of the system can like the rise time be found from Figure 4.21. The maximum overshoot from equation 4.22 is $M_P = 11\ \%$. The overshoot is found to be $M_P = 4\ \%$, then the requirement is respected.

4.10.3 Phase Margin

To have a stable system, it has to have a phase margin of at least 45°, but as the system is simulated to be stable, the phase margin is not found.

From the presented fact it is concluded that the overall performance of the controller is acceptable.

Chapter 5

Discussion

Contents

The motivation for the project is to construct and design an autonomous car, able to make a video map of an unknown room. This motivation spawned from the increasing autonomity we are surrounded by today. The subject artificial intelligence, which autonomity is a part of, has high enough priority in this age that it is on the agenda of the Ministries of Science.

This interest and the introduction to IARC brought up thoughts on how to make an autonomous robot car, which is light enough to be carried by another autonomous robot, a helicopter, robust enough to sustain impact with a floor after being dropped from one meters height, and intelligent enough to find the dimensions and make a visual map of the room it is send into.

The car's main purpose is to make a video map of a room with unknown dimensions and an unknown quantity of obstacle, such as furniture. Other types of robots were considered. For instance, another aereal vehicle, but this would introduce other issues such as a more complex position and movement control, which would have to be active during the entire operation. The advantage of the car in accordance to the movement control, is that the car can be given an reference position and upon reaching this position it can enter a idle mode. If the aereal subvehicle were to enter idle mode, it would simply fall to the ground and most likely not survive the impact with the floor.

Finding the dimensions and ensuring that the entire room has been mapped is where the car's autonomity and intelligence has to prove its worth. Though the task of finding the room dimensions is ended at a predefined trigger, based on the numbers of corners in the room, the car has enough autonomity to determine whether the room dimensions makes sense. After finding the room dimensions, the rest of the mapping is performed based on the car's gathered information, on the lengths of the walls and the positions of the corners.

During the mapping the car can encounter both open doors and furniture. These are the two sorts of obstacles we have analysed, as these cover most of what the car has a chance to encounter, when it is send into an office, as in accordance to IARC, hostage rescue. The handling of these obstacles is introduced with feedforwards, which naturally introduces some uncertainties regarding the precision of the car movements. To compensate for these obstacle while driving alongside a wall it is preferable that the controller can compensate for these uncertainties.

5.1 Hardware

The main purpose with the hardware chapter is to design and construct a car capable of making a video map of a office like room. To make the video map, a wireless spy cam was found on the internet. The advantage of the chosen camera is that it is light weight and that it has a compact size. To have the car display the right behavior, in accordance to the analysis, sensors and μ-servos were found, to make the car able to sense its surroundings and move around.

With the electronics on the car such as the camera, μ-servos and micro computer a power supply is needed. A power supply were designed that could keep the car active for about 72 *minutes*, which is expected to be enough to map a room. The disadvantage of the power supply is the weight. It might be possible to design a more light weight supply. For instance, a cell phone battery is used instead.

Another subject considered in the hardware chapter is the robustness of the car. The car has to be able to sustain hitting the floor from the height of a window, which is assumed to be a one meter fall. The robustness of the car was limited to find light weight components, to lessen the impact force. No suspension were added to the car, and no tests were conducted to find out if the car actually could sustain the impact on the floor.

5.2 Controller

In the controller chapter the methods for making controllers for the car was discussed. The first thing discussed was whether to use non-linear control if the car system is non-linear, but as there are few proven methods, this is not the preferable way to design a controller. The system is non-linear and unstable, to cope with the first issue, the non-linearity, it was described how to linearise a system in an operating point.

The discussion of whether to use model or non-model based control are initiated by an description and a discussion of what a model is and why it may be preferred to have a model of the system rather than making nonmodel-based control. Some of the advantages of a model is that it is easier to analyse for stability, and it gives the designer a larger selection of controller design methods.

Regardless of whether the control is model-based or not, the classic control can be used. For the purpose of this project the classic PID control has been chosen, as it is expected to be usable for the regulation needed and as this is a simple form of controller based on linear systems.

For the controller to compensate for an offset, which is deviations along an axis perpendicular to the wall the car is driving alongside, it will have to give the car propulsion towards the reference distance. This propulsion cannot be applied directly in the wanted direction, but can be increased by increasing the angle the car has to the wall. The model was made in Matlab Simulink and describes the relation between the wheels speeds and the position in both the x and y direction, as well as the angle the car has to the axis parallel to the wall.

We chose to make both a model and a nonmodel based controller and compare the results to see which is the better for our system. Both controller are PID's, but two different controller design methods are used.

The model based controller is designed using Root-Locus on the state space model, found using Matlab. The controller designed is a PD controller.

For the design of the nonmodel based controller the system had to be made stable first. This meant making a feedback loop. The first feedback made was from the angle to the input describing the turn speed. This stabilised the system, but the initial idea was to have the position along the axis perpendicular to the wall in the feedback, as this describes the reference distance. But as the initial idea was to regulate the distance to the wall by use of the angle, the wanted characteristic for both the angle and the distance to the wall was analysed. The result was a proportional gain between the two outputs. This meant that the position is used to find the reference for the angle. This setup is not without issues though. The angle is a calculated parameter and is not measured like the distance to the wall is. To compensate for this an extra sensor has to be implemented, a place where it can give an reading on the car's angle to the wall. Another way to do it is by adding an observer, but this would require the making of a model, as controller can only be used with controllers. The design method used for the nonmodel-based controller is Ziegler-Nichols' tuning by decay ratio of 0.25. The controller designed is a PID controller.

The two controllers have been held up against each other and the nonmodel-based controller showed the best performance. A reason for this might be that the model-based controller is based on a linearized model of the nonlinear system. Looking at the nonmodel based controller and comparing this to the requirements, it is deemed acceptable, though the rise time for it is more than twice as large as the required, does it have a nice trajectory and shows little overshoot.

Putting the qualities of the controller in perspective of the car driving alongside the wall. The slower rise time will at this perspective result in that the car will detect objects, which would not be in its drive path, were it driving at the reference distance. As a result of this the car will have to enter more feedforwards along the way around in the room. This is unwanted, as the most precise dimensioning of the room will be conducted if the car can drive between the corners, using the feedback all the way. The reason that the controller is deemed to display an acceptable performance anyway, is that the furniture legs which in this case would be the cause of the extra feed-forwards might as well be placed about 42 cm from the walls as the can be place closer to the walls.

Chapter 6

Conclusion

The problem formulation states that a car has to be designed and constructed so that it, when put into an unknown shaped room, can make a complete visual map of the room.

This holds three major challenges: To design and construct a car which can sense and film its surrounding. Find its way in an unknown room and making the dimensions of the room known. And when the room dimensions is known making a complete video map of the room.

As a frame for the project IARC has been chosen, as this competition allows us to make some reasonable assumptions on the room in question. In brief, the assumption is that the room is an office and is for instance, limited to having 90° corners.

The parameters affecting the car construction, as well as those affecting the behavior of the car, has been analysed. Concerning the car design and construction, the analysis has divided the mapping procedure into tasks and capabilities the car needs to posses for it to be able to move around in the room. The wanted capabilities of the car has then been used to analyse how the car make its way around in the room to make sure it gets the best video coverage of the room walls, interior, and persons within it.

Each task the car should be able to perform has been analysed, as to why it has to perform the task, and how performing the task affects the overall performance of the car. The tasks has furthermore been analysed as to how these can be simulated, and tested whether the car is capable of performing the task and what the criteria for the car to succeed. These tests is specified as accepttests.

None of the accepttests has been performed, neither the simulation part nor the test part. Though, a controller has been designed and simulated with the cars physical limitation, such as the forwards speed of the car. This controller can be used for the simulations of the accepttests, but will require further development of the car or the controller, if it should be implementable in the car's physical system.

With the controller implemented, either in simulation, test or both, the car can perform some of the tasks and gain some of the capabilities required of it to complete the mapping procedure.

The way the car is to dimension the unknown room is by driving alongside the walls until it reaches its initial position. For this to be performed with only small deviances in the measuring of the lengths of the walls the car will need the controller to keep a constant distance to the walls, and thereby make it easier to estimate the room dimensions.

This leads to the conclusion that the controller cannot be used, without further development, by the physical car to perform the needed tasks to dimension and film an unknown office-like room, but in simulations the controller can be implemented in the car and help improve the performance of the car, which means that it indirectly improves the mapping procedure.

Appendix A

Up/Down Sensor

Contents

A.1 Introduction

During its fall, the car may flip and thereby land upside down. Therefore it has to identify which side is up. To do this a sensor is placed facing downwards, initially, as illustrated in Figure A.1.

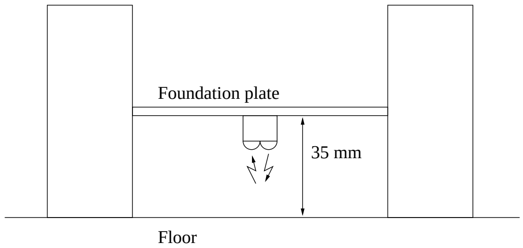


FIGURE A.1: The sensor, used to detect the floor, is initially facing downward.

The idea is to measure the distance from the bottom of the car to the floor and thereby determine whether the car has flipped. In practice a threshold is defined and if the distance is greater, the car has flipped.

A.2 Requirement Specification

Distance measuring: From the bottom of the car to the floor is an estimated 35 *mm* due to the wheel diameter. The difference in sensor output, depending on distance, has to be distinct.

