Mälardalen University
School of Innovation, Design and Engineering
Västerås, Sweden

Project course in robotics and advanced embedded systems,
DVA473 and DVA474
Autumn and winter 2018

The UNICORN project

Examiner: Mikael Ekström
Mälardalen University, Västerås, Sweden

Supervisor: Mirgita Frasheri
Mälardalen University, Västerås, Sweden

Project manager: Ulrik Åkesson
Mälardalen University, Västerås, Sweden

Project participants: Sebastian Andersson
Sweta Chakraborty
Marielle Gallardo
Mujtaba Hasanzadeh
Alexandra Hengl
Mälardalen University, Västerås, Sweden

Research partners:
MÄLARDALENS HÖGSKOLA ESKILSTUNA VÄSTERÅS
CHALMERS
Husqvarna Group
VOLVO VOLVO GROUP
PWS Part of the ESE World
HIAB
Göteborgs Stad

Sponsors:
WE WÜRTH ELEKTRONIK
ROLLCO SPECIALIZED ON LINEAR MOTION

February 3, 2019

Abstract

The UNICORN project is a three-year project spanning 2017-2020. The vision is to streamline the refuse handling in housing areas by introducing autonomous systems and robots. The UNICORN project is a collaborative object with multiple partners where the area of work for Mälardalens University is to develop the autonomous robots. This report is the documentation of the work done during the second iteration of the project within the scope of the courses "Project course in robotics" (DVA473) and "Project in advanced embedded systems" (DVA474). During this iteration of the project a lift system has been constructed, a human-robot intention system has been implemented, and localisation with the help of ultra-wideband technology has been explored.

Acknowledgements

We would like to thank our supervisors Mikael Ekström and Mirgita Frasheri for their support, guidance, and faith in us students.

We would like to thank Joaquin Ballesteros for his support and guidance in ROS. Without him, the code would have made alot less sense.

We thank Würth Elektronik and ROLLCO for sponsoring this project with parts that made the project possible.

We would also like to thank the rest of the faculty of IDT for support and knowledge during the project.

Abbreviation

2-D Two Dimensional

AoA Angle of Arrival

AMCL Adaptive Monte Carlo Localization

DC Direct Current

HRP Husqvarna Research Platform

LiPo Lithium polymer

LOS Line-Of-Sight

MDH Mälardalens Högskola *Mälardalen University*

NLOS Non-Line-Of-Sight

PCB Printed Circuit Board

PDB Power Distribution Board

PLA Polylactic Acid

ROAR RObot based Autonomous Refuse handling

ROS Robot Operating System

RSS Received Signal Strength

RTSL Real-Time Location System

ToA Time of Arrival

ToF Time of Flight

TDoA Time Difference of Arrival

TWR Two-Way Ranging

UKF Unscented Kalman Filter

URDF Unified Robot Description Format

USB Universal Serial Bus

UWB Ultra Wideband

Table of contents

1 Introduction

Author: Ulrik Åkesson

Refuse handling has been a subject of great importance for as long as large populations of humans have lived together. After waste and filth were found to be the cause of many of the Cholera outbreaks in the 19th century, ways to properly remove and handle waste begun to be developed [1]. However, as populations continue to grow, the strain on the waste management system increases. In Sweden alone, each citizen produces 440 kg of refuse per year on average, according to the latest report from Naturvårdverket (Swedish Environmental protection Agency) [2]. Further improvement upon the existing waste management system is therefore important so that it can meet the needs of a population that continues to grow.

The UNICORN project is a collaboration between Mälardalens University, Chalmers University of Technology, Husqvarna Group, Volvo Group Truck Operations, PWS Nordic, HIAB, and Gothenburg Municipality. The vision of UNICORN is to streamline the handling of refuse in housing areas with the help of autonomous robots. The project started in 2017 with the goal to demonstrate the project at Apelsingatan in Gothenburg 2020.

All the research partners have taken on different tasks in the project. Chalmers is the coordinator of the project, managing risks, and deadlines, as well as being responsible for creating the control towers that will coordinate and schedule the different parts of the infrastructure. PWS is responsible for creating the bins that will be used, as well as a central underground bin where the robots will place the refuse. HIAB is in charge of constructing an autonomous crane that can be used on the trucks that empty the central refuse station. Volvo GTO and Husqvarna will assist with the development and with their extensive knowledge. MDH is in charge of creating the small autonomous robots that will work in the housing areas.

Creating a robot that is supposed to work in an unsupervised urban area presents multiple challenges, especially with regards to safety, security, communication, and navigation. When the robot is fully developed, it is expected to be able to plan a path, retrieve, empty, and return refuse bins, drive autonomously in an urban environment, and communicate its intention of movement.

1.1 Previous iterations

This iteration of the UNICORN project is a direct continuation of the work done on Charlie during the project course of 2017. Which in turn is a continuation and reboot of the ROAR project from 2016.

1.1.1 ROARy

In 2016, the robot ROARy was developed as a part of project ROAR which purpose was to show how technology and smart machines in the future could help with different everyday activities such as refuse collection. ROARy was constructed on the Husky robot platform and was designed to be deployed from the back of a refuse truck and retrieve refuse bins at the households, thus reducing the amount of heavy labour the truck's operator would have to do. ROAR was a collaboration between Mälardalens University, Chalmers University of technology, Penn State University, Volvo Group, and Renova.

1.1.2 UNICORN

Due to ROARy's size, it was slow, heavy, and noisy resulting in the re-imagination of the project in 2017 when the scope of the UNICORN project was set. As previously mentioned UNICORN is a three-year project and in this first year, the basis of the platform was developed. Charlie was built on a Husqvarna Automover 430X which have been unlocked as a Husqvarna Research Platform (HRP) and the Robot Operating System (ROS) was implemented as a middleware. The first iteration ended with Charlie in a state where it could receive orders, plan a global and local path, and with sensors for vision and range finding implemented. [3]

1.2 UNICORN continuation

The scope for this iteration of the project is to implement a robot which is able to navigate in an unknown indoor environment where it is expected to be able to retrieve, transport, and empty refuse bins without colliding with obstacles. Since the robot will work in close proximity to humans, their safety and well-being needs to be ensured. From these requirements, three different areas have been identified as the primary focus for this year's project.

Lift system
In order to handle refuse bins a lift system needs to be created. The lift system should be able to retrieve refuse bins mounted on the wall and weighing up to 15 kg.

Human robot interaction
Ways for the robot to interact with humans in a way that ensures safety and the well-being of humans will be researched and implemented.

Localisation with Ultra Wideband (UWB)
In order to improve Charlies navigation capabilities, the possibility to use UWB technology for localisation will be researched and tested.

The robot will be referred to as "Charlie" or "the robot" throughout the report.

1.3 Report structure

The report is structured as follows:. In sections 3 through 5 the three main areas which have been developed in this project are presented in depth including the results from any tests as well as any discussion related to that area. In sections 6 the final state of Charlie as of the end of the project is described including changes made to the global path planner from the previous iteration. In section 7 the suggestions for future work is listed. And finally, in section 8 the conclusions drawn after this project, are presented.

2 ROS

In this section the basics of ROS are briefly described. For a more extensive description visit the ROS Wiki [4].

Node
A node in ROS is basically a program, it can be written in C++ or Python. A node communicates with other nodes over topics that is can subscribe and/or publish messages to.

Topic
A topic in ROS is what the nodes communicates over, it can be seen as edges in a graph which connects the nodes.

Subscriber
A node that recieves messages from a topic is called a subscriber. It can be seen as a consumer that does something with the data it recives.

Publisher
A node that publishes messages to a topic is called a publisher. It can be seen as a producer that sends data that it creates.

Master
The server to which connects all the nodes.

Message
A data structure that is sent over topics.

3 Lift system

Author: Sebastian Andersson

One of the main purposes of the robot is not only to navigate to refuse bins but also to detach them from a wall-mount. To meet those purposes, a lift-system was developed after suggestions from the last iteration of the UNICORN project. That includes the thesis work by B. Lindgren and G. Kousmanen from the spring of 2018 [5]. Figure 1 demonstrate how the lifting system is mounted on the robot.

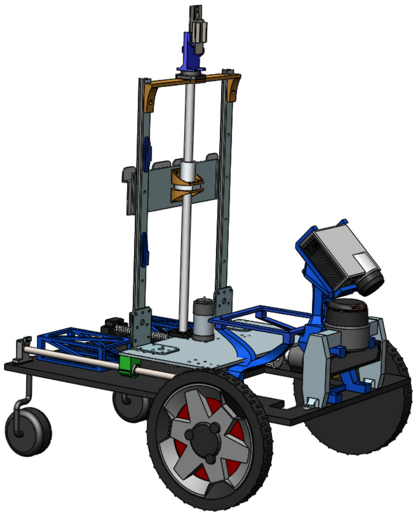


Figure 1: The figure demonstrate the entire model of Charlie where the lift system is mounted.

3.1 Selection

Several proposals on lifting system models were provided so that a decision could be made from many suggestions. The additional solutions are shown in Figure 2. All suggestions were summarised and discussed with a representative from Husqvarna. The chosen approach was considered most stable and was the approach that appeared not too complex to construct. A representation of that model can be seen in Figure 3

3.2 Linear motion

The lifting system is designed to provide two-dimensional linear motion. The ability to operate in the vertical and horizontal axis is achieved by inserting a ball-screw in each dimension. The vertical axis motion ensures that the bin is lifted from its mount. The horizontal axis motion is required to position the load as close to the robot's centre as possible. That would result in a more even weight distribution, which is done to try to avoid slippage. Positioning the bin closer to the driving wheels will ensure higher friction according to equation 1.

$$F_f = \mu * F_n \tag{1}$$

Where

F_f : The friction force (N)

μ : The friction coefficient of the surface

F_n : The normal force (N) perpendicular to the surface.

The equation says that the closer the centre of gravity is to the driving wheels, the higher normal force distribution will apply on those wheels, which in turn will induce higher friction.

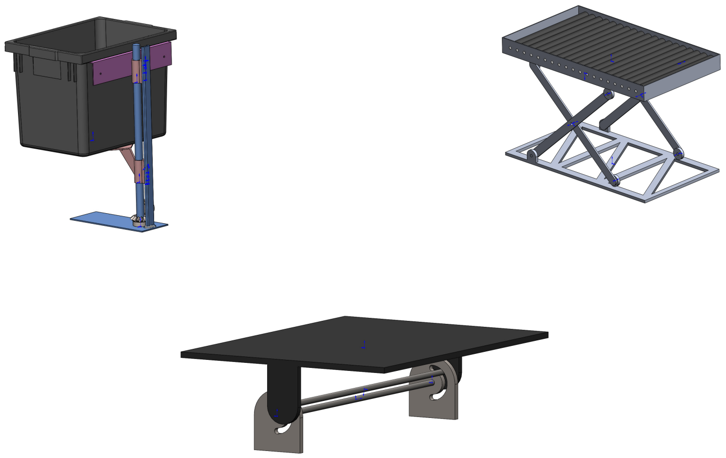


Figure 2: The figure represent additional proposed solutions to a lift system. The first is based on a system with a ball screw and two nuts that hold the bin in place. The second solution lift the bin by shortening the distance between the lower end points of the x-shaped structure, which forces the top base upward. The third solution take advantage of a specific shape of a slot, that forces the plate upwards.

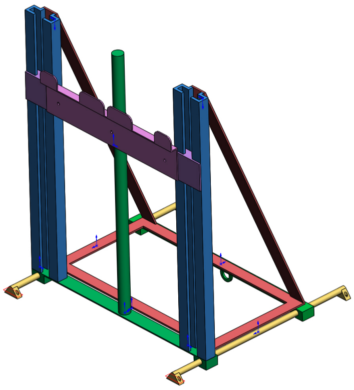


Figure 3: The prototype version of the lift system that was chosen for further development.

Furthermore, the ability to relocate the bin and position it on the surface of the robot will put the weight on the surface, and relief the ball-screw of the load. Thus, the bin would be more securely positioned, which is a vital parameter during transportation.

3.3 Hardware

This section will describe the aspects of hardware for Charlie. It is a combination of the created parts, the electronics, and the mechanics for the lift system.

3.3.1 Material

The material used for the milled parts is aluminium, with the DIN EN-AW 5754 alloy. A thickness of 3mm provided the most reasonable trade-off between weight and robustness. Aluminium possess various properties, however, the most vital ones is a low weight, its robustness, and that it is easy to work with. Thus, aluminium is a suitable choice of material for this application. Since most of the load from the refuse bin will be directed to the base plate of the lifting system, the thickness of that part was chosen to be 5mm

The angle irons that are mounted on the surface plate are used to hold the tower in place.