Weight: As with the design of the rest of the car, the sensor has to be light weight.

A.3 Possibilities

Of the sensors available, two has been chosen. These are tested with respect to the two requirements.

SHARP GP2D15. This is an IR sensor. It has an internal distance measuring IC and a connector, with supply, ground, and output signal pins. These features eases the integration with other hardware components. The output is digital, which means that the output, while closer than a predefined distance, is high and low, when not.

The distance interval of the GP2D15 is $10 - 80$ cm.
The weight of the GP2D15 is 3.6 g.

Optek OPB704. This is also an IR sensor, but it consists only of a LED and a photo transistor in a casing. To control the current through the LED, an external circuit will have to be designed.
The distance interval of the OPB704 is $5 - 20$ mm.
The weight of the OPB704 is 1.4 g, plus any extra components needed.[OPB704-extra.pdf]

A.4 Test of a GP2D15

The test of the GP2D15 is conducted to determine whether, or how, suitable it is for the purpose of detecting whether the car is turning upside down.

The following is used for the test:

	Manufacturer	Model	Lab.No.
Power Supply	HAMEG	HM7042-3	52755
Oscilloscope	Agilent	54621D	52772

GP2D15 circuit: According to the data sheet, the GP2D15 needs a pull up resistor of 12 $k\Omega$. The circuit is illustrated in Figure A.2.

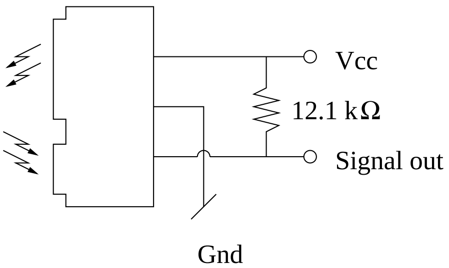


FIGURE A.2: GP2D15 circuit.

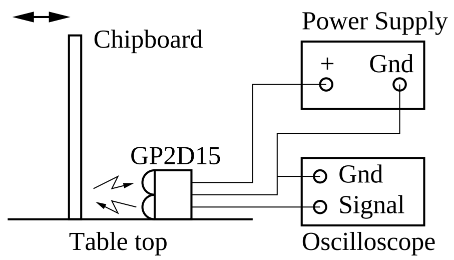


FIGURE A.3: GP2D15 test set-up.

A.4.1 Test Set-up

The test was conducted by connecting the GP2D15, supply and ground, to the power supply and the GP2D15, signal and ground, to the oscilloscope. The sensor was then placed on a table with the detection direction parallel to the table. A chipboard was used to act as a floor and was placed at 16 different distances from the sensor, starting with 0 mm and increased with a 50 mm interval. The test was performed three times at different times in the day, at 9am, 1pm, and 4pm. Figure A.3 illustrates the test set-up.

The test was expected to show that above a threshold, ≈ 240 mm, the output signal is low and below the threshold the output signal is high.

A.4.2 Results

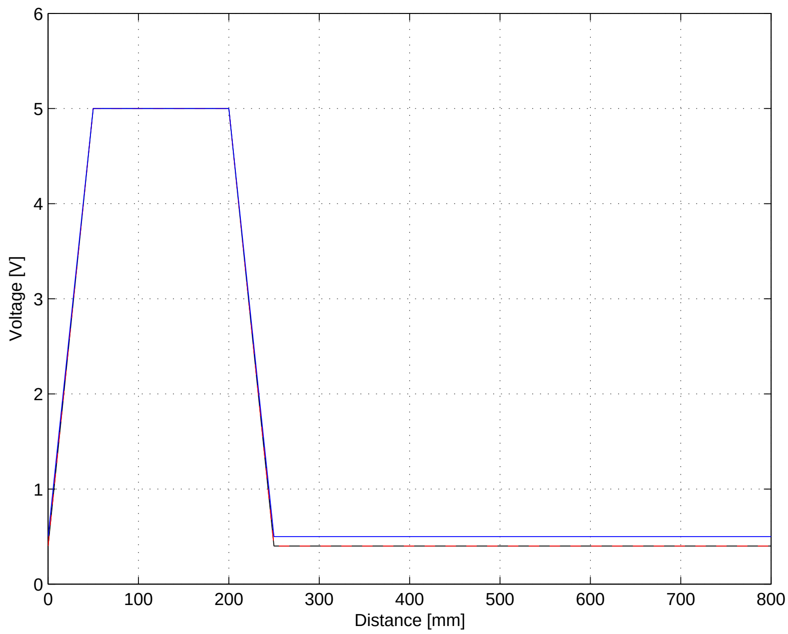


FIGURE A.4: Test of the GP2D15 IR digital distance sensor.

A.4.3 Discussion

The test result, illustrated in Figure A.4, shows that the sensor acts as expected within its range of operation, $10 - 80$ cm. Furthermore the sensor is working acceptable consistent at varying light effects. At 0 cm the GP2D15 sensor detects no object though, just as it would if an object was at a distance above the threshold. Therefore an additional test were performed where the chipboard started at 0 mm and moved away from the sensor 10 times, with an interval of 5 mm. The test results from this is shown in Figure A.5.

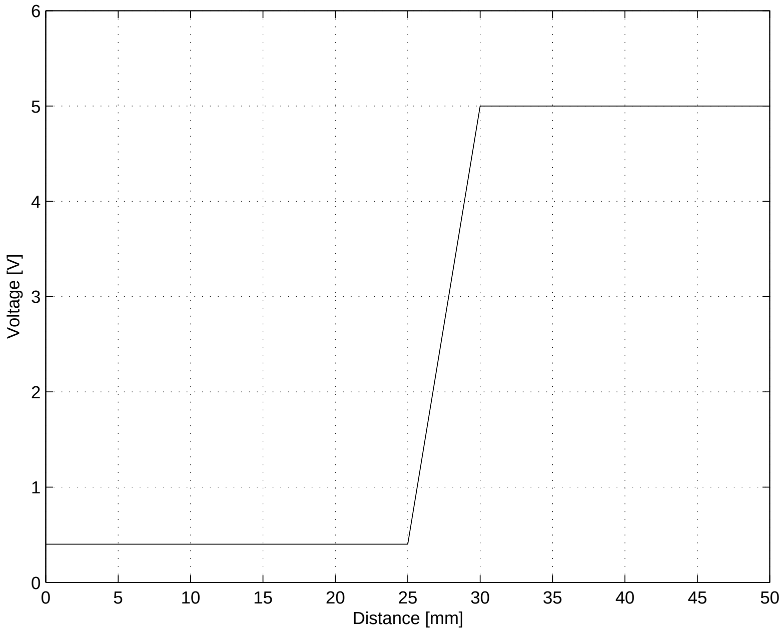


FIGURE A.5: Result of the additional test of the GP2D15 sensor.

The results of the additional test shows that the output signal is low at 25 mm, which is the same as at maximum distance (35 mm minus the sensor height) and therefore the GP2D15 sensor is deemed not usable.

A.5 Test of a OPB704

This test were conducted to find the variation in the output signal depending on distance.

The following test equipment were used:

	Manufacturer	Model	Lab.No.
Power Supply	HAMEG	HM7042-3	52755
Oscilloscope	Agilent	54621D	52772

OPB704 circuit: To gain a useful output from the sensor an external circuit is needed. The circuit used, is illustrated in Figure A.6[1]. The basic idea is to have a low output when the sensor is detecting the floor and high otherwise. Therefore a resistor connects the output with Vcc.

Furthermore there is a need to control the current through the LED. The current wanted is 20 mA. A resistor is serial connected from Vcc to the anode on the LED. The voltage drop across the LED is 1.6 V. Resistor dimensioning:

$$R = \frac{(U_{Vcc} - U_{LED})}{I} = \frac{5\ V - 1.6\ V}{20\ mA} \approx 180\ \Omega$$

The emitter of the photo transistor and the cathode of the LED is both connected to ground.

A.5.1 Test Set-up

The test was conducted by connecting the power supply and the oscilloscope to the circuit. The sensor was then placed on a table with the detection direction parallel to the table. A chipboard was used to act as a floor, as seen in Figure A.7, and was placed at 11 different distances from the sensor, starting with 0 mm and increased with a 5 mm interval.

The test was expected to show that above a threshold, $\approx$ 15 mm, the output signal is high and below the threshold the output signal is low.

[1]The circuit was found on the Internet, at http://www.roborugby.org/optical.html

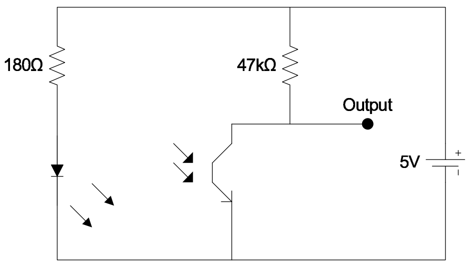


FIGURE A.6: OPB704 circuit.

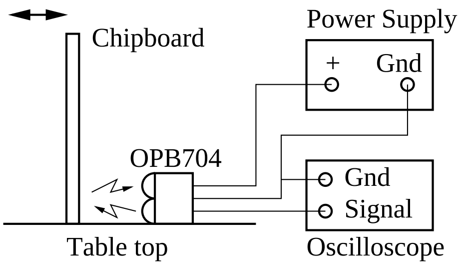


FIGURE A.7: OPB704 test set-up.

A.5.2 Results

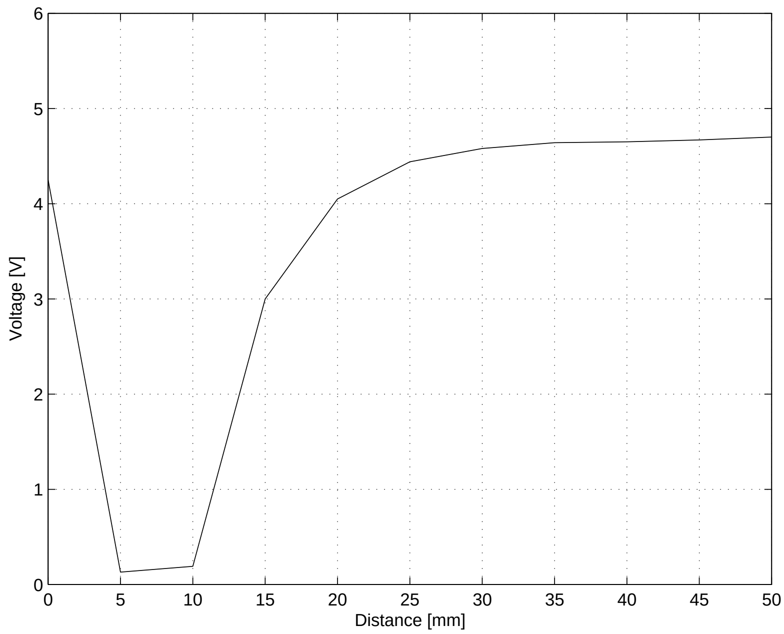


FIGURE A.8: Test results from OPB704.

A.5.3 Discussion

The test results, illustrated in Figure A.8, show that there is a difference in the signal output depending on whether the distance is 15 mm or more than 20 mm. The difference is not as distinct as preferred. This means that the sensor will have to be lowered $5 - 10$ mm from the bottom of the car.

The total weight of the OPB704 and circuit is 2.4 g, which is acceptable.

The OPB704 sensor is deemed usable.

A.6 Conclusion

The tests of the two sensors were conducted to find the better one and with the requirements stated in the beginning of this appendix, only one of the sensors are usable, the OPB704. Another fact to back up this choice is that the OPB704 is the most light weight of the two sensors.

Appendix B

Distance Sensors

Contents

To find its way in the room, the car has four distance sensors mounted. Two pointing forwards and one pointing to either side, as shown in Figure B.1.

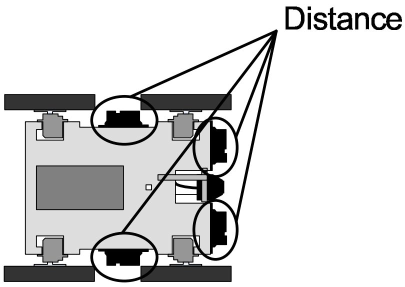


FIGURE B.1: The sensors are needed for the car to identify doors, objects, and corners.

The side sensors will furthermore be used for feedback control to have the car drive alongside the walls.

B.1 Requirements

Distance: The sensor have to be able measure distances up to at least 50 cm. This will give the car time to respond to what the sensors measures.

Weight The distance sensors has to be light weight as with the rest of the components of the car.

B.2 Possibilities

SHARP GP2D12. This is an IR sensor. It has an internal distance measuring IC and a connector with supply, ground, and signal pins. It is easy mountable with two screws. Its output is an analogue signal of $0.6-4.7\ V$ which corresponds to the distance range $10-80\ cm$.[GP2D12.pdf] The weight of the sensor is 3.6 g.

Other types of sensors has been looked into, but as the GP2D12 should live up to the requirements and it is available, it was chosen to be used.

B.3 Test of the GP2D12

The sensor type was already estimated usable for this project, so what needed testing is the exact relationship between the distance and the output voltage.

The car four sensors mounted and each of these were tested individually while mounted. The sensors used the power supply from the Robostix, as they were already mounted on the car.

To read the output voltage a multimeter were used:

	Manufacturer	Model	Lab.No.
Multimeter	UNIGOT	A43	08097

B.3.1 Test Set-up

The GP2D12 was connected to the Robostix and the multimeter to the output and ground. the car was placed on the floor with the sensor being tested 10 *cm* from the wall. The car was the moved five centimeters at the time to a distance of 80 *cm*.

This was repeated twice for each sensor to ensure the reliability of the test.

The expected result was that the output would decrease the further away the sensor got from the wall and that the difference in output voltage would be significantly enough, depending on distance, to have a resolution of at least five centimeters.

B.3.2 Results

B.3.3 Discussion

The minimum measuring distance for all four sensors is 10 *cm*, as Figure B.2 implies. The two front sensors has a difference in output voltage at distances up to 70 *cm*, as seen from the two top graphs in Figure B.2. The two side sensors have a lesser range of up to 60 *cm*. This difference might be caused by interruptions from the wheels, which are place less than a centimeter from the side sensors. None that the sensors live up to the 80 *cm* stated in the data sheet, which can be the cause of other light conditions, and a different surface of the object, to which the distance is measured.

For this project the most used sensor ranges will be about $10 - 50$ *cm*. Therefore does the lack of $10 - 20$ *cm* maximum measuring distance not interfere with the car performing its tasks depending on the sensors.

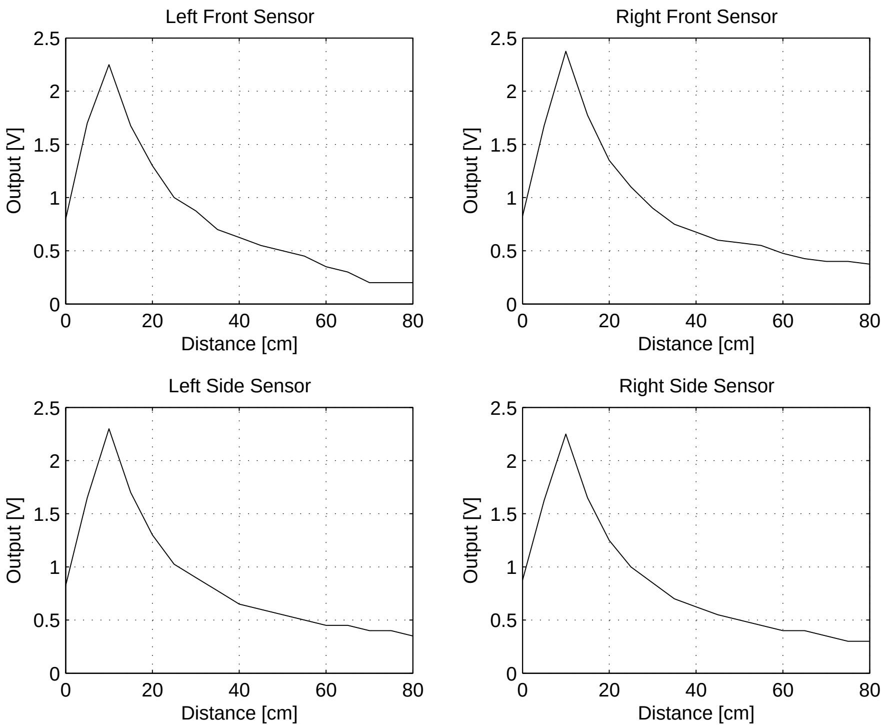


FIGURE B.2: The graphs shows the average of the two test conducted for each sensor.

B.3.4 Conclusion

The test of the GP2D12 showed that they all live up to the requirement of having a maximum range of more than 50 cm, as it has a measuring range of $10 - 60$ cm. The weight is 3.6 g per sensor, which is deemed an acceptable low weight.

Appendix C

Motors

Contents

The electric propulsion of the car is chosen to be provided by servo motors, and in particular μ-servos due to the light weight and small size. The μ-servo candidates were found on the internet, from a danish store the university have used before.

C.1 Introduction

Weight is an issue in the design of the car, which means that the power-to-weight ratio considerations are of high importance. The car has to sustain a fall and, as such, it is deemed necessary to minimize the mechanics in the design of the car. As a result of this, the car will have an engine for each wheel to avoid the use of gears and drive belts, which could result in mechanical failure as a consequence of the car hitting the floor at the end of its flight. Choosing to use four motors has the disadvantage of making the car heavier. The motor set-up is illustrated in figure C.1.

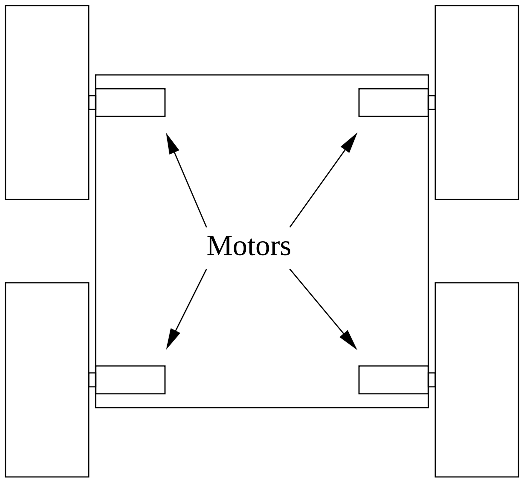


FIGURE C.1: The four motor set-up is chosen to lessen the use of mechanical components.

Two factors is considered in the choice of engine: Weight, and speed.

Weight, m: Addressing the weight factor, no maximum weight limit is specified. Instead, a specific type of motor is chosen. μ-servos have the advantage of low weight and enough torque to drive low weight vehicles and are even usable for small RC helicopters. Most μ-servos weigh less than 10 g.

Speed, v: Time is a factor, and as such is the speed of the car a factor as well. The speed of μ-servos is typically $58 - 111$ rpm.