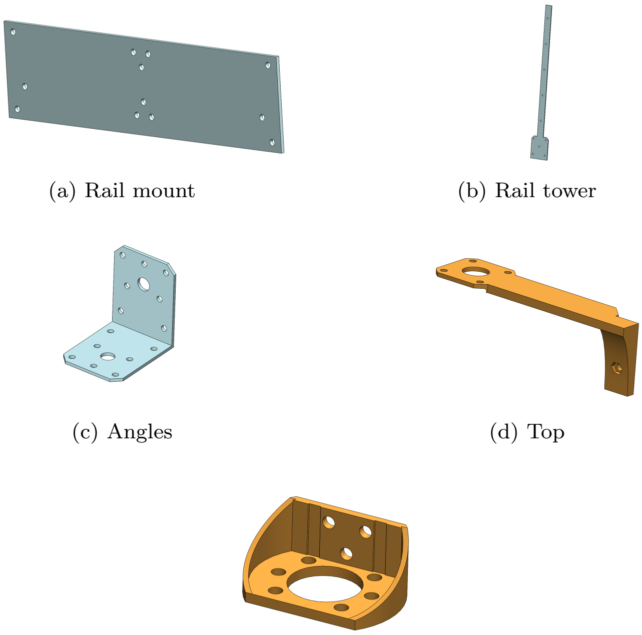
Therefore, they were bought from a hardware store to provide ample load capabilities.

In addition, complicated parts that were not appropriate for milling was instead printed in PLA plastics by a 3D-printer. By changing the amount of infill inside the parts, as well as the formation of the infill, the structure could be printed for robustness. The layer thickness could also be determined, to enable a more dense part. Another beneficial aspect of using the 3D-printer is the location of important features. They are positioned with high precision, which can not be achieved by hand. Thus, the manufacturer can be certain that a specific feature will be located exactly where intended.

3.3.2 Modelled parts

This section will describe the design and intention of constructed parts for Charlie. Firstly the parts needed for vertical motion will be presented, and later the horizontal motion parts.

The parts for the vertical axis lifting motion can be seen in Figure 4. These parts are for instance the Rail mount that act as a connection between the vertical c-rails and the bin mount, and the Clamp that connect the ballscrew and the Rail mount. The parts are a combination of Rollco parts, aluminium parts, and 3D-printed parts. The ball-screw provide the vertical motion, and two parallel c-rails will ensure smooth motion, whilst holding the refuse bin in a steady horizontal orientation. The angle irons are mounted at the base and provide support while holding the parts in place. These provide an angle close to 90 degrees, and a sturdy connection to the base plate. At the top, a special piece has been designed and constructed to provide stability for the top of the ball-screw. Underneath the base plate, two interconnected cog-wheels are located. These are designed to transmit the motion from the motor to the ball-screw.



(a) Rail mount

(b) Rail tower

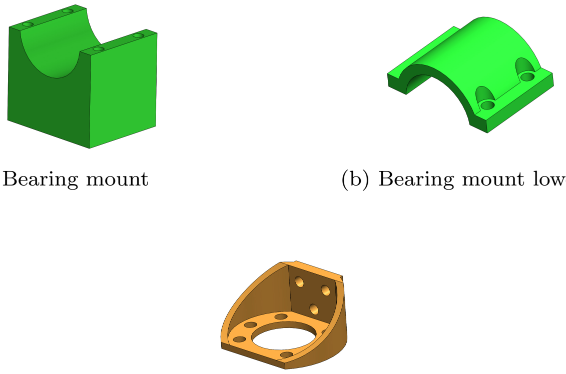
(c) Angles

(d) Top

(e) Clamp

Figure 4: The figure show the parts modelled for the vertical motion of the lift system. The parts are created in aluminium, iron, and PLA plastic.

The parts needed for the horizontal motion can be seen in Figure 5. These parts are designed to fit the horizontal axis motion parts from Rollco to the structure. The parts from Rollco that need to be fastened is the ball bearings as well as the nut on the ball-screw. Thus, the bearing mounts and the clamp has been designed. The upper and lower bearing mounts have the function of tightening the bearing in the space in between. The clamp has similar functionality as the ball-screw.



(a) Bearing mount

(b) Bearing mount lower

(c) Clamp lower

Figure 5: This figure demonstrates the created parts for the horizontal motion. The bearing mounts are made to connect the base plate to the stainless rods and the ball bearings. The Clamp is made to connect the base plate with the horizontal ball screw.

3.3.3 Rollco parts

The parts for all the linear motions were provided by the company Rollco. As well as providing the parts, they also contributed with very valuable input to the proposed solution. Rollco possesses many products of varying sizes and functionality, and therefore brief research was conducted to find those that seemed most reasonable.

The mindset when searching amongst their products were to find parts that were as small and light as possible while remaining high robustness. In Figure 6 all Rollco parts are listed according to their product name in its catalogue.

The vertical and horizontal motion possess two low-friction parts and one ball-screw respectively. Apart from that, the rest of the parts are mounts to attach the structures.

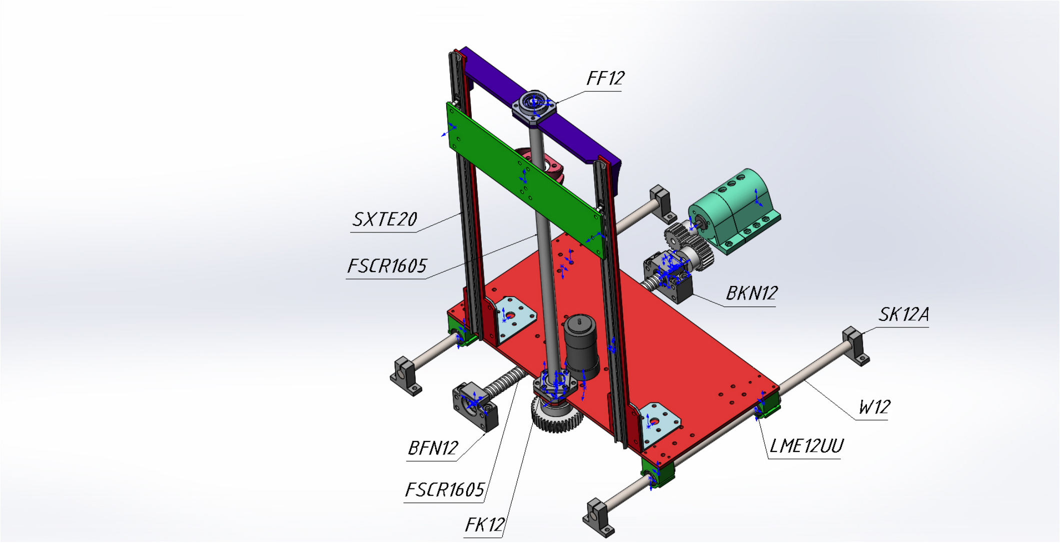


Figure 6: This figure display the corresponding product name for the ROLLCO parts.

3.3.4 Avans parts

Avans is a company that distribute mechanical motion parts. They supplied us with gears that were used with the motors. Four gears were ordered. Two of them had 20 cogs, one had 40, and one 26. All gears were equipped with stop screws for proper mounting.

3.3.5 Torque calculation

Author: Mujtaba Hasanzadeh
In order to choose a suitable DC motor for the lift system, the torque of the ball screws should be obtained. The lift system has a range of movement in two different axes. The following formula has been used for both vertical and horizontal motion.

$$\tau = \frac{m * g * L}{2\pi\nu} * 10^{-3} \tag{2}$$

Where

τ: The torque (Nm) of the system in y axis
m: The mass (kg) of the bin
g: Gravity acceleration (9.82 m/s^2)
L: Lead (mm)
ν: Normal efficiency (0.90-0.95)

For the vertical motion, the friction of the c-rails are too small and can be neglected. The total mass that has been considered in this case is 16 kg which includes a full bin and the c-rails and the mounts. The lead of the ball screw is 5mm and the normal efficiency is considered to be 0.9. The torque of the vertical motion using formula 2 is:

$$\tau = \frac{16 * 9.82 * 5}{2\pi * 0.90} * 10^{-3} = 0.14Nm \tag{3}$$

In the horizontal motion some more weights need to be taken into account such as the weight of the base, ball screw, bin etc. The total mass is approximately equal to 20 kg and all other variables are defined as in the vertical motion. By using the equation above the obtained torque for the horizontal motion is:

$$\tau = \frac{20 * 9.82 * 5}{2\pi * 0.90} * 10^{-3} = 0.17Nm \tag{4}$$

The results of the calculation show that a DC motor that has a minimum torque of 0.14 Nm for the vertical motion and a motor with a minimum torque of 0.17 Nm for the horizontal motion should be chosen in order to drive the lifting system. There are two possible ways to use the motors in the lift system. The motors can be directly connected to the screw rods or indirectly connected using gears, this depends on how much the selected motors can be loaded on its axial shaft.

3.3.6 Motor

Two motors are needed to drive the lift system in the vertical and horizontal axis. The possible motors which can be used in this application are stepper, brush-less and brushed DC motors. Table 1 shows some of their pros and cons.

	Advantages	Disadvantages
Stepper	* High torque density at low speeds * Repeatability * Low cost	* Noisy * Can skip steps * Draws continuous current
Brushed	* Simple speed control * Low cost * High power density	* Medium lifespan * High maintenance
Brushless	* High efficiency * Low maintenance * Long lifespan	* Expensive * Higher drive complexity

Table 1: Shows some of advantages and disadvantages of the mentioned motors

Based on their advantages, disadvantages, cost and drive complexity, a brushed DC motor is preferred over the others. Selecting a suitable motor requires the specification of the necessary torque, current, voltage, as well as the consideration of speed with and without load. According to the requirements above, the selected motors are two identical 12V geared brushed DC motors with a torque of 0.2Nm [6] which will be used together with external gears to drive the lift-system. The main reason for using gears is that the rods cannot be attached directly to the motors since the motors have a maximum axial shaft load of 100N which is less than the force created by the lift system.

3.3.7 Gears

As mentioned earlier, overloading the motors' shafts needs to be avoided. To avoid overloading, gears are connected to the motors and to the screw rods. Since the torque of the motors are large enough, the possible gear ratio is 1:1. However, it is good to amplify the torque of the motors anyway in order to have a satisfactory tolerance range. Gear ratio tells how much the torque of the motor is amplified and how much slower it operates. For example, a 2:1 gear ratio for a motor will result in a two-time torque amplification but a two-time speed decrease. For the vertical motion where the bin will be picked from the wall, the chosen gear ratio is 2:1 which means that the torque of the motor is amplified to 0.4Nm and for the horizontal motion, the chosen gear ratio is 6:5 i.e. 1.2:1 which means 0.24Nm.

3.3.8 DC to DC converter

As mentioned earlier, the needed voltage for the motors is 12V but the PDB can only generate 3.3V, 5V, and 15V, according to this a DC/DC converter was needed to converts the input voltage from the PDB into 12V. The conversion can be done either using a linear regulator or a switching regulator. A linear regulator uses a closed-loop to regulate the output voltage while in switching regulators, the transistors alternate between on and off to generate the desired output [7]. Table 2 shows some of the pros and the cons of these regulators.

	Advantages	Disadvantages
Linear regulator	Low noise Low cost Simple	Only step down operation Poor efficiency
Switching regulator	High efficiency Step up/down operation	Complex design Higher noise

Table 2: Some of advantages and disadvantages of the linear and switching regulator

In this case, either a step-down or step-up converter can be used. A step-down converter converts a voltage from a higher voltage to lower voltage and vice versa for a step up converter. Based on the pros and the cons of the different regulators, an isolated switching step down/up converter has been selected in this application [8]. An isolated DC/DC converter provides more safety and increases noise immunity. The selected DC/DC converter operates between 9V and 36V and generates 12V and the maximum output current is 6A with a effect of 72W which indicates that all requirements are fulfilled to deliver the desired current and voltage for the motors .

To integrate the converter into the lifting system and connect it to the PDB and the motor shield, a PCB was made. Figure 7 shows the designed circuit board which is created in Ultiboard.

3.3.9 Micro-switch

Variation of weights in the bin affects the execution time and the speed of the motors. This makes it harder to define the execution time of the motors in advance. To deal with this issue, micro-switches are adopted into the lift system to start, stop and change the direction of the motors. A micro-switch is an electronic switch which needs a little physical force to get actuated. The micro-switches data-sheet can be found on the retailer's website [9].