C.2 Possibilities

Of the μ-servos studied, two were considered to be best suitable for the car. One, which has the lowest weight and one, which has the highest speed. In Table C.1 the technical data for the two μ-servos is specified.

Data	Contraction	Hitec HS-50	Blue Bird BMS-303	Units
Weight	m	5	3.4	[g]
Speed	v	111	90.9	[rpm]
Torque	τ	0.0586	0.0777	[Nm]

TABLE C.1: Best suitable engine choices for the car. The lightest and the fastest of the μ-servos studied.

The Hitec HS-50 has been chosen, as it is faster, and because four engines are used the torque is assumed to be high enough to drag the car.

C.3 Test

The test is done by giving the μ-servo a PWM signal, and measure the rpm of the wheel. To measure the rpm a digital tachometer is used, which detects reflecting light. To reflect the light on the wheel two reflectors are put on the wheel opposed to each other to make the measurement more reliable. The test setup is shown in Figure C.2, and the test equipment is listed in the following Table.

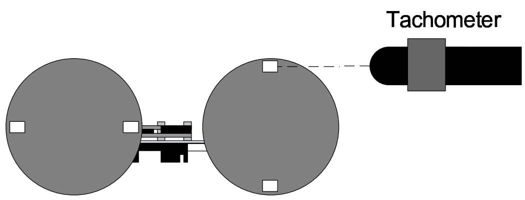


FIGURE C.2: Test setup for calibrating the pulse width with a speed of 100 rpm.

	Manufacturer	Model	Lab.No.
Tachometer	Shimpo	DT-205	40158703

One μ-servo was tested at a rpm of 100 without load, and the pulse width of the signal required to reach this were found. The frequency used is 50 Hz which is common for servo's. The results are shown in Table C.2.

μ-servo nr.	RPM	Clockwise [μs]	Counter Clockwise [μs]
1	100	2745	370

TABLE C.2: Results from PWM test of one μ-servo, with a frequency of 50 Hz.

When testing another of the four μ-servos it did not give the same rpm, which indicate that the reference point for the μ-servos are placed at different places. The PWM signal were found for each μ-servo to reach exactly 100 rpm. The results can be seen in Table C.3.

μ-servo nr.	RPM	Clockwise [μs]	Counter Clockwise [μs]
1	100	2745	370
2	100	2775	350
3	100	2700	350
4	100	2780	350

TABLE C.3: The results from PWM test of all four μ-servos, with a frequency of 50 Hz.

The 100 rpm is chosen, because it is still faster then the alternate μ-servo type, and gives a velocity of the car on 1.68 km/h, which is sufficient to map a room of the given size. Furthermore it gives the possibility to calibrate the velocity of one side of the car up and down, if for some reason one side does not drive as fast as the other one, and thereby still obtain a velocity of the car on 1.68 km/h which corresponds to 47 cm/s. The test shows that this speed can be uphold with no load applied, but when the load of the cars weight is applied the car drives with 49 rpm, which corresponds to 23 cm/s. This is due to the low torque of the μ-servo.

C.4 Discussion

The test showed that the HS-50 μ-servos that have been chosen have different reference points, and therefore needed to be calibrated separately. It also showed that the μ-servos can hold a rpm of 100 as wanted, and thereby keep a velocity of 1.68 km/h of the car with the chosen wheels, but the test were conducted without load. The test with the load of the car showed that the velocity of the car dropped to 23 cm/s, which is a sufficient decrease and is due to the low torque.

C.5 Conclusion

The HS-50 μ-servos were chosen because it was the fastest, and the test showed that it works as wanted for the purpose of this project. With this μ-servo the car can drive with a velocity of 23 cm/h, which is assumed sufficient for now. Therefore the conclusion is that the HS-50 μ-servo is used in this project as driving engines.

Appendix D

The Camera

Contents

The on board camera was found browsing the Internet for spy cameras, as we assumed these has the functionality, and the light weight, we need. The camera deemed best, judging from the specifications given, is the Wireless Camera GP-811T. The receiver for this camera connects to the USB-port in a computer and the camera is online right after installing the software that comes with the camera on a cd.

To use the camera in an optimal way we had to make modifications to its power supply cord, as it had a connector almost the size of the camera itself. Also the view angles of the camera had to be found to define how far from the walls we need to be to film the faces of people in the room. Furthermore, a device was constructed to mount the camera on the car.

D.1 Transmission

Transmission Frequency: ISM-2.400 $\approx$ 2.483 GHz

Modulation Type: FM

Bandwidth: 18 MHz

Undisturbed Transmission Range: 100 m

D.2 Supply Cord

The camera uses an eight voltage supply which is assured by a transformer, connected to the mains. To use it on the car, the supply cord was cut and connected to the car batteries instead. The connection is shown in Figure D.1.

D.3 View Angles

The desired camera coverage is to be able to see approximately two meters up at a distance of two meters from the wall. This means the camera will need to have a horizontal view angle of $\sim 45°$. For the chosen camera the horizontal view angle is $\sim 44°$, which, by itself, is acceptable. And when taking into consideration that the camera is elevated 5 or 5.7 cm depending on which side of the car face upwards, the actual view height is approximately 1.98 m at a two meters distance to the wall. The vertical and horizontal view angles is show in Figures D.2 and D.3 respectively.

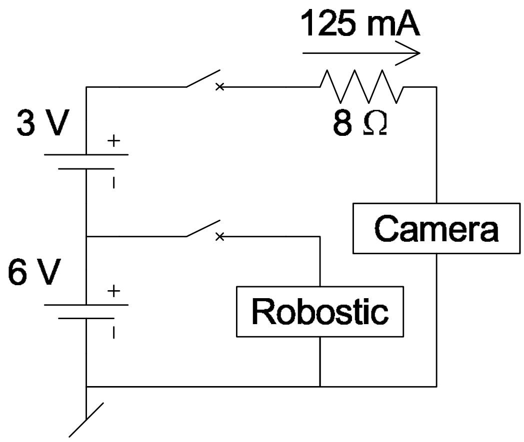


FIGURE D.1: The camera circuit.

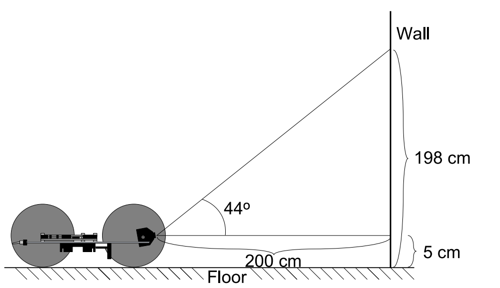


FIGURE D.2: The vertical view angle of the camera.

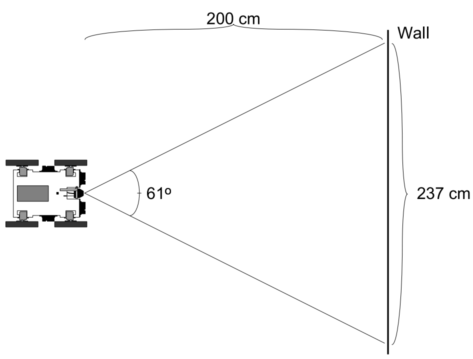


FIGURE D.3: The horizontal view angle of the camera.

D.4 Mounting

To film the room the car has a mounted camera pointing forwards. For full use of the camera's scope, a tilting mechanism is designed to have the camera film as little of the floor as possible, measured at its operating distance. The tilt mechanism is shown in Figure D.4 and its mount on the car is illustrated in Figure D.5.

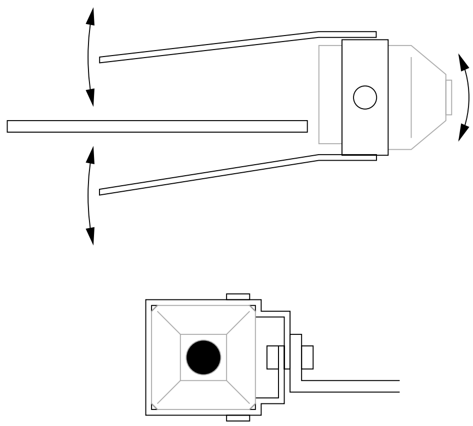


Figure D.4: The tilt mechanism.

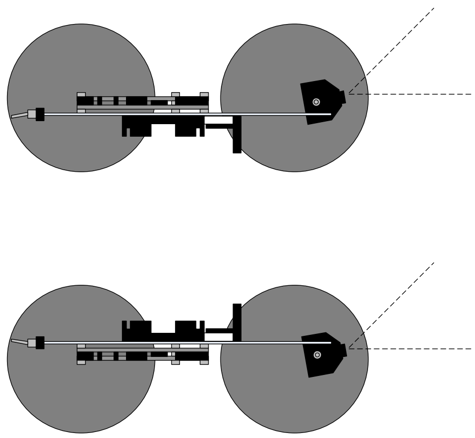


Figure D.5: The mounted tilt mechanism.

Appendix E

Software Design

This is a worksheet directly copied from the old/previous report. The only section not included is Mapping Consideration as these are included in the new report in an updated version.

Structured Software Development briefly:
Structured Program Development (SPD) is a method developed for software development and is originally directed at software developers and project leaders. In this project it will be used for full system analysis, as the method leads to a testable way of making a specification of requirements.

The analysis takes the perspective of the car, as there is no influence from any user, once the car is activated. That is a view on how the car will interact with its surroundings and how it needs to communicate with the helicopter.

The method used to analyse is based on the SPD method to find the functionality requirements.

E.1 General Description

E.1.1 System Description

The system is the autonomous car. An overview of the system is given in the deployment diagram in Figure E.1. With this in mind, the components of the system will be described.

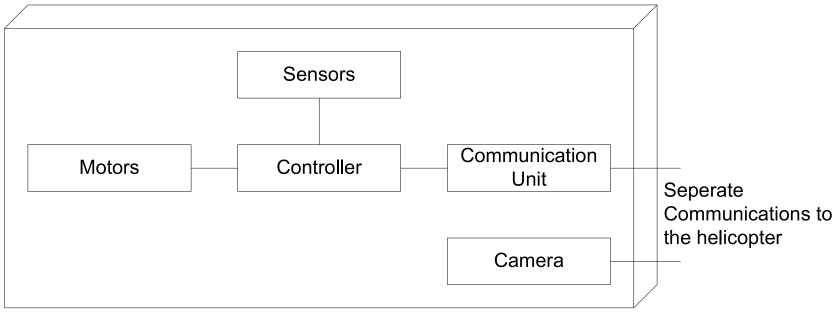


FIGURE E.1: Deployment diagram of the system.

Controller

This component controls the timing of the inputs and outputs, regulates the speed of the motors, calculates the path, and has a signal send to the helicopter, when the mapping is done.
The only component the controller does not interact with is the camera, as it transmits its pictures independent of the rest of the system.

Motors

The motors of the car are used solely for the car propulsion. That is, there are no motor controlled movable parts mounted on the car except for the wheels.

Sensors

There are four obstacles the car can encounter. To detect these, the car has four sensors mounted; two sensors pointing forward and one pointing to either side.

Furthermore there is a need for the car to know whether it has the bottom of it facing the floor, as it can flip over during the flight through the window. Due to this a sensor is placed facing downwards.

Communication Unit

The car control is activated from the beginning of the IARC mission. The only communication necessary from the helicopter to the car is a start signal. The communication needed for the car is to send a signal when it has completed mapping.

Camera

The camera is working independently of the rest of the system. It is activated, along with the rest of the system, at the beginning of the IARC mission. The only communication the camera performs is continuously transmitting pictures to the helicopter.

E.1.2 Functionality

The car has some main functionalities, like a motor control to handle the communication to the motors. The main functionalities are shown in Figure E.2. Also in this figure the directions of communication between them are outlined.

The car has to drive in a straight line forward, following the walls. To do this the measurements from the side sensors are used. The car will have to take measurements from the side sensors with a suitable frequency, which is chosen to be every five centimeters, as it assumable leaves enough time for the motor driver functionality to compute the response to changes.

The Motor Control should take care of all the basic functionalities, like drive forwards, drive backwards, rotate, and stop. While the Central Control should take care of the more advanced issues, like where to go next, when encountering an object. The Central Control has furthermore been divided into two extra functionalities, the Object Identifier and the Communication Unit, to point out the importance of these.

The Object Identifier is used when an object is detected, for instance a wall or a piece of furniture, to identify what sort of object it is to help the central control determine how to react.

The car acts on its own, without influence from the helicopter or other

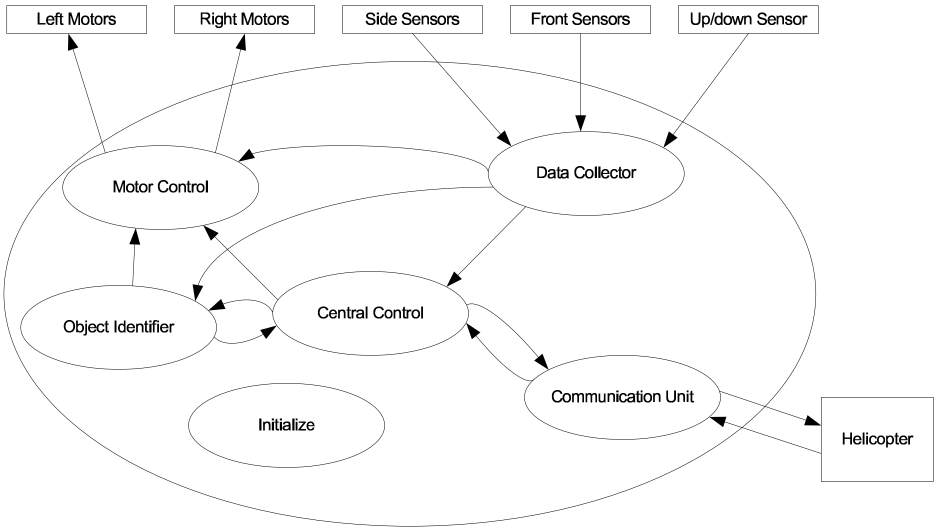


FIGURE E.2: The main functionalities and communications internally.

external device, during the mapping process. This means, the only data-communication taking place is a transmission from the helicopter to the car when the mapping should start and a transmission from the car to the helicopter when the mapping is complete. This functionality is provided by the Communication Unit.

E.1.3 Limitations

The car is developed in order to handle:

- Rooms with 90°.

- Corners and no holes in the floor, e.g. stairs down.

- Doors that are 82.5 *cm* wide, which is one of the commonly used door width i Denmark [SWEDOOR-KILSGAARD, 2007].

E.2 Specific Requirements

The first part of this chapter is based on finding the requirements for the functionalities in Figure E.2. The second part is illustrating these. This is made by use of timing diagrams.

E.2.1 Functionally Requirements

Motor Control. This functionality controls the four motors used for propulsion. The communications of the motor control functionality is illustrated in Figure E.3. An option is to implement some cognitive capabilities, and thereby make it able to drive adapting the motors speeds.

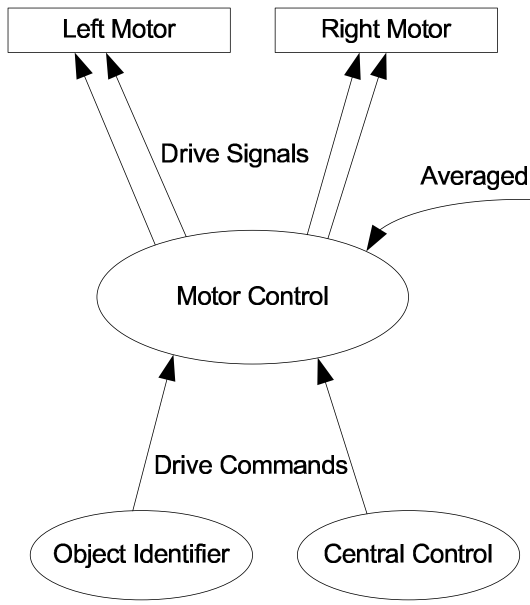


FIGURE E.3: The Motor Control uses feedback from the side sensors to drive straight. The drive command from the Central Control overwrites the feedback control.

Input: Three inputs are given to this functionality, an average of the side sensor measurements and drive commands, which are given from both the Central Control and the Object Identifier.

Side sensor averages: The input from the data collector is read every 0.1 s, and has values ranging from $10 - 60$.

Drive commands: These have a number from $0 - 5$ to represent the action required.

Function: With the motors chosen, the motor control's update rate is 10 Hz, which corresponds to the 5 cm side sensor update rate stated in E.1.2. The input from the Data Collector will be used as feedback to the update algorithm for the front left motor. This feedback shall ensure the car's ability to drive alongside the walls. The feedback will only directly act on one motor to simplify the control algorithm. The way this should be done is by having the back left motor dependent on the front left motor and have the two right motors driving at constant speeds. The feedback control shall not be in effect, when a new drive command is given by the central control, and should in general only be active, when the car is driving either forwards or backwards.

Output: There are four outputs from this functionality, one for each motor.

Front left motor: This is the only motor directly affected by the feedback from the Data Collector.

Front right motor: This has three speeds, 100, 0, and -100 rpm.

Back left motor: The update algorithm for this motor shall be dependent on the updated algorithm of the front left motor, which means that, indirectly, this one is also affected by the feedback control.

Back right motor: As with the front right motor this one has three speeds, 100, 0, and -100 rpm.

Data Collection. The purpose of this functionality is to collect sensor measurement data, verify them, and make them usable for the motor control and central control functionalities. A diagram of this functionality is shown in Figure E.4. An option is to have this functionality use fault detection or sensor information fusion.

Input: This functionality has five inputs, from two side sensors, two front sensors, and a sensor pointing downwards. The distance range and digital resolution for the side and front sensors are the same, and will henceforth be referred to as the distance sensors.

Distance range: $10 - 60$ cm

Digital resolution: 6 bit (1 cm resolution)

The sensor mounted at the bottom of the car is used to detect the floor. As such, its distance range is indicating the low and high output range used.