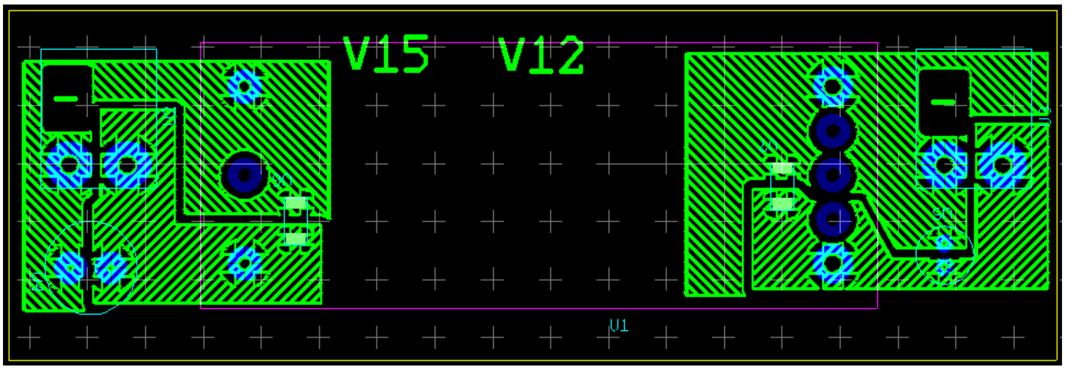


Figure 7: The input and output side are marked with 15V and 12V respectively, as well as negative pins with a minus sign. According to the converter's specification, there were four pins that were not needed to use which can be seen in the picture that they are not connected to the plane.

3.3.10 Motion controller

The motion of the lift-system is controlled by an Arduino Uno [10] attached with a motor shield [11]. Four micro-switches are attached to the lift-system on different positions. These micro-switches act as end of travel sensors and control the direction of the motors. The motion is divided into four different states. The lift system starts in the idle state and waits to get a message. Switching from one state to the next state occurs when a micro-switch is pushed and when lifting is done, it will send a message to the main system. Figure 8 shows a block diagram over the lift systems functionality.

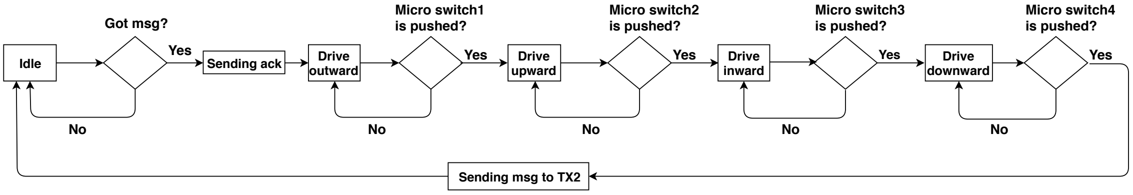


Figure 8: The block diagram of the lifting system's motion

3.4 Software

Author: Mujtaba Hasanzadeh

Since the micro-controller is an Arduino board, the software that is used to program it is Arduino IDE while the main computer uses ROS, as mentioned earlier. To use ROS on an Arduino and to be able to communicate with the TX2 through USB-serial, ROS library and rosserial packages are used. Two nodes are created, one in the TX2 and another one in the Arduino as it is shown in Figure 9. The node in the TX2 is added into the UNICORN state machine implementation and a state for the lifting is defined. When the lifting state is chosen, the node becomes activated and sends a message to the lifting system. When the lifting system gets the message, it will send back an acknowledgement to the TX2 and start to processing. Once the lifting is done, the lifting system sends a message to the TX2 that it has accomplished the lifting and the robot can move further.

3.5 Tests

This section describes the two tests performed. The tests were developed to evaluate the lift system itself, and its effect to Charlie when mounted.

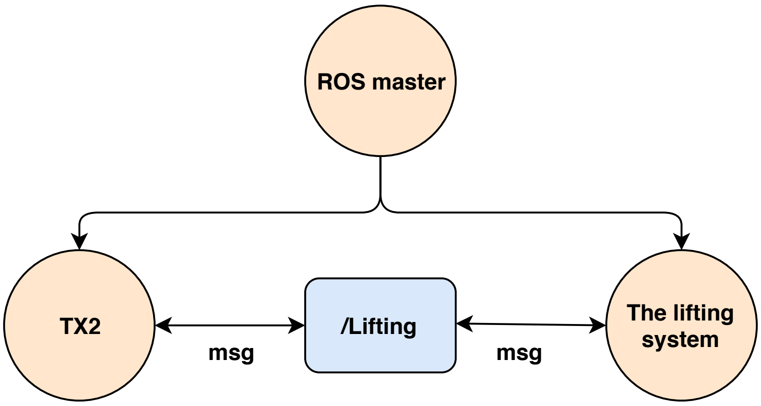


Figure 9: The nodes and the topic are outlined with circles and rectangle respectively. Each node has its own publisher and subscriber that publishes and subscribes on the Lifting topic. The used baud rate for this application is set to 57600.

3.5.1 Lifting

Author: Sebastian Andersson
The test for the lift system is developed to see the differences in lifting time between weights, to evaluate its performance.

3.5.2 Setup

The bin will be loaded with many smaller pieces, to achieve the desired weight, to simulate how a bin could be loaded when in use. Thus, an even distribution of weight in the bin is desired.

The test would be considered as a failed test if the system would not reach the top of the lifting capacity so that no time for the lifting could be collected. Otherwise, when the test pass, the time for the lifting operation would be measured, from the lowest to the highest of lifting capacity, in seconds. Furthermore, the tests will operate with an increment of the payload with 3kg at a time, starting at 0kg and ending at 15kg.

The collecting of time will be performed with the software. This is to try to ensure equal timing opportunities for all test cases and iterations.

3.5.3 Expected outcome

An expectation in limitations is that all weights will pass the tests. The expectations on time for the lifting operation is that there is no great deviation between weight distributions. Furthermore, it is expected that the time to lift the bin will be affected according to the increase in weight.

3.5.4 Transportation

Another aspect that requires testing is the transportation of refuse bins. The test is designed to see how different weights of a refuse bin in different inclinations alter the movement of Charlie.

3.5.5 Setup

The refuse bin is to be loaded with weights between 0kg and 15kg, with a 3kg increment with each test iteration. The position of the bin will be at the rightmost position, i.e. as close to the driving wheels as possible. To measure the results, the time it takes for Charlie to travel a distance of 6.3m is to be investigated. The distance will have an inclination, and Charlie will transport the bin from both directions, such that the inclination will be first upward, and then downward. The timing will be performed by observing the moment Charlie pass the start and finish point, and record the time stamps at those occasions to find the time duration.

The test will be performed with three repetitions for each weight and direction, to produce a more reliable mean value.

3.5.6 Expected outcome

A prediction is that for the same position of the bin, but with different payloads, the time should vary according to the difference in weight. Furthermore, the upwards direction is predicted to have a longer mean value in time duration than the downward direction.

3.6 Results

The results for the tests are demonstrated in this section. In Table 3 the lifting test results are demonstrated, and in Table 4 and Table 5 the results for the transportation tests can be seen. Table 4 demonstrate the transportation test where the path is in an upwards inclination, whereas Table 5 show the results where the path is in a downwards inclination.

Lifting	**1**	**2**	**3**	**Mean**
0	5900	5900	5800	5867
3	6200	6100	6100	6133
6	6500	6401	6300	6400
9	6900	7001	6900	6934
12	8000	8100	8000	8033
15	8700	8650	8700	8683

Table 3: This table demonstrate the sampled times for the lifting test in milliseconds. The columns show the three test iterations, where the fourth column show the mean of the three. The rows represent the corresponding weight of the bin in kilograms.

Upwards	**1**	**2**	**3**	**Mean**
0	21.24	21.36	21.32	21.31
3	21.25	21.31	21.26	21.27
6	21.28	21.35	20.45	21.03
9	21.38	21.30	21.08	21.25
12	21.58	21.52	21.31	21.47
15	21.26	21.51	21.52	21.43

Table 4: This table demonstrate the sampled times in seconds between test iterations for the transportation test in an upwards inclined path. The columns represent the number of times the test was repeated, whereas the rows represent the weight of the bin in kilograms.

Downwards	**1**	**2**	**3**	**Mean**
0	20.54	20.40	20.57	20.50
3	20.29	20.32	20.39	20.33
6	20.58	20.14	20.91	20.54
9	20.37	20.37	20.40	20.38
12	20.35	20.99	20.66	20.67
15	20.31	20.66	20.73	20.57

Table 5: Similar table as table 4, where the travel time for Charlie in an upward slope is evaluated.

3.7 Discussion

Author: Mujtaba Hasanzadeh

The tests showed that the lifting system is able to lift a bin with a weight of 15kg and that the time it takes to lift is increased with the added weight. But looking at the results of driving upwards and downwards, a positive aspect is that almost the same drive time is obtained while increasing the weights in the bin. This indicates that adding more weights will not affect the execution time of the motors and Charlie's velocity because its speed controller is sufficiently sophisticated. Currently, there is no system for aligning the robot with the bin, and thus some issues where observed. For instance, if Charlie was not properly aligned under the bin, the lifting operation would not be successful. Sometimes the bin would be positioned incorrectly, and sometimes it fell off completely. Apart from that, the lift system performs as expected.

4 Human robot interaction

Author: Alexandra Hengl

In the future, as it is anticipated in this project among others, humans and autonomous robots will co-exist in the same space. Human-robot-interaction is thus a curious interdisciplinary topic that gives rise to many questions regarding safety and efficiency. Some of the important questions are: What are the effects of co-existing robots on human psychology and behaviour? What are the best practices for human-robot communication and collaboration? How to avoid accidents?

There are numerous studies that investigate different aspects of human-robot communication. In a study conducted by Dragan *et al.* [12], it is shown that, in scenarios where the robot's goal is unknown to the human participant, legible motion can enhance the subjective and objective effectiveness of human-robot collaboration. This phenomenon is also observed in the study of Watanabe *et al.* [13] and Chadalavada *et al.* [14], where motion- and navigation intent are investigated by projection of planned trajectory. Scenarios in which the navigation intent of the robot was made clear, enabled the human participants to react sooner.

In each of the above studies the results are promising, and in addition to that, the implementation of the latter study resembles that of the UNICORN project: Both of them involve freely moving autonomous robots and seek solution specifically for path intention communication. Since Chadalavada *et al.* [14] utilised a projector based solution with promising results, the UNICORN project has also aimed to implement a similar practice. The following subsection, 4.1, will give an insight into the hardware (projector) related considerations, while 4.2 will explain the software implementation.

4.1 Projector considerations

There are lots of technical differences between projectors that make them more or less adequate for different types of use cases. In this subsection, the properties that are most important for the particular use case of *this* project, are presented and discussed.

Size and weight

When it comes to projectors, a larger form factor usually accommodates more numerous technological features with less compromise than a tiny form factor. However, size affects weight, which can quickly become an issue on a mobile robot-like Charlie. Therefore, "pico" type projectors, that usually weigh around 0.5 kg, have been considered.

Brightness

This property affects the visibility of the projected image. Brightness will be reduced when increasing the distance between projector and projection surface. The brightness of the image is also affected by different light conditions. A strongly illuminated or sunny environment requires a more powerful projection source than an indoor or night environment. The brightness property has greatly varying values among projectors. Usually, there is a linear correlation between it and the size and price factors. Brighter projectors also operate on more power, which is rather limited on a mobile device.

Image size

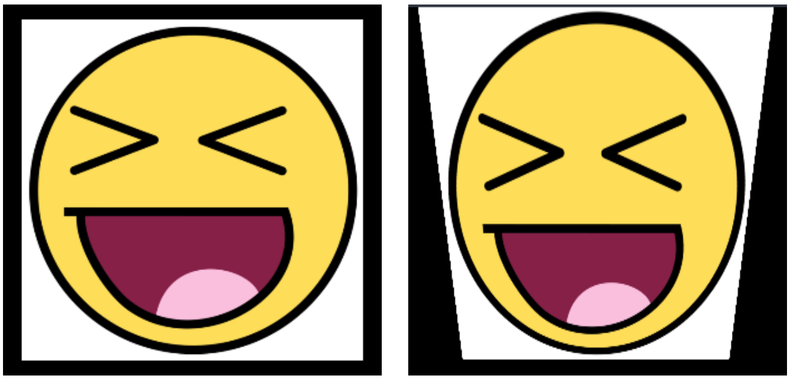
The projected image size can be manipulated by increasing or decreasing the distance between projector and projection surface. At the same time, at different distances, the focal length of the lens varies, which results in an increasingly blurry image. To correct this effect, re-adjustment of the focus value of the lens is required. However, there is a minimum and a maximum limit to the projector's focusing ability. Thus, the minimum and maximum image size is the consequence of the projector's focusing ability at the shortest and the longest distances from the projection surface. It is important to consider the required image size and choose a projector that is able to produce a sharp image with that size.

Throw distance

The minimum and maximum throw distance are the distances at which the projector is able to produce the minimum and the maximum image sizes. Projectors that are able to produce a relatively large image from a relatively short distance, are called ultra short throw. They usually utilise a special mirror for indirect projection. These kind of projectors are beneficial in spatially constrained situations. In the UNICORN project, the projector is placed relatively close to the ground. Therefore, a short throw projector is preferable, so that the projection can be sufficiently large despite the short distance between the projector and the ground.

Resolution

The resolution property tells the number of pixels and the horizontal and vertical layout of those pixels. The larger the projected image, the larger the area is on which the same number of pixels are laid out. Resolution becomes particularly important when the lens is not perpendicular with the projection surface. In such a setup, the image becomes distorted: It will be stretched out and its rectangular shape becomes a trapezoid (see Figure 10). In result, most of the pixels will be small and concentrated at the bottom of the image, and fewer, larger pixels will be laid out on a much larger area toward the top of the image. While it is possible to regain the correct angles in the picture by software-wise image processing, the layout of the pixels themselves, cannot. A projector with higher pixel count on the vertical axis will distribute more pixels toward the top of the image than a lower pixel count projector. This can give more possibilities to correct the image by software processing.



(a) Correct image (b) Distorted image

Figure 10: Image distortion

The chosen projector

The final choice of projector has been an Optoma ML750st LED pico projector (see Figure 11). It has a compact size of 11x12x5.7 cm and light weight, 420g. It has a maximum brightness of 800 ANSI lumens. This is a short throw projector with a native resolution of 1280 x 800 pixels and a minimum image size of 63 cm at a minimum screen distance of 43 cm.

Figure 11: https://images-na.ssl-images-amazon.com/images/I/71cwRKjaSJL._SL1500_.jpg

In addition to the hardware, a software application is needed to read and process available data about the movement of the robot and, to handle graphics for the path visualisation. In the next section, the software solution for the robot intention communication will be discussed.

4.2 Implementation

The goal of the path intention communication is to give the human counterpart an early hint about the robot's planned path. In this project, this is done by projecting an approximated path in front of the robot. This path is represented by a line whose curve intends to approximate the actual turning angle. The length of the projection should provide a couple of seconds worth of insight into the future. The key steps taken in the method of implementation are the following:

- The *cmd_vel* ROS topic is utilised to get information about the angle of a progressing turn.
- A set of static images (see section 4.2.1 Figure 13) are used to represent different turning angles.
- Since there are too many possible turning angles, the image representation is limited to a defined set of angular intervals.
- Additionally, image transformations are enabled to counteract image distortion that arises due to the angled positioning of the projector lens to the projection plane.

The remaining parts of this section will give more details about the steps above.

4.2.1 Utilising the *cmd_vel* ROS topic

Messages of type *geometry_msgs/Twist* are published on the cmd_vel topic in order to give velocity commands to the wheels. These messages include information about the speed and the angle of turn of the robot, in terms of x, y and z-coordinates. For instance, a message: *linear: [x:0.9 y:0 z:0] angular: [x:0 y:0 z:-0.5]*, is interpreted as the robot is moving forward on the ground plane with 0.9 m/s and at the same time, turning right with 0.5 rad/s.

The application created subscribes to the *cmd_vel* topic and reads the *geometry_msgs/Twist* message to know the current direction of the robot. However, there are some challenges to consider when relying on this particular source of information. Firstly, these messages contain a lot of noise. On the way to the goal position, the robot makes very fine adjustments to its movement, changing its direction from left to right. Secondly, these adjustments happen at a rate of 10 Hz. If every message was read and, at the same pace, the image that represents the current direction was changed, a fast flickering of images would be seen. To reduce the effect of sudden changes in direction, five angular intervals have been defined (see figures 12 and 13): between the values -0.2 and 0.2 radians, the grade of the turn is minimal enough so that it can be considered *straight*. A turn with a value between -0.2 and -0.5 is considered a *soft right* turn, and under -0.5, a *sharp right* turn. A *soft left* turn is one between 0.2 and 0.5, and *sharp left* above 0.5. Furthermore, it is realised that reading two messages per second creates a short enough reaction time for the directional changes.

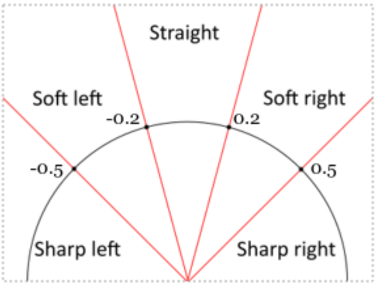
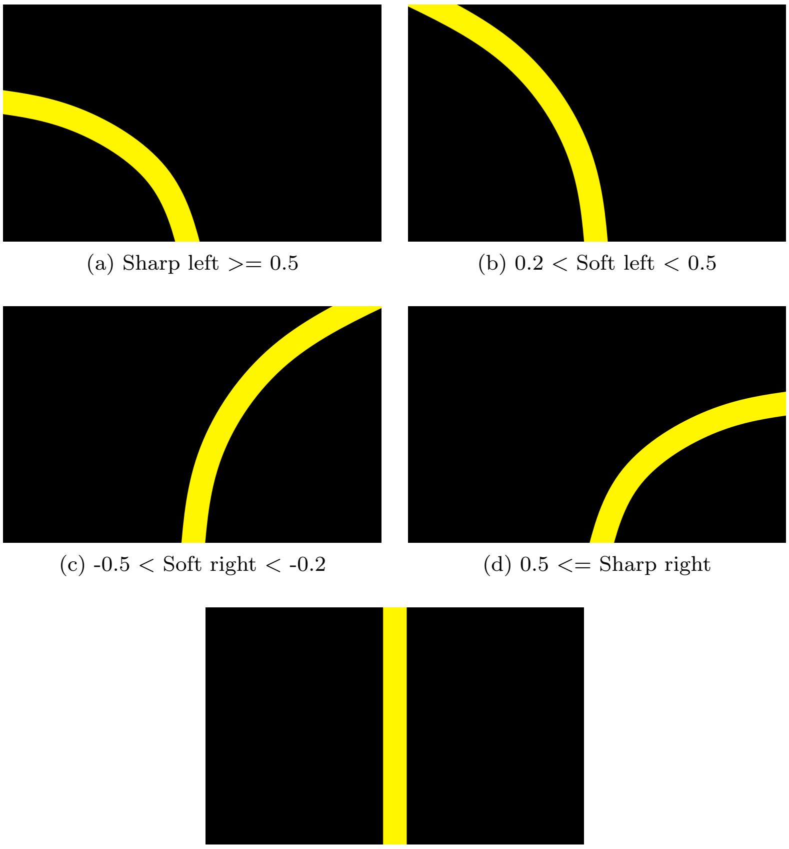


Figure 12: The five defined angular intervals. The values are in radians.



(a) Sharp left >= 0.5

(b) 0.2 < Soft left < 0.5

(c) -0.5 < Soft right < -0.2

(d) 0.5 <= Sharp right

(e) -0.2 <= Straight <= 0.2

Figure 13: The five directions paired with the five defined angular intervals. The values are in radians.

4.2.2 The GLUT framework

The GLUT framework is utilised to create the necessary conditions for displaying graphical user interface elements such as a window in which the projection image is placed. Moreover, this framework is responsible for responding to events, such as keyboard events, timer events and requests for re-displaying graphical elements.

Keyboard events are enabled for applying manual image correction using the keyboard. Timer events are utilised for the periodic reading of ROS messages on the *cmd_vel* topic. *Redisplay* events are required for changing the displayed image in case of a change in the robot's moving direction.

4.2.3 Image transformations with OpenGL

Due to the projector mount, the projector lens' position is not perpendicular to the projection plane. Thus, the projected image appears distorted on the surface. The OpenGL library is employed in the software application to provide tools to counteract image distortion.

The images that represent the intended direction, are loaded into the application as OpenGL textures. The procedure for showing textures in OpenGL requires that a set of vertices are defined first. In this application, the vertices are defined to form a rectangle. This rectangle is used as the surface on which the textures are interpolated. OpenGL provides transformation tools, such as scaling, rotation and translation, which are applied to the object defined by vertices. For instance, to correct the image distortion caused by the angle in which the projector is rotated, the rectangle object also has to be rotated with the same angle as the projector but in the opposite direction. Enabling the GLUT keyboard events makes it possible to assign different OpenGL transformations to different keys. Using the keyboard, one can manually set the image so that it regains its original proportions and size.

5 Localisation with UWB

Author: Marielle Gallardo

In this section ultra wideband technology and its possible use for localisation will be presented and discussed.

5.1 Background

Today there is an increasing demand for autonomous robots to serve and interact with humans indoor and outdoor [15]. Thus accurate localisation of these machines is of utmost importance. However, achieving high precision, particularly in an indoor environment, is a huge challenge. This is a result of signals being scattered and reflected by various objects. Active research has been conducted to address this problem by employing various existing technologies [16].

Ultra-Wide Band (UWB) is an emerging radio technology that allows for high time-resolution, accurate positioning, low power consumption, and great penetrating capabilities [17]. This is due to UWB operating over a wide spectrum of frequencies while emitting short-time pulses. The characteristics of the pulses, as illustrated in Figure 14, makes the signals easily distinguished even in a noisy environment. The effect of multipath interference is similarly minimised since reflected signals do not affect the direct signal. Unlike narrow-band wireless technology, such as WiFi, that provide localisation down to a meter, UWB provides the capability for cm-level accuracy [18]. Figure 15 illustrates how UWB operates in comparison to other technologies.

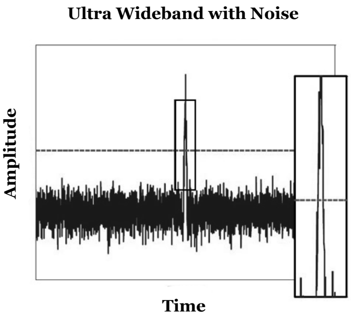


Figure 14: The UWB pulses are easily distinguished, even in a noisy environment, due to the steep characteristics of the signals.

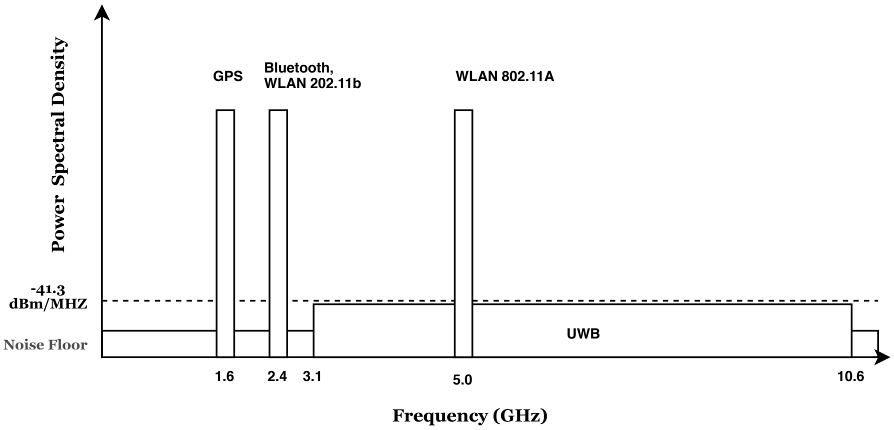


Figure 15: Frequency chart of how UWB operates compared to other popular wireless communication standards.

In order to utilise wireless technologies for position estimation, several positioning algorithms have been developed. These algorithms extract information from the radio signals sent between reference nodes and the target node. One technique named Time of Flight (ToF) measures the distance by calculating the time taken for the signal to travel between the sender and receiver. Algorithms that rely on ToF are Angle of Arrival (AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Two-Way Ranging (TWR). Received Signal Strength (RSS) is another technique for determining the position of a target. Instead of determining the time difference from the sender and receiver the received signal strength and propagation of the signal sent between the transmitter and receiver are utilised for position estimation. The algorithms mentioned differ in accuracy, range, application, robustness and complexity [19]. Therefore careful consideration should be taken while determining the suitable algorithm to utilise. For UWB systems the techniques that are dependent on ToF is preferred since they provide more precise measurements compared to the RSS method. The wide frequency of UWB allows for fine time-resolution which is an important factor for achieving accurate localisation of objects [20]. A summary and comparison between the algorithms are presented below.

5.1.1 Angle of Arrival

In the AoA method, the direction of arrival of signals is estimated utilising an antenna array. The intersection of the angle direction lines is then calculated to get the position of the target, as can be seen in Figure 16. The main advantages with the method are that synchronisation is not required between the nodes and that only two reference nodes are needed for 2D positioning estimation [21]. The complexity of the AoA technique is higher compared to other methods. This is due to the need to utilise several antennas for measuring the angles, which increases the cost and size of the AoA system. Furthermore, accuracy decreases as the distance between the reference nodes increases. Additionally multi-path causes large carrier phase error at the antennas making the method not function well indoors [21].