Distance range: Low: $0.5 - 1.5$ cm, high: $1.5+$ cm

Digital resolution: 1 bit (low:0, high:1)

Function: For each of the distance sensors the moving average will be computed, and for this two ring buffers are used, one with the real data and one with the average. The sensor at the bottom of the

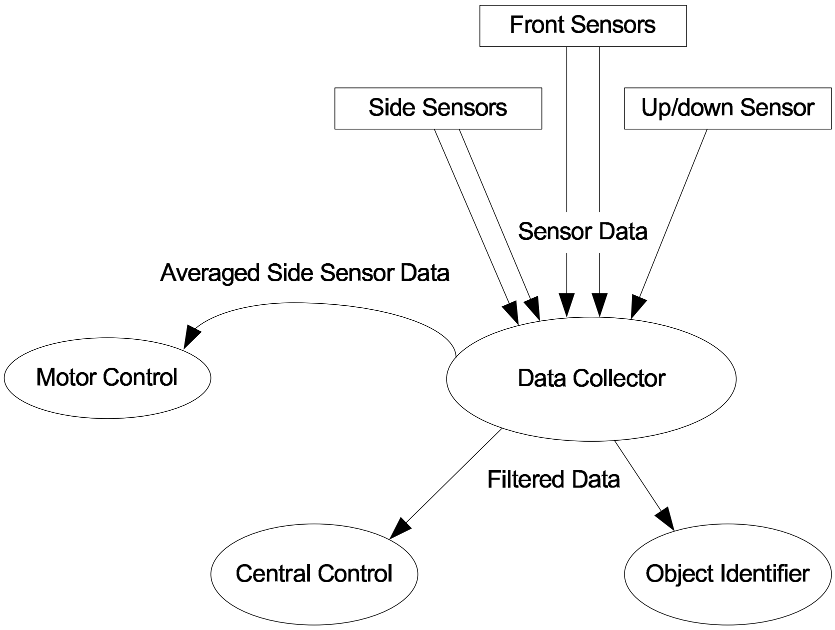


FIGURE E.4: The Data Collector functionality taking inputs from the sensors and filtering them for the motor control and central control.

car will only be used once in the mapping process, and the data used will be the average of ten measurements.

Output: This functionality has three outputs, one to the motor control, one to the central control, and one to the Object Identifier.

Motor control: The averaged data from the side sensors are made available for the motor control. The motor control runs with 10 Hz and will therefore have a sensor update every 0.1 s.

Central control and Object Identifier: All averaged data are send stored and thereby the central control can use it. The ring buffer with the average sensor data will be in a block of memory accessible by the central control.

Central Control. This is the functionality, which controls the mapping process. It estimates and counts the number of corners, estimates how much of the room has been mapped, determines what actions to take when encountering an object in the drive path, and makes sure a signal is send when the mapping is done. This functionality is illustrated in the diagram in Figure E.5.

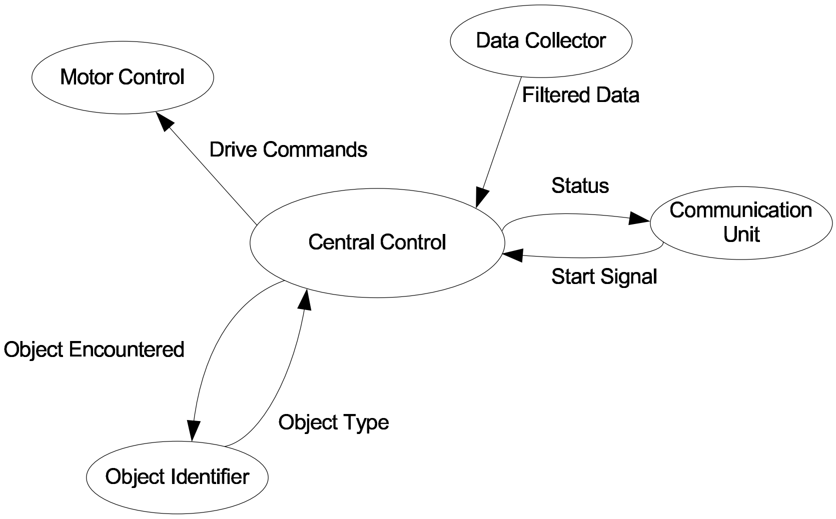


FIGURE E.5: Diagram of the system's Central Control.

Input: The Central Control has three inputs, Filtered Data, Start Signal, and Object Type.

Filtered Data: This is the filtered data from all the sensors. Values from the distance sensors are ranging from 10 to 60 and the data from the bottom sensor have a range of 0 to 1.

Start Signal: The signal from the helicopter to the car to commence mapping.

Object Type: This shall indicate what sort of object the car has encountered. Depending on how many types of object that can be encountered this should be ranging from 0-9.

Function: The Central Control takes command of the car when an abnormality is encountered. That is, when the front sensors detect an object closer than 30 cm, or on the right side sensor measurement has an increment of more than 10 cm.

In both cases the Central Control will send a stop command to the Motor Control and call upon the Object Identifier to identify the type of abnormality. The Object Identifier can give one of four replies, wall, object, door, or no door.

Wall: If the car has encountered a wall it is detected by the front sensors. The right side sensor is then read, and if it measures less than 60 cm it is estimated to be a corner, and the Central Control calculates a command sequence for the Motor Control. The process of dealing with a closed corner is shown in

Figure E.6.

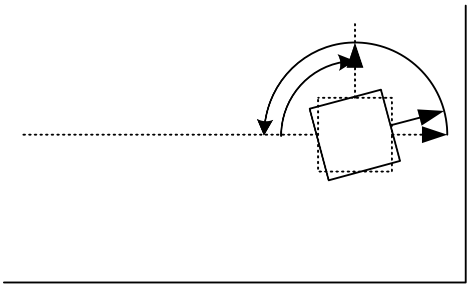


FIGURE E.6: When the car encounters a closed corner it makes a sweep to film inwards in the room.

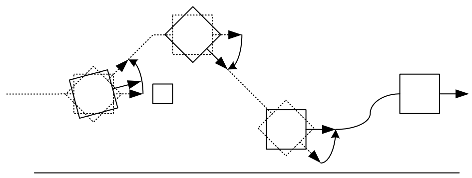


FIGURE E.7: When encountering an object, the car drives left around it.

Object: In this case the Central Control finds a command sequence taking the car to the left side of the object, and when the object is no longer detected by the right side sensor, the Motor Control's feedback control takes over again. An illustration of how the car passes an object is illustrated in Figure E.7.

No door: This reply means there is an open corner on the car's right and the Central Control sends the command to the Motor Control.

Door: If a door is detected the car is in the doorway, and the Central Control makes the car rotate left until the front sensors are pointing at the door frame. Then Central Control gives the command to Object Identifier, as if an object is in the drive path.

When the mapping is done the Central Control passes a done signal to the Communication Unit.

Output: The Central Control has three outputs, drive commands to the Motor Control, status to the Communications Unit, and object encountered to the Object Identifier.

Drive Commands: These range from 0-5 and contains the basic drive functionalities, stop, forwards, backwards, rotate left, and rotate right.

Status: This contains an error and a finished mapping signal.

Object Encountered: With this the Object Identifier functionality is called.

Object Identifier. This functionality is called every time an abnormality is encountered by the car. There are two possible abnormalities considered, when the front sensor measurements decreases to less than 20 *cm*, and when the side sensor measurements show an increase of more than 10 *cm*. A diagram of the Object Identifier's interactions with other functionalities is shown in Figure E.8.

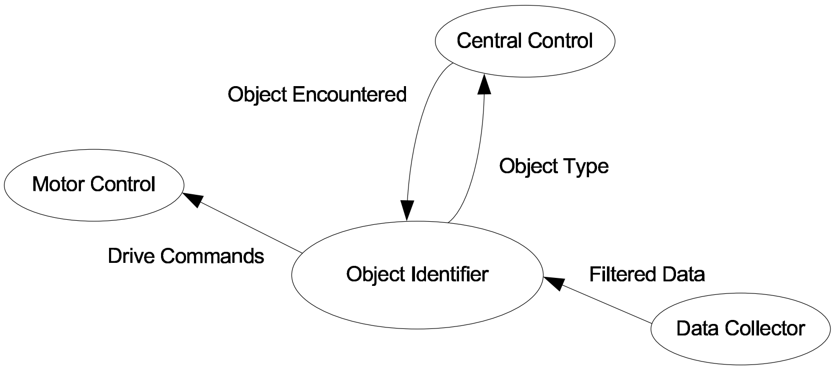


FIGURE E.8: When the Object Identifier functionality is called, it can give drive commands to the Motor Control.

Input: The inputs for this functionality are object encountered from the Central Control and the filtered data from the Data Collector.

Object Encountered: The input can contain one of two values, 1 or 2.

Filtered Data: This is the filtered data from all the sensors. Values from the distance sensors are ranging from 10 to 60 and the data from the bottom sensor have a range of 0 to 1.

Function: If the input is 1, the car has met either a wall or an object in front of it. To determine which, the Object Identifier sends a rotate left command to the Motor Control and reads the change in the front distance measurements. If, during the rotation, one of the front sensors shows significantly more than the other, and more than the initial distance, the car has encountered an object. Otherwise it is a wall. In both cases the information is sent to the Central Control.

If the input is 2, the car has either a door or an open corner on the car's right. To determine which, the Object Identifier should find a command sequence that brings the car drive around the open corner, or through the door way, and at the same time poll on the left side sensor. If there is a reaction on the left side sensor a door is identified. If there is no reaction on the left side sensor within a certain time, a open corner is detested. In both cases the information is sent to the Central Control. The door encounter is shown in Figure E.10.

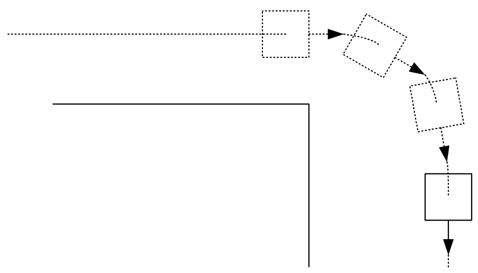


FIGURE E.9: The car encountering an open corner.