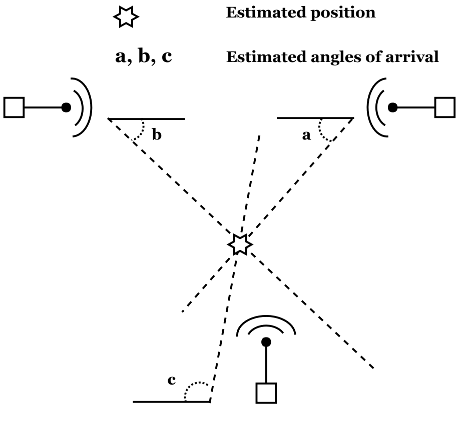


Figure 16: In AoA the intersection of the estimated angle direction lines is utilised to calculate the position of the target.

5.1.2 Time of Arrival

In the ToA method, the signal propagation time is utilised to derive the distance between the sender and the receiver. Positioning estimations are done by calculating the intersection of the circles of the reference transmitters, as depicted in Figure 17. The radius of the circles is the distance measured between the target and the reference node. The complexity of the ToA technique is high since time synchronisation is required between all nodes and achieving optimal synchronisation is difficult due to clock drift [22]. Any delays in the system must be accounted for to calculate correct distance measurements. Furthermore, the method requires complex geometric computations that are computationally heavy [23]. However, the ToA method offers very high accuracy if assuming accurate clock synchronisation [24].

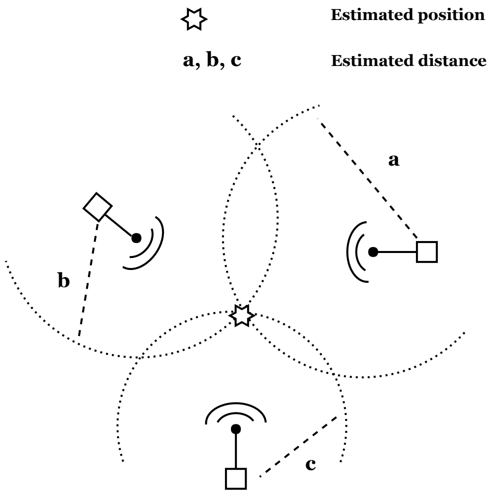


Figure 17: In ToA the position estimation is calculated from the intersection of the circles surrounding the reference transmitters.

5.1.3 Time Difference of Arrival

Position estimations in the TDoA technique are made by calculating the delta between the arrival times of two transmitted signal and finding the intersection between the hyperbola branches defined by the time-of-arrival differences. The processes of estimating the target utilising the TDoA method together with three reference nodes can be viewed in Figure 18. Unlike the ToA method, synchronisation is only required between the reference nodes. The method has very high accuracy if precise synchronisation is obtained between the receiving nodes. However, it is less accurate than the ToA technique assuming same system geometry [19].

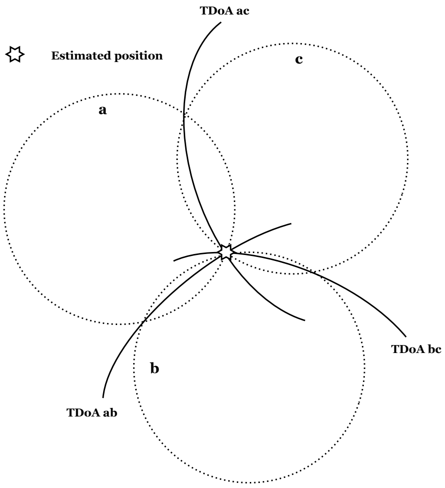


Figure 18: The difference in arrival time can be used in TDoA to calculate the difference in distances between the target and the reference transmitters.

5.1.4 Two-Way Ranging

The Two-Way Ranging method measures distance by extracting time-stamp information from the bi-directional message exchange between the target and the reference nodes, as illustrated in Figure 19. Distance and ToF information can be extracted from the message exchange with Equation 5 and 6

$$ToF = ((t4 - t1) - (t3 - t2) + (t6 - t3) - (t5 - t4)) \tag{5}$$

$$Distance = ToF \times c \tag{6}$$

where c is the speed of light. The target can then be located by either calculating the intersection of the circles of the reference nodes or by deriving the delta between the arrival times of two sent signals. The Two-Way Ranging technique eliminates the need for synchronisation since time-stamp information is obtained by exchanging messages [25]. However, the message exchange can be resource intensive. Furthermore, the method is less scalable compared to other solutions since the target is obligated to know the address of every reference node to initiate communication [26].

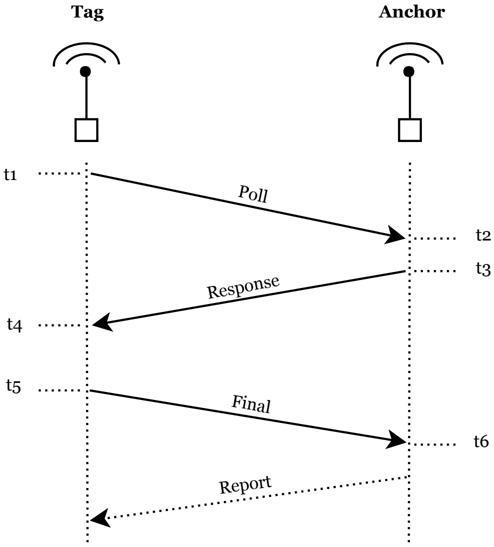


Figure 19: In TWR the requirement for synchronisation is eliminated with the exchange of messages containing time-stamp information.

5.1.5 Received Signal Strength

RSS as a positioning system measures distance by calculating the received signal strength from the target to several reference nodes. This information is then utilised together with a radio wave propagation model. An example of a propagation model is presented in Equation 7

$$\bar{P}(d) = P_0 - 10n \times log_{10}\left(\frac{d}{d_0}\right) \tag{7}$$

where the strength of the signal propagated through a medium is decreased by path loss which is formulated as proportional to the distance between the target and reference node, and the received power $\bar{P}(d)$ at distance d is calculated by the received power P_0 at the reference distance d_0 and n is the path loss exponent, whose value is normally in the range of 2 to 4.

This method is easy to implement however prone to error since the received signal strength is vulnerable to various environmental conditions in addition to path loss, e.g. shadowing and multi-path. As such it is difficult to estimate all conditions that affect the RSS [27].

5.2 Hardware

There is a wide range of low-cost UWB products available that offer different capabilities and features. Jiménez *et al.* [28] made a performance comparison between three UWB systems developed by Ubisense, BeSpoon and, Decawave respectively. The performance analysis was conducted in an industrial indoor environment with Line-Of-Sight (LOS) and diverse Non-Line-Of-Sight (NLOS) conditions. By observing the LOS and NLOS measurements conclusions could be drawn regarding the UWB component's behaviour in open (non-obstructed) as well as obstructed environments. A Bayesian filter with a particle filter and a measurement model was also implemented to account for bad measurements and identify outliers in the 3D positioning estimations made by the different UWB systems. The authors concluded that the Decawave system provides a significantly more accurate positioning performance compared to the Ubisense system. The error distribution between the Decawave and BeSpoon system was similar. However, the Decawave system still had slightly more reliable measurements.

5.2.1 Decawave DWM1000

The Decawave DWM1000 module is a low power chip that incorporates a UWB transceiver IC, antenna, radio frequency circuit, clock circuit and power management in one single chip. It is fully compliant with the IEEE 802.15.4-2011 UWB standard. The module can operate in four different radio frequency bands between 3.5 GHz to 6.5 GHz and supports data rates of 110 kbps, 850 kbps, and 6.8 Mbps. The positioning techniques that can be utilised are TDoA and Two-Way Ranging with the capability of providing ranging measurements with an accuracy of ±10 cm [28]. The transceiver chip is a real-time location system (RTLS) as targets are located in real time [29].

5.2.2 UWB breakout boards

Author: Ulrik Åkesson

In order to use the DWM1000 UWB modules mentioned in section 5.2.1 an Arduino breakout board needed to be created. The design for this board is based on a similar design by GitHub user J. M. Jimeno [30]. Jimeno's design is built around an Arduino Pro Mini and together with Decawave DWM1000 tags and anchors were created.

The design in this project uses an Arduino MKRZero as the interface to the UWB module. The MKRZero has a couple of features which makes it suitable for the project where the most prominent are presented in Table 6 where the MKRZero is compared with the Nano and the Pro Mini which are common similarly sized Arduinos. The block diagram in Figure 20 describes how the Arduino and DWM1000 modules communicate with one another.

	Pro Mini	Nano	MKRZero
Processor	ATmega328P	ATmega168* ATmega328P	SAMD21 Cortex-M0 + 32bit low power ARM MCU
CPU speed [MHz]	8 16	16	48
EEPROM [kB]	1	0.512* 1	No
SRAM [kB]	2	1* 2	32
Flash [kB]	32	16* 32	256
Support for LiPO battery	No	No	Yes

Table 6: Comparison between the Pro Mini, Nano, and MKRZero arduinos. All data come from the arduino website [31].

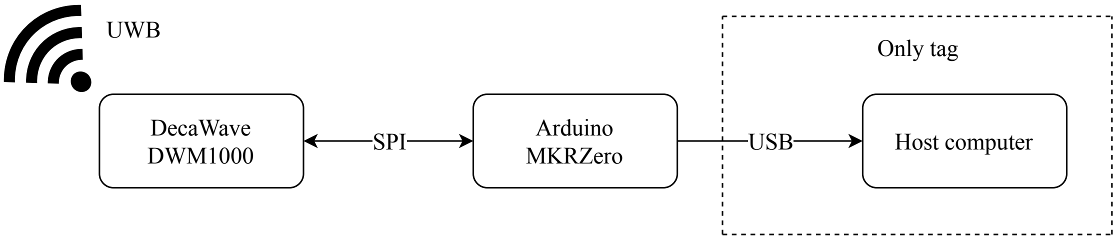


Figure 20: Both the tag and anchors are based around a DWM1000 module which communicates with the MKRZero using the SPI protocol. The DWM1000 is powered by the Arduino and communicates with the tag/anchor using over UWB. The anchors are powered by 3.7V LiPo batteries. The tags communicate with and are powered by the host computer over USB.

5.3 Software

Author: Marielle Gallardo

The software for localising the robot in the environment with UWB is mainly based on two standalone ROS packages. The *ros_dwm1000* package created by J. M. Jimeno [30] implements software that allows the Decawave DWM1000 modules to communicate with each other and locate one UWB module configured as the target. Modifications were made to incorporate the package into the existing robot software. However, to integrate the UWB pose with the existing localisation system in ROS, the *robot localisation* package created by T. More and D. Stouch [32] was utilised. The package allows developers to modify the ROS 2D navigation stack primarily responsible for mapping and navigating the robot. In the default navigation stack solely odometry and scan information are used for mapping and localisation [33]. The robot localisation package provides the capability to fuse an arbitrary number of sensors to estimate the state of the robot with the aim to improve the accuracy of the estimations [32]. A further discussion of how the distinct packages operate is presented below.

5.3.1 Algorithms

The UWB localisation software by J. M. Jimeno performs two-way ranging and multilateration to derive the distance of the target relative to the reference nodes placed in the environment. Multilateration is carried out once enough reference nodes are found to determine the target's position by utilising the geometry of circles and calculating the intersection of the circles. The distance can be described using Equation 8

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \qquad (i = 1, 2, ..., n), \tag{8}$$

where i detonates the anchor number, n the total number of anchors, x, y, z are the coordinates of the target node and x_i, y_i, z_i are the coordinates of the anchor nodes. The relative pose of the target coordinate frame related to the world frame is then sent to the ROS system by broadcasting a transform between the frames.

In ROS the robot is able to localise itself in the world with the help of the navigation stack. The robot is assumed to be configured in a particular manner to function as intended [34]. The diagram presented in Figure 21 shows an overview of the indicated configuration. The navigation stack determines the position and rotation of the moving robot in 2D space with a probabilistic localisation system named Adaptive Monte Carlo Localisation (AMCL). The localisation system utilises a particle filter to visualise the uncertainty about the robot pose and each particle contains a full description of a possible future pose. The cloud of particles shrinks in size as the robot moves around in the environment. Particles that do not match additional scan and odometry readings are discarded allowing AMCL to refine its estimate of the robot's pose.

In order to utilise the UWB pose to further refine the estimation of the robot pose, the *robot localisation* software was configured. The software works by continuously tracking and updating the 15-dimensional state of the robot $(x, y, z, roll, pitch, yaw, \dot{x}, \dot{y}, \dot{z}, \dot{roll}, \dot{pitch}, \dot{yaw}, \ddot{x}, \ddot{y}, \ddot{z})$ as it receives measurements from different sensors. In addition to the navigation stack, that only utilises

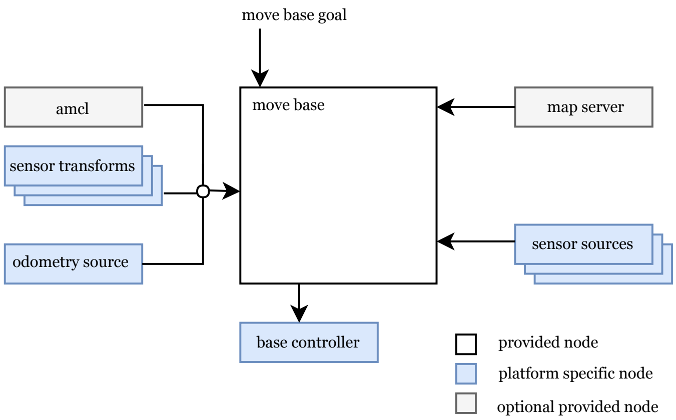


Figure 21: Simplified diagram of the robot setup for the ROS navigation stack

one node to provide the pose estimation, robot localisation can fuse several estimate nodes to provide the absolute position of the robot. In this configuration, the AMCL and UWB pose were fused together with existing odometry data to provide a more accurate estimation. An overview of the configuration can be viewed in the diagram presented in Figure 22. A more detailed description of the implemented setup is presented below.

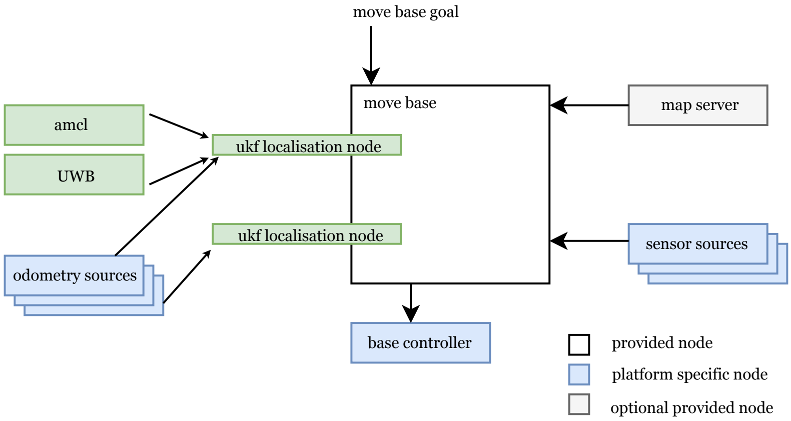


Figure 22: Two-level UKF robot setup operating with the ROS nagivation stack

5.3.2 Implementation

Both AMCL and UWB provide global absolute position data. In order to fuse their data several configurations needed to be implemented.

The robot location package fuses sensors by subscribing to specific geometry messages that provide pose information over ROS. The UWB position was therefore published by broadcasting a PoseWithCovarianceStamped geometry message containing pose information and a covariance matrix. Earlier configurations on the robot were already fusing odometry data from the sonic sensor and wheel encoders utilising an unscented Kalman filter (UKF) state estimation node. An extra UKF node was added to create a two-level UKF setup.

The first instance is responsible for fusing odometry data to provide the odom to base link transformation. The next instance fuses the same odometry data together with absolute position data to provide the map to base link transformation. With these settings, data provided by UWB and AMCL was combined in the second instance. For the UWB pose only the x and y values were given while for AMCL the x, y, and yaw values were included in the estimation.

In TF, only one parent is allowed for a given frame. The second UKF node will, for this reason, stop publishing the map to base link transformation, listen for the odom to base link transformation and create the desired map to odom transform. Usually, AMCL is responsible for creating the transform between the global map frame and the odometry frame and as such it was important to prevent AMCL to publish this transform. Similarly, in the default navigation stack configuration the moving base node, responsible for navigating the robot in the environment, expects pose data from AMCL. Changes were made to prevent this and let the node get pose data from the second UKF node instead.

Lastly, URDF files ware added for the UWB tag to visually represent the placement of the UWB tag relative to the base link of the robot.

5.4 Tests and results

In order to test the localisation of the robot in the environment with UWB, three distinct tests were carried out. The first one involved testing the ros_dwm1000 package, with the modifications made, to evaluate how accurately the tag's position was estimated. The test was conducted in an indoor corridor environment with LOS. The corridor is depicted in figure 23. The anchors were placed at the same height of 1 meter, and 4 meters apart from each other. The robot had the tag mounted on itself and was instructed to move around in the 16 square meter area.

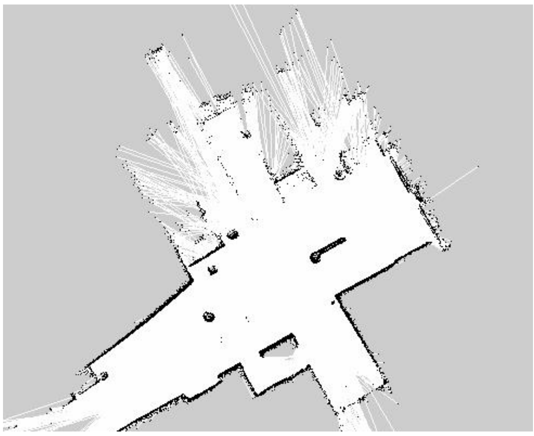


Figure 23: Map of the corridor slammed by the robot

The corridor, the anchors and the tag were visualised in the Rviz to view the performance of the software. The anchors were also instructed to send the estimated distance of the target relative to themselves. These values were compared to the real distance measured. Accuracy to within ±20 cm was observed. However, the test was also conducted in NLOS office environment of a 5 square meter area. Chairs, tables, humans and other objects were present. It was noted that when humans prevented LOS between two anchors the communication between them momentarily stop working, resulting in large errors in the estimation of the target's position. As soon as the human moved away from the test area the communication was restored.

The third test was carried out in the LOS corridor environment and the anchors were arranged in the same way as the first test. This time the odometry data, UWB and AMCL pose was fused together with the robot localisation package and the robot was instructed to autonomously move around the area. The results were unsatisfactory. The robot could not localise itself and as a result, not navigate properly.

5.5 Discussion

Regarding the UWB localisation, the accuracy of the target's estimated position utilising the Decawave UWB modules together with the ros_dwm1000 software was shown satisfactory. In LOS environment the ranging measurements had an accuracy of ±20 cm. However, it was evident that strong signal attenuation occurred when humans were near the UWB modules and disrupted the

LOS condition. Therefore, to avoid having an obstacle between your the target device and the reference nodes, the nodes have to be placed high enough to get a clear line of sight.

For the integration of the UWB pose to further attempt to refine the estimation of the robot pose, the robot localisation software provided poor performance. The position estimation accuracy, on the contrary, was lower when fusing all positioning data provided by the different sources. The expected result is that the more data sources that provide position estimations of a device, the more accurate the estimation should be. The poor performance is believed to occur due to misconfigured parameters. In the UKF setup, there are many configurable parameters to consider. There is a tutorial showing the best practices for sensor integration however achieving precise behaviour for the state estimate requires knowledge and understanding of every single configurable parameter available. The authors, unfortunately, did not have enough time to tune every parameter.

6 Improvements and changes

Author: Ulrik Åkesson

In this section, the state of Charlie as of the end of the project is described. That includes a description of the architecture 6.1, the changes made to the physical appearance of Charlie 6.2, and the reasoning behind the implementation of a new path planning algorithm 6.3.

6.1 System architecture

In this iteration of the project, two blocks were added to the system architecture. The new additions to the system are the UWB tag, which is used for localisation, and the lift system designed for handling the bins. Each of these are described in further detail in sections 5 and 3 respectively. In figure 24 the block diagram of the system architecture is presented.

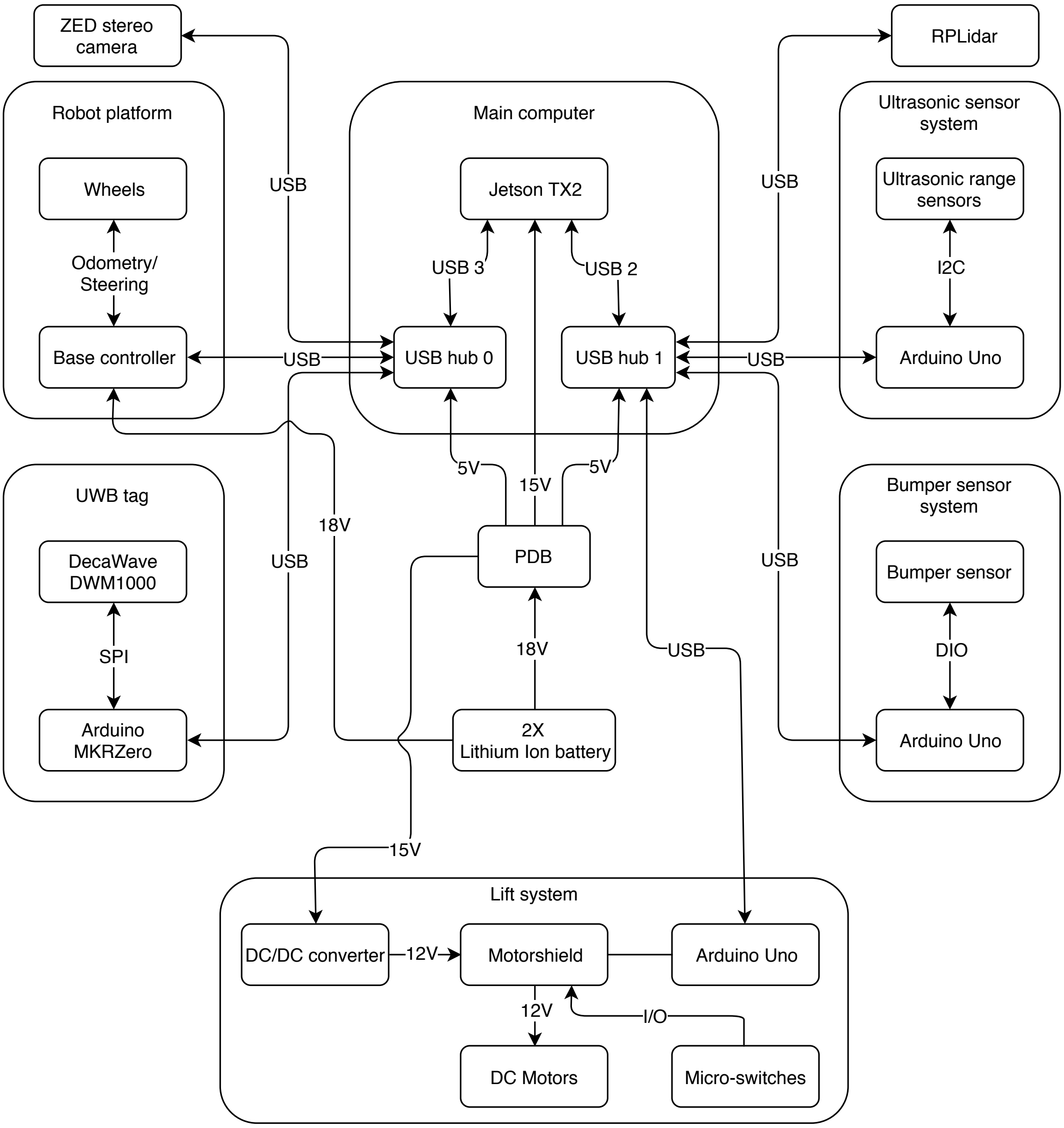
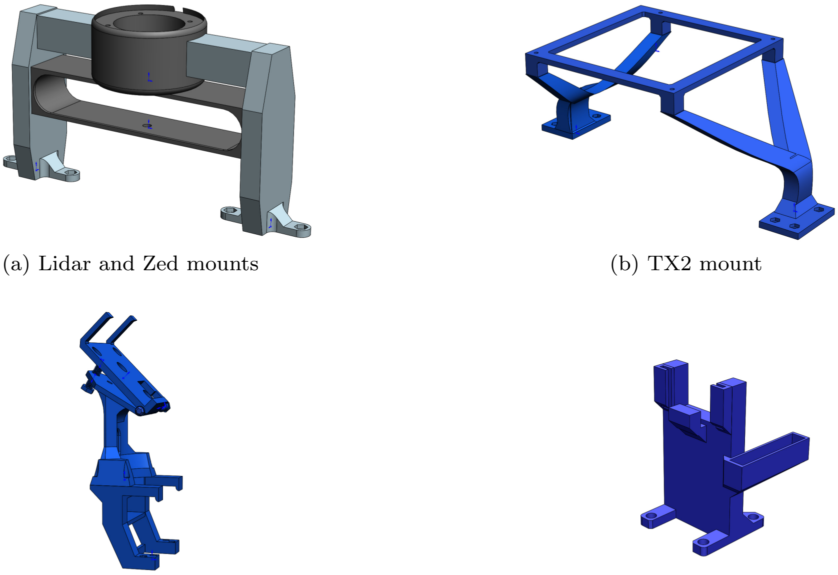


Figure 24: The robot's architecture at the end of the project. New for this iteration is the lift system and the UWB tag.