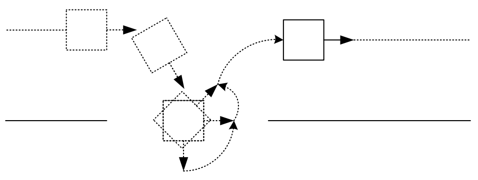


FIGURE E.10: When encountering a door, the car rotates left until the front sensors are pointing on the door frame, and runs the Object Identifier.

Output: The Object Identifier has two outputs.

Object Type: This is a number, send to the Central Control, in the range 0-9 send to the Central Control, which indicates which type of abnormality the car has encountered.

Drive Command: As with the drive command from the Central Control to the Motor Driver. It ranges from 0-5 and has a basic drive functionality associated with each of the numbers.

Communication Unit. The purpose of this functionality is to receive the start signal from the helicopter and send the done mapping signal to the helicopter when done.

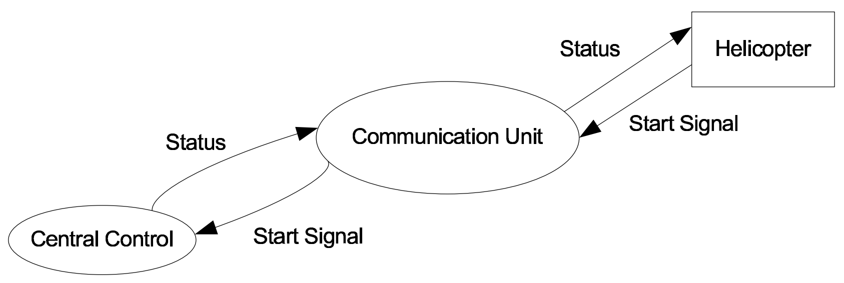


FIGURE E.11: The Communication Unit controls the transmissions and data reception to and from the helicopter.

Input: There are two inputs, one from the helicopter and one from the Central Control functionality.

Status: This is the done signal from the Central Control.

Start: This signal is send from the helicopter to the car.

Function: The WLAN communication gets activated when the system starts up. Upon receiving the start signal from the helicopter, the Communication Unit have to shut down the WLAN to avoid interruption from elements outside the system. When the car is done mapping the Communication Unit shall activate the WLAN again and start transmitting the done mapping signal.

Outputs: The outputs are a done signal to the helicopter, transmitted via WLAN, and a start signal to the Central Control.

Initialize. This functionality shall not interact with the rest of the functionalities and, as such, does not have any inputs or outputs. As the name implies it is the start up functionality.

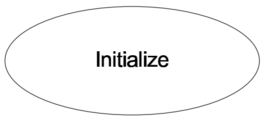


FIGURE E.12: Initialize is responsible for the system start up.

Function: To start up the system, activate the WLAN, the motor driver and the sensors.

E.2.2 Timing Diagrams

Initialize: When the system is first activated the Initialize functionality starts up the Motor Control and the Central Control. The Motor Control starts looping and is initially set to stop. The Central Control calls the Communication Unit, which starts pending for a start signal to pass to the Central Control. The initialization process is shown in the timing diagram in figure E.13.

Closed Corner Handling: The car drives alongside a wall using feedback control. Measurements from the front sensors of less than 30 cm, passed from the Data Collector to the Central Control, means that the car has encountered either a wall or an object. The Central Control will take over, send a stop signal to the Motor Control, and call upon the

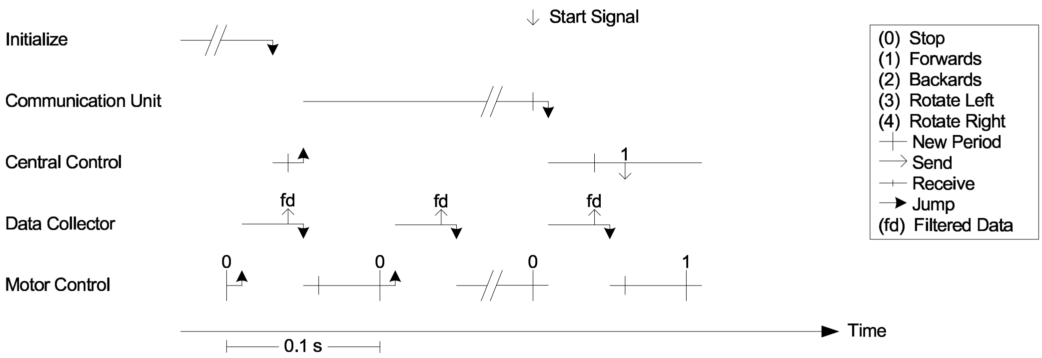


FIGURE E.13: Timing diagram for the initialization process.

Object Identifier to determine which kind of abnormality the car has encountered.

The Object Identifier sends a rotate left signal to the Motor Control. The information, that it is a wall, is passed to the Central control and computes a command sequence, which is send with 1 Hz to the Motor Control. The given command sequence makes the car rotate left to point in its former driving direction and 90° back, pointing in a new direction along the new wall encountered. When done the control is once again given to the Motor Control and the feedback control is resumed. This routine is illustrated in Figure E.14.

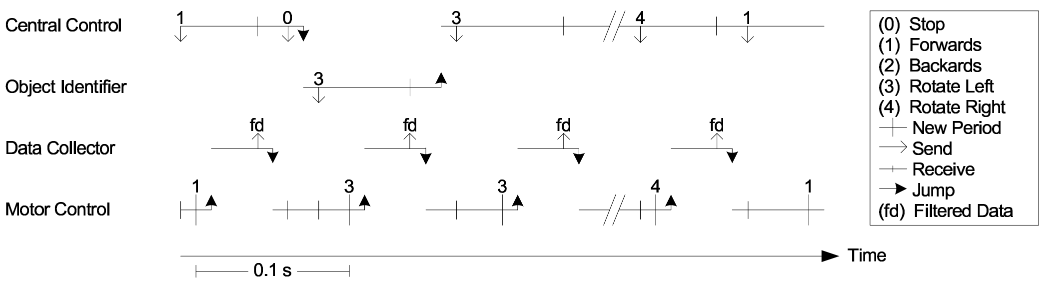


FIGURE E.14: The timing diagram for handling a closed corner.

Object Handling: As with the closed corner handling, an abnormality is encountered in front of the car and, the feedback control in the Motor Control stops, the Central Control takes over, and the control is passed further on to the Object Identifier. The Object Identifier sends a rotate left signal to the Motor Control. The information, that it is an object, is passed to the Central Control, which then computes a command sequence to take the car to the left side of the object. Once on the side of the object the control is given back to the Motor Control and the feedback control is resumed. The timing diagram for this routine is shown in Figure E.15.

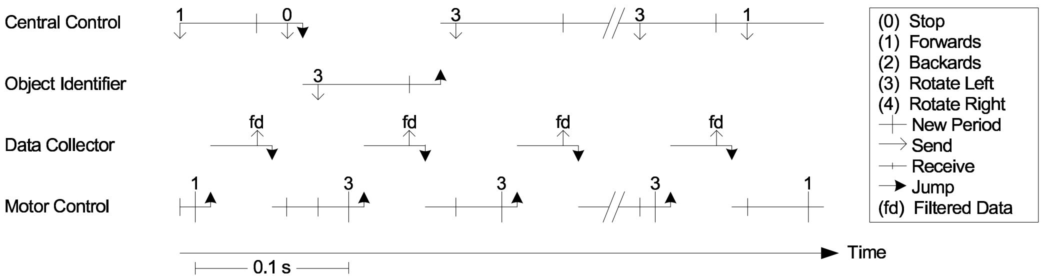


FIGURE E.15: The timing diagram for handling an object blocking the drive path.

Open Corner and Door Handling: The car drives alongside a wall and the data from the side sensors sent to the Central Control indicates a more than 10 *cm* increment. The Central Control sends a drive command to the Motor Controller and notates that a corner is encountered. Furthermore data from the left side sensor is read, to distinguish an open corner from a door. The timing diagram for when encountering a door or an open corner is much alike, as with the object and closed corner timing diagrams, and as such, only one timing diagram covering both are made. This is illustrated in Figure E.16.

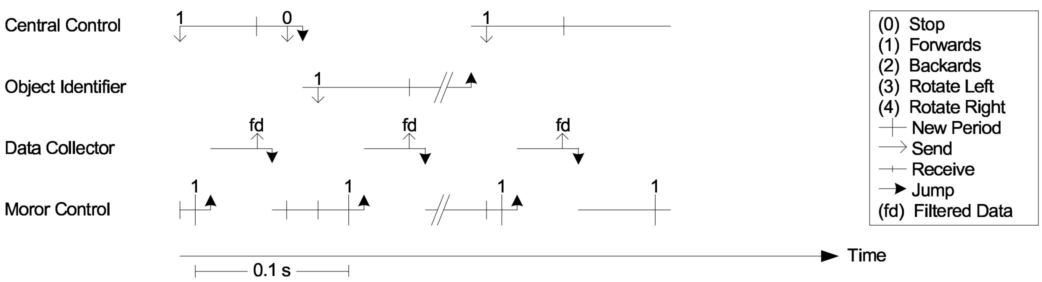


FIGURE E.16: Timing diagram for encountering an open corner or a door.

Mapping Done: When the last corner is reached, a signal needs to be sent. The Central Control sends a stop command to the Motor Control and calls the Communication Unit and it sends the done signal. After that is done the control is given back to the Central Control. This routine is illustrated in Figure E.17.