6.2 Physical changes

During this project, the need to adapt some of the physical characteristics of Charlie was discovered. For the first, it was discovered that the sensor-mounts needed reinforcement, and to be placed in a way that made it easier to use the positions of the sensors in ROS. The mounts that were created for the peripherals can be seen in Figure 25. Furthermore, when adding the new mounts to the old base-plate on which all components are mounted, was found that it was too flexible to support the Zed camera and Lidar tower which swayed back and forth. Therefore the old base-plate was exchanged to a sturdier one, eliminating the problem.



(a) Lidar and Zed mounts

(b) TX2 mount

(c) Projector mount

(d) UWB mount

Figure 25: The main components have all had new mounts designed for them which can be seen in this figure.

6.3 Path planning

Author: Sweta Chakraborty

Nowadays, there have been many research areas focusing on intelligent navigation of autonomous robots as observed in the study of H.Zhang *et al* [35], A.C. Murillo *et al.* [36] and V.D. Hoang *et al.* [37]. The four basic components of the navigation stack are shown in Figure 26. One of the most complex and vital building block components of the navigation stack is path planning. Therefore, path planning is a growing research aspect in the autonomous navigation of a mobile robot. A survey conducted by Thi Thoa Mac *et al.* [38] compared different navigation algorithms employed for autonomous robot locomotion.

The path planning algorithm can be classified into global path planner and local path planner categories. The global path planner generates a low-resolution high-level path for the robot to traverse in the workspace, thereby resulting in minimization of computation time, distance and energy expenditure. A known map is provided to the global planner which helps to produce a collision-free path for the robot in advance, by avoiding static obstacles present on the map. This is called an off-line algorithm. On the other hand, a local planner is capable of producing high resolution, low-level paths in response to dynamic environmental changes. This approach is called an on-line algorithm.

In this iteration of the UNICORN project, the path planning algorithm is revisited to implement recent advancements in this area. Considering time limitation and complexity of the problem, this project has focused only on the implementation of the global path planner. Although the global planner is able to produce an optimal path by avoiding static obstacles in the map, it is found

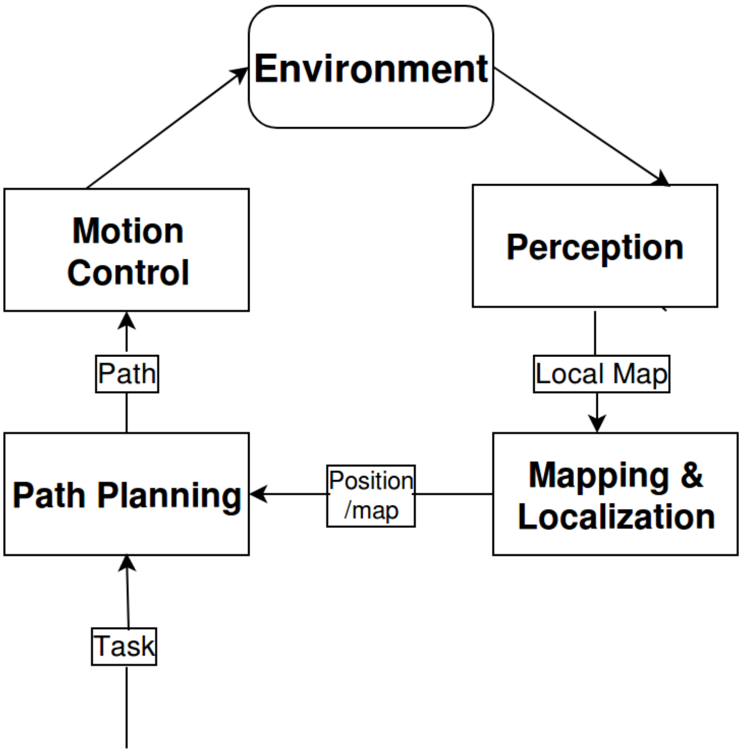


Figure 26: Robot Navigation Structure

lacking in its reaction to certain cases of dynamic or unforeseen obstacles cropping up in the environment. An appropriate local path planner is, therefore, necessary to be incorporated, which is capable of eliminating this inherent weakness of the global planner.

So far, numerous path planning algorithms and implementations have been proposed such as Dijkstra, A*, D* and several others, for autonomous navigation of a mobile robot. Different algorithms may complete the same task by taking a similar amount of time, energy consumption, efficiency and space. Some of the algorithms are compared and contrasted below for a better grasp of the topic:

Dijkstra

One of the most conventional yet effective search-based approaches is called the Dijkstra's algorithm. The algorithm constantly searches for the lowest cost path between any two nodes in a map represented as a graph. Firstly, it selects an unvisited node on the map. Then it proceeds to calculate the distance between it and every unvisited node. Finally, it updates the neighbouring node with the shortest distance. For Dijkstra's algorithm to produce the optimal path in the map, the edges in the graph should be non-negative. The algorithm illustrates the shortest path in its operational environment as observed in [39]. However, the biggest challenge with this method is the fact that it follows a blind search mechanism to find the optimal path in the map, which requires a high computation time.

A*

The A* algorithm considers the Best-First-Search method philosophy, using heuristic functions rather than optimal search mechanism [40]. It maintains a tree of paths, originating from a start node and extends those paths across vertical or horizontal edges one at a time. This process iterates until the terminal node has been reached. The algorithm has proven to be highly efficient at finding an optimal path in the map, considering an admissible heuristic function was provided at the outset. To produce the shortest path in the map, the algorithm expands fewer nodes as compared to Dijkstra, thus, in essence, creating a faster search process. The main drawback of the A* algorithm is the huge computational memory requirement. The algorithm is severely space limited in complex environments.

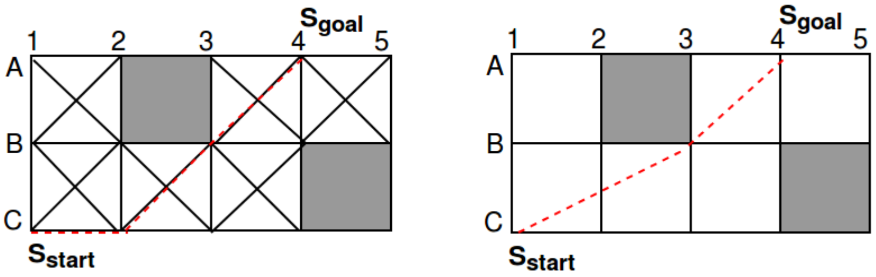
D*

D* is an incremental search algorithm [41], which is a popular re-planning method, seeking to find the shortest path in real time by incrementally restoring paths to the robot's state. It is also known as the dynamic version of the A* algorithm, as the cost function can change when the algorithm is being run. The execution of D* planning can be divided into two distinct phases: initial planning and re-planning. Initial planning is activated at the robot's start position and re-planning is performed during the robot's locomotion as it detects changes of node occupancy in the map. The number of expanded nodes is kept minimum for fast execution [42].

All of the above mentioned approaches possess different strengths and weaknesses. Therefore, selection of an appropriate algorithm for path planning is important in order to ensure that the navigation process runs smoothly. An optimal global path planner should be able to create a short path with low computation time. Lan Anh *et al.* [43] proposed a new global path planner called Theta* with *static flow field* satisfying the aforementioned criteria. The same path planner concept is thereby implemented in the UNICORN project.

Theta *

The Theta* planner, also known as "Any-angle path planner", creates a smooth path to be traversed by the autonomous mobile robot with the inclusion of few turns from the starting point to the final destination. The Theta* algorithm finds a shorter path than the A* as it propagates information to the node edges without constraining the path to the node edges [44]. Thus it results in an any-angle pathfinding algorithm with low computation time. Figure 27 illustrates an optimal path produced by both the Theta* and the A* planners.



(a) A* Planner

(b) Theta* Planner

Figure 27: A* planner versus Theta* planner

While moving towards the destination, the robot may deviate from its original path in order to avoid obstacles. The Theta* planner is a computationally heavy software for a large operational environment, thus it is not advisable to re-calculate a new path for every small update in the map or every small deviation of the robot from it's computed path. To cope with the robot's deviation, a static flow field along the planned path is defined to attract the robot back to the computed path and continue progressing towards the final goal. The basic idea of this method is to assign the obstacle and the goal as repulsive and attractive forces, respectively. In effect, the robot moves towards its target location while moving away from the obstacles along the way. Figure 28 represents how the static flow field is generated around the computed path of the robot.

To reduce the computational load and expedite processing time, the Theta* algorithm is designed to only re-calculate a new path if there are any significant changes in the environment or the robot moves too far from the area covered by the static flow field. The static flow field force for point x near the path produced by the Theta* planner is calculated by equation 9.

$$F_{flow}(x) = F_a(x) + F_r(x) \tag{9}$$

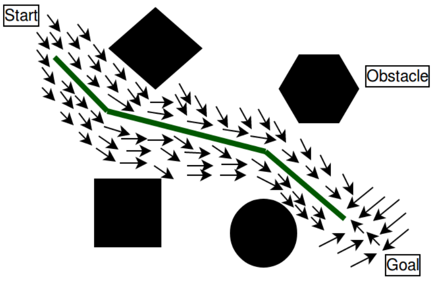


Figure 28: Initial Path Configured with Static Flow field

Here $F_a(x)$ represents the attractive force which draws back the robot towards its configured path and $F_r(x)$ represents the repulsive force created by static obstacles.

6.3.1 Implementation

In the previous year's UNICORN project, an existing path planning algorithm, named Dijkstra's algorithm was included in the ROS navigation stack package. In the current year's approach, the new global path planner, Theta* with static flow field is implemented as a wrapper for the ROS navigation stack. A 2-D map of the operating environment was provided to the planner. The other advantageous functionalities (mapping, localisation and local planner) of the navigation stack remains unchanged. In order to integrate a new global path planner on to the navigation stack, the new planner must adhere to *nav_core::BaseGlobalPlanner* interface as specified in the *nav_core* package. The Theta* planner with static flow field algorithm is added as a plug-in to the ROS navigation stack, which can be further utilised by a *move_base* node.

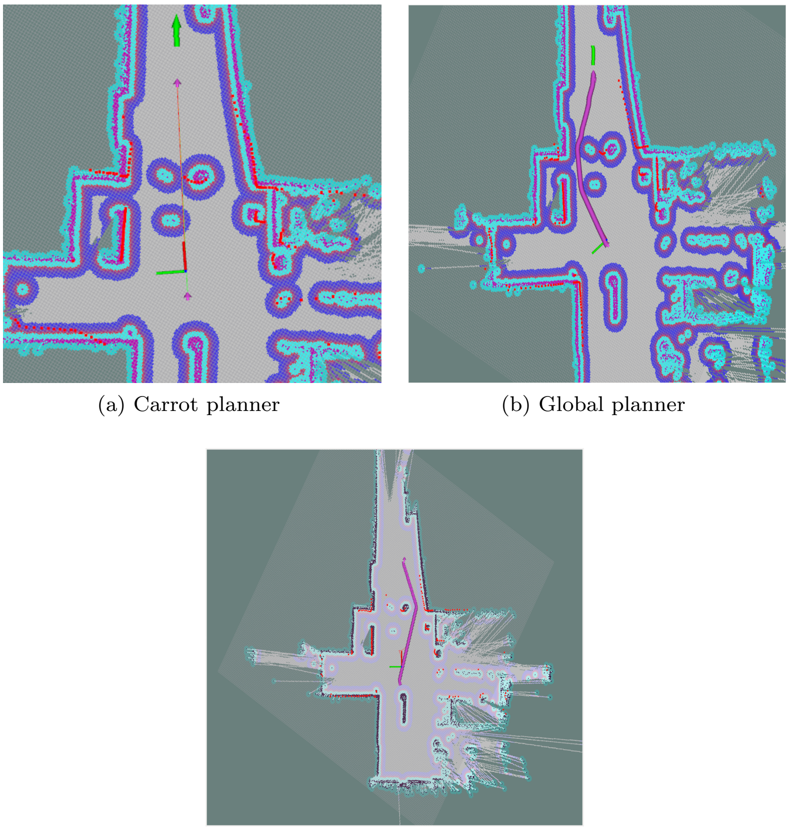
6.3.2 Test & Result

To test the new global path planning algorithm, three distinct tests were performed. The tests were carried out in an indoor environment and the results were observed using the visualisation tool Rviz. The robot was provided with a map of the environment and the goal position was assigned by the user in Rviz.

- Initially, the *carrot planner* was implemented as a plug-in to the *nav_core::BaseGlobalPlanner* interface for the *move_base* node. The main purpose for testing this planner was to make sure that Charlie are able to follow a straight path before testing the more advanced planners. This planner will compute a valid goal position for the local planner. The planner visualises the obstacles present in the map, and it moves backwards until the next goal point free from any obstacle is found. Figure 29a below shows the path created by the *carrot planner*.
- Secondly, the existing *global planer* (Dijkstra's algorithm) was tested. Figure 29b below shows the shortest path produced by the Dijkstra's algorithms
- Finally, a new Theta* with static flow field algorithm was also tested (see Figure 29c). The path produced by the new planner is visually similar to Dijkstra's algorithm.

6.3.3 Discussion

All the implemented path planning algorithms were able to produce a path from a starting position to the final goal location, provided with a map of the environment to the path planner. The *carrot planner* was unable to produce a collision free path towards its destination location. This planner does not necessarily consider the static obstacles in the map. Instead, it positioned the robot as close as possible to the defined final goal location. Thus, this planner is inappropriate to implement for autonomous navigation of a mobile robot. Dijkstra's algorithm found the shortest collision free path between the starting and the goal position. Finally, the Theta* path planner also generated



(a) Carrot planner

(b) Global planner

(c) Theta* planner

Figure 29: Path generated by different planners

a collision free path from an initial point to the final location. The optimal path generated by the Theta* planner is visually similar to Dijkstra's algorithm. To select the best planner for autonomous navigation for Charlie, it is necessary to conduct a software test on computation time for both Dijkstra's and the Theta* planner. However, because of the time limitation, such tests were not performed in this project.

7 Future work

Author: Ulrik Åkesson

Even though a lot has been done in the different areas during this iteration of the project, there are still many areas to continue working in. During this iteration, the scope was limited to an indoor environment and thus only some attention where given into how our system would fare outdoors. Therefore adapting the robot for an outdoor environment is necessary, a suggestion is to investigate how to weatherproof the different peripherals.

Author: Mujtaba Hasanzadeh

The lifting system works properly but there are still some improvements and changes that need to be done. When the DC/DC converter is connected to the PDB and the lift state is chosen, a voltage drop occurs, it is because of the PDB that can not generate enough current for the motors. To deal with this issue, a new PDB could be designed. Maybe a 12V port can be created directly on the new PDB and thus removing the need of an external converter. Currently, the lifting system can be activated manually i.e by choosing the lift state in the state machine. However, currently no solution for finding the refuse bin exist and thus needs to be developed. It could be beneficial to add a weight measuring sensor to the lift system. That would enable the system to avoid lifting heavier than 15 kg, thus minimising the risk of parts breakage. Some mounts did break during the testing of the lift system, they should be created and mounted on the Charlie again, maybe in another material or with a new design. The micro-switches are fragile and can easily break from repetitive use. Exploring a contact-free solution might be beneficial.

Author: Sebastian Andersson
Another aspect that might need changes is the refuse handling. Today, the possibility to lift a bin is solely dependent on that the lift system is positioned perfectly under the bin. Some sensor solution or a more dynamic mechanical solution might be required. Furthermore, a late mounting addition was made. Currently, without that addition, the bin will press the lower switch prematurely. There are many moving parts on Charlie, and it might be a good approach to construct some sort of cover to keep items from getting stuck.

Author: Marielle Gallardo
One of the main objectives of the UNICORN project is to utilise a robust communication medium between the trash-collecting robots and a control centre. The robots need to be able to send their position to the control centre, that is responsible for giving further instructions. In this project, the robust UWB medium was investigated, implemented and tested for positioning and considered for potential communication medium. The implementation of UWB as a communication channel was not realised and as such should be further investigated to evaluate its performance. Another radio technology that has been receiving attention is 5G. Research could be conducted to see whether 5G is more promising than UWB. Further investigation is advised regarding the lower accuracy in the estimation of the robot position when fusing several estimations. Tuning parameters could be the solution for achieving high positioning accuracy. The accuracy is also dependent on the number of reference nodes. It was observed that when one anchor lost communication the position estimation was completely incorrect. Maybe this could be prevented by implementing more anchors.

Author: Alexandra Hengl
The path intention communication has been implemented based on the robot's direction commands sent as ROS messages. These messages only tell where the robot is directed to move but they are not based on any heuristic about a concrete path. Such a path could be created by prediction, by utilising knowledge about the end goal and which way the robot is facing. If the robot is facing a different direction than the location of the end goal, then the next direction can be predicted. To increase the precision of prediction, perhaps knowledge about previous directions can be taken into consideration when constructing the predicted path.

Author: Sweta Chakraborty
Future research investigating path planning algorithms could be directed towards comparing the computation run-time for both Dijkstra's and Theta* planner algorithm. This would be beneficial to identify the fastest and more efficient path planner for autonomous mobile robot navigation. In addition, the work could possibly be extended with implementing the local path planner proposed by Lan Anh et al. [43]. Finally, it would be interesting to validate the algorithm in an outdoor environment that resembles complex environmental conditions.

8 Conclusion

Author: Ulrik Åkesson
In this iteration of the UNICORN project, considerable steps have been made within three main areas. The robot has been equipped with a lift system which in tests was able to lift and drive around with a refuse bin. By mounting a projector the robot, it has now a way of showing its intention while manoeuvring. The possibility of increasing the efficiency of the localisation has been explored by using UWB technology with mixed results. The UWB modules and the package on their own provided good results. However, fusing it with the other sensor data did not succeed and more work is needed in that area. With the work done in those areas and the results achieved, the vision of automating refuse handling in residential areas is closer to being envisioned. This

project has been a great learning experience for everyone involved, in part by introducing a lot of the different steps that are a part of building and working on a prototype like Charlie.

References

[1] U. Andréasson, "Renhållning på liv och död," *Populär Historia*, March 2001, accessed 2018-11-21. [Online]. Available: https://popularhistoria.se/vardagsliv/hygien-halsa/renhallning-pa-liv-och-dod

[2] SMED (Svenska MiljöEmissionsData), "Avfall i sverige 2016," http://www.naturvardsverket.se/978-91-620-6839-4, June 2018.

[3] B. Lindgren and M. Ekström, "Student project - unicorn," Academy of innovation, design, and engineering, Tech. Rep., January 2018. [Online]. Available: http://www.es.mdh.se/publications/5121-

[4] O. S. R. Foundation, "Ros wiki," "http://wiki.ros.org/", accessed: 2019-02-03.

[5] B. Lindgren and G. Kuosmanen, "An autonomous robot for collecting waste bins in an office environment," 2018.

[6] ELFA, "Dc motors," "https://www.elfa.se/Web/Downloads/55/07/05445507.pdf", 2018, accessed: 2018-10-22.

[7] H. J. Zhang, "Basic concepts of linear regulator and switching mode power supplies," *Analog Devices*, 10 2014.

[8] Farnell, "Dc/dc converter," "https://se.farnell.com/murata-power-solutions/uwe-12-6-q12p-c/converter-dc-dc-12v-6a/dp/1652245", 2018, accessed: 2018-10-22.

[9] ——, "Microswitch," "https://se.farnell.com/omron-electronic-components/d2fs-fl-n/microswitch-lever-spst-0-1a-6vdc/dp/2402445?st=microbrytare%20spak%20miniatyr", 2018, accessed: 2018-10-22.

[10] Arduino, "Arduino uno," "https://store.arduino.cc/arduino-uno-rev3", 2018, accessed: 2018-10-22.

[11] Arduino, "Motor shiled," "https://store.arduino.cc/arduino-motor-shield-rev3", 2018, accessed: 2018-10-22.

[12] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa, "Effects of robot motion on human-robot collaboration," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction.* ACM, 2015, pp. 51–58.

[13] A. Watanabe, T. Ikeda, Y. Morales, K. Shinozawa, T. Miyashita, and N. Hagita, "Communicating robotic navigational intentions," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 2015, pp. 5763–5769.

[14] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal, "That's on my mind! robot to human intention communication through on-board projection on shared floor space," in *Mobile Robots (ECMR), 2015 European Conference on.* IEEE, 2015, pp. 1–6.

[15] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an autonomous mobile robot based on ros," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sept 2017, pp. 1–5.

[16] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.

[17] J. Fu and S. Pan, "Uwb-over-fiber sensor network for accurate localization based on optical time-division multiplexing," in *2013 12th International Conference on Optical Communications and Networks (ICOCN)*, July 2013, pp. 1–4.

[18] R. S. Kulikov, "Integrated uwb/imu system for high rate indoor navigation with cm-level accuracy," in *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, March 2018, pp. 1–4.

[19] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/5/707

[20] X. Li, J. He, L. Xu, and Q. Wang, "The effect of multipath and nlos on toa ranging error and energy based on uwb," in *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, May 2016, pp. 1–2.

[21] A. Azeez Khudhair, S. Jabbar, M. Qasim Sulttan, D. Wang, and M. Sulttan, "Wireless indoor localization systems and techniques: Survey and comparative study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, pp. 392–409, 08 2016.

[22] K. Yimei and W. Jiawei, "A high-precision toa-based positioning algorithm without the restriction of strict time synchronization for wireless systems," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, Nov 2016, pp. 1666–1670.

[23] X. Li, Z. Deng, L. T. Rauchenstein, and T. Carlson, "Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements," *Review of Scientific Instruments*, vol. 87, p. 041502, 04 2016.

[24] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 547–568, 2014. [Online]. Available: https://doi.org/10.1177/0278364913500364

[25] Y. Jiang and V. Leung, "An asymmetric double sided two-way ranging for crystal offset," 07 2007, pp. 525 – 528.

[26] B. Grosswindhager, C. A. Boano, M. Rath, and K. Römer, "Concurrent ranging with ultra-wideband radios: From experimental evidence to a practical solution," 07 2018.

[27] A. Zanella and A. Bardella, "Rss-based ranging by multichannel rss averaging," *IEEE Wireless Communications Letters*, vol. 3, no. 1, pp. 10–13, February 2014.

[28] A. Jiménez and F. Seco, "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. PP, pp. 1–12, 04 2017.

[29] *Ultra Wideband Module*, decaWave, 2016, ver 1.7.

[30] J. M. Jimeno, "linorobot/ros_dwm1000," "https://https://github.com/linorobot/ros_dwm1000", 2017, accessed: 2018-10-15, Last updated 2017-05-02.

[31] Arduino, "Compare board specs," "https://www.arduino.cc/en/products.compare", 2018, accessed: 2018-10-22.

[32] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[33] E. Marder-Eppstein, "2d navigation stack," "http://wiki.ros.org/navigation", 2017, accessed: 2019-01-15.

[34] J. S. Gill, “Setup and configuration of the navigation stack on a robot,” ”http://wiki.ros.org/navigation/Tutorials/RobotSetup”, 2017, accessed: 2019-01-15.

[35] H. Zhang, A. Geiger, and R. Urtasun, “Understanding high-level semantics by modeling traffic patterns,” in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[36] A. C. Murillo, G. Singh, J. Kosecká, and J. J. Guerrero, “Localization in urban environments using a panoramic gist descriptor.” *IEEE Trans. Robotics*, vol. 29, no. 1, pp. 146–160, 2013.

[37] V.-D. Hoang, D. C. Hernandez, M.-H. Le, and K.-H. Jo, “3d motion estimation based on pitch and azimuth from respective camera and laser rangefinder sensing,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 735–740.

[38] D. T. R. K. Thi Thoa Mac, CosminCopot, “Heuristic approaches in robot path planning: A survey,” in *Robotics and Autonomous Systems.* ScienceDirect, December2016, pp. 13–28.

[39] S. M. R. Al-Arif, A. I. Ferdous, and S. H. Nijami, “Comparative study of different path planning algorithms: a water based rescue system,” *International Journal of Computer Applications*, vol. 39, 2012.

[40] S. Puls, P. Betz, M. Wyden, and H. Wörn, “Path planning for industrial robots in human-robot interaction,” in *IEEE/RSJ IROS Workshop on Robot Motion Planning: Online, Reactive, and in Real-Time*, 2012.

[41] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *ICRA*, vol. 94, 1994, pp. 3310–3317.

[42] M. Dakulovic, S. Horvatic, and I. Petrovic, “Complete coverage d* algorithm for path planning of a floor-cleaning mobile robot,” in *18th international federation of automatic control (IFAC) world congress*, 2011, pp. 5950–5955.

[43] L. A. Trinh, M. Ekström, and B. Cürüklü, “Toward shared working space of human and robotic agents through dipole flow field for dependable path planning,” *Frontiers in neurorobotics*, vol. 12, 2018.

[44] A. Nash, K. Daniel, S. Koenig, and A. Felner, “Theta\^*: any-angle path planning on grids,” in *AAAI*, vol. 7, 2007, pp. 1177–1183.

List of Figures

List of Tables