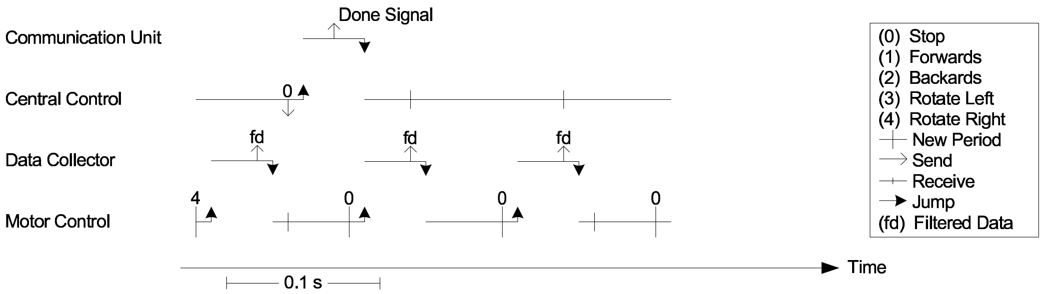


FIGURE E.17: Timing diagram for when the car is done mapping.

E.3 Accepttest Specification

The main goal of the project is to have the car make a video map of an arbitrary room. This task is divided into smaller tasks. Each of these tasks is described from a test point of view, which means that each task will have to be testable. The tasks can furthermore be divided into two groups

- the tasks concerning the car variables, for instance start position,
- and the tasks concerning the room variables, for instance room shape.

When both these groups are tested, a test including both is performed.
In all tests the car is expected to map the room and stop when it is done.

E.3.1 Test Setup: Car Variables

The test setup for tasks concerning the car variables is shown in Figure E.18. In this line of tests the room has constant dimension, 4 $m \times 6$ m. There are no objects in the room and there are only closed doors.
The car variables are:

Start position: Where in the room does the car land.

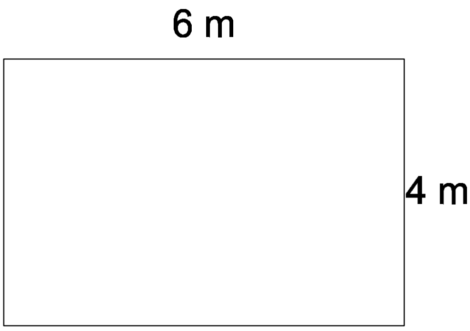


FIGURE E.18: Room test setup.

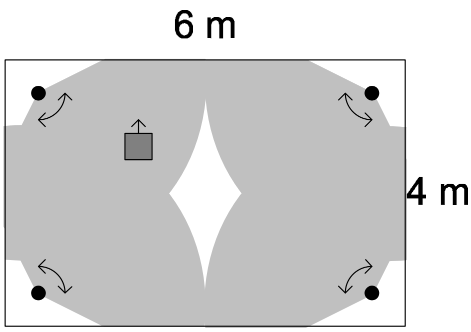


FIGURE E.19: Camera coverage from the corners in test one.

Start direction: In which direction does the camera point.

Up/down: Which side of the car is up.

Test 1: All Fixed

This is a test of whether the car can map the room starting from the position and with the direction shown in Figure E.19. The top of the car is facing upwards. The expected result is to have a video map of the entire room, and what is not covered by the rotational movement in the corners, is covered when driving between the corners.

Test 2: Random Start Position

Different start positions are randomly generated to test whether the car car start at a random start position. These positions are shown in Figure E.20. The start direction is the same for all three positions. and the top of the car is facing upwards.

Test 3: Random Start Direction

Here the random start direction is tested by randomly generating different directions, as shown in Figure E.21. The start position is fixed. The top of the car is facing upwards.

Test 4: Random Side Up

The bottom of the car is facing upwards, which means the motors will have to run backwards to move the car forwards. The start position and direction

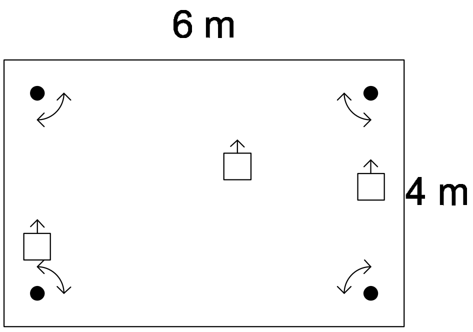


FIGURE E.20: Test setup for test 2. Three different start positions.

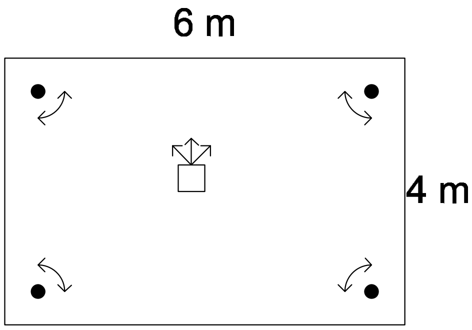


FIGURE E.21: Test setup for test 3. Three different start directions.

is predetermined to be the same as in test 1. Figure E.22 shows how the car turns.

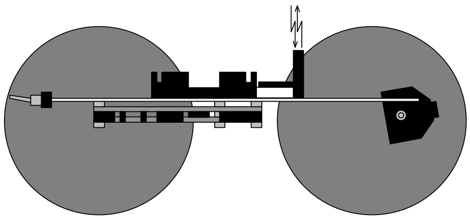


FIGURE E.22: Test setup for test 4. Turning the car upside-down.

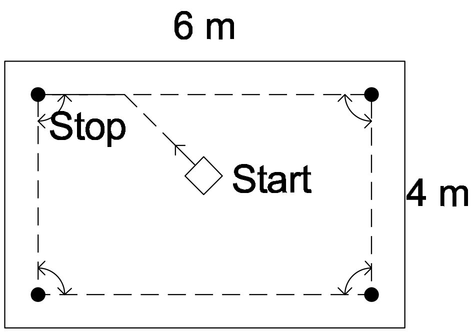


FIGURE E.23: Test of whether the three previous tasks can be performed simultaneously.

Test 5: All Random

To round up the first line of test, a test is made where start position and direction is random and which side of the car is upwards is random. The specific test setup is illustrated in Figure E.23.

E.3.2 Test Setup: Room Variables

For this line of tests the car starts in the same position as in test one, with the same direction and the same side up.

The room variables are:

Room shape: The number of corners and length of the walls.

Doors: Doors placed where the car should detect a wall.

Objects: The objects placed on the floor, specifically in the drive path of the car.

Test 6: Random Shape

This is a test of the car's ability to map regardless of the number and kind of corners and regardless of the lengths of the walls. The test setup is a wall where the car will first encounter a closed corner, then an open corner and lastly another closed corner.

Test 7: Door

An open door is introduced in this test. The door is placed in a wall and will appear to the simulated car after one closed corner encounter.

Test 8: Object in Drive Path

An object is placed in the drive path of the car. The object's dimensions are: Width= 10 mm, depth= 10 mm, and height= 200 mm. The specific test setup is a ten meter wall with 3 objects placed in the drive path.

Test 9: All Random

To round up this second line of tests, the mapping process is performed with the length of the walls varying from 4 m to 6 m, one open door, two objects of the same size as in test 8 and 9, one placed in the drive path and one in a grid point. The specific test setup is illustrated in Figure E.24.

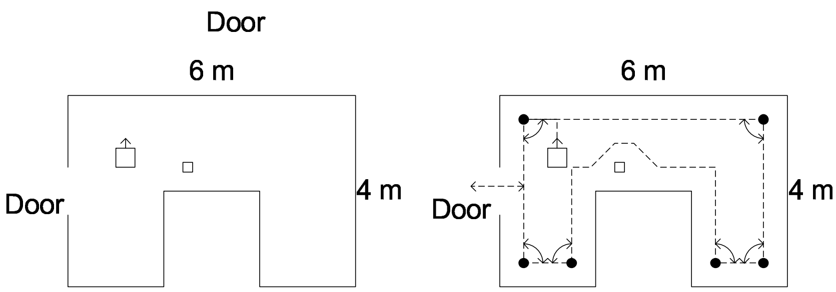


FIGURE E.24: A test of all the room variables combined.

Bibliography

[Danes Grow Taller and Taller, 2003] Mette Kjøller and Michael Davidsen. *Danes Grow Taller and Taller*, 2003.
URL: `http://www.si-folkesundhed.dk/Ugens\%20tal\%20for\%folkesundhed/Ugens\%%20tal/Danskernes\%20h\%C3\%B8jde.aspx`.
Downloaded 2007-07-08.

[The Danish Ministry of Science, Technology and Developement, 2006] Teknologiens fremsyn - om kognition og robotter (the vision for technology - on cognition and robots), April 2006. The Danish Ministry of Science, Technology and Developement.

[Franklin et al., 2002] Gene F. Franklin, J. David Powell and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, 2002, 4. udgave.
ISBN: 0-13-098041-2.

[GP2D12.pdf] Sharp. *GP2D12 datasheet*.

[Gumstix inc., 2007] Gumstix inc. *Gumstix specifications*, 2007.
URL: `http://gumstix.com/store/catalog/product_info.php?products_id=135`.
Downloaded 2007-06-06.

[MAX238.pdf] Maxim. *MAX232A datasheet*, 2004.

[OPB704-extra.pdf] Optek. *OPB704 datasheet*, 1996.

[Servey & Beichner, 2000] Raymond A. Servey and Robert J. Beichner. *Physics For Scientists and Engineers with Modern Physics*. Harcourt, 2000, 5th. udgave.

[SWEDOOR-KILSGAARD, 2007] SWEDOOR-KILSGAARD. *Måleskema*, 2007.
URL: `http://www.swedoor.dk/glatte_formpressede_maalskema.`

pdf.
Downloaded 2007-05-15